FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Community-based Sports Articles Generation Platform using NLG and Post-Editing

Pedro Miguel Sousa Fernandes

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Sérgio Nunes

Second Supervisor: Luís Santos

July 23, 2021

Community-based Sports Articles Generation Platform using NLG and Post-Editing

Pedro Miguel Sousa Fernandes

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. João Pedro Mendes Moreira
External Examiner: Prof. Alípio Jorge
Supervisor: Prof. Sérgio Nunes

July 23, 2021

Abstract

Every week, hundreds of sports matches are played. From these, a large collection of data is generated and stored in databases, to be displayed in sports websites and applications. Such data can be used as a basis to write news articles, and there is an opportunity to utilize it and produce informative content through automation. To explore the potential of the great volume of existing data, automatic sports news generation tools are increasingly common. Leveraging natural language generation techniques, these tools, categorized as data-to-text, are able to process structured data about a match (e.g. goals, fouls, substitutions), and produce a text piece that summarizes a match.

One of these tools is Prosebot, a system developed by ZOS – a Portuguese media company that holds a large database of sports games, players, and teams, with a reported total amounting to 5 million items. Prosebot is a template-based system, whose templates are textual constructions where variables are interpolated and allow the generation of sentences. There is room for improvement in the system, however, the main problem addressed in this work is how to accelerate the process of creating textual summaries for more matches.

To achieve this, we present a community-based platform for the publication of sports match summaries using a draft generated by a data-to-text system and human post-editing. The platform presents *zerozero.pt's* collaborators a text initially generated by Prosebot describing general information about the match, its events, and the post-match situation. This text can be freely edited, and is automatically published to *zerozero.pt* upon submission. This platform was designed with the help of *zerozero.pt*'s journalists via a survey, which also inquired on the impact of automatic tools in newsrooms. This survey was later conducted with Portuguese journalists from different backgrounds. Results show an overall negative sentiment on automatic tools, as journalists do not recognize advantages such as time-freedom or wider coverage, but believe they may be replaced in the future. Therefore, journalists are not favorable to the unsupervised publication of automatically generated news articles. In constrast, *zerozero.pt*'s newsroom is much more receptive, which may be a result of their previous contact with Prosebot.

For the platform's assessment, we have conducted a human evaluation through a survey presented after a summary submission, as well as an automatic evaluation via the comparison of Prosebot's drafts and the submitted texts. The results show a very positive reception from the collaborators, as well as a high amount of generated text kept in the final summary, indicating it was useful for the collaborators.

Keywords: Natural Language Generation, Automated Journalism, Sports Journalism

Resumo

Todas as semanas jogam-se centenas de partidas desportivas. Para cada jogo, uma coleção de dados é gerada e guardada em bases de dados, que pode ser mostrada em *websites* e aplicações desportivas. Estes dados podem ser usados como base para escrever conteúdo noticioso, e há uma oportunidade para os utilizar e produzir conteúdo informativo através de automação. Para explorar o potencial do grande volume de dados existentes, têm surgido ferramentas de geração automática de notícias desportivas. Usando técnicas de geração de linguagem natural, estas ferramentas, categorizadas como "dados-para-texto", processam dados estruturados sobre um jogo (como golos, faltas, substituições), e produzem um texto que sumariza uma partida.

Uma destas ferramentas é o Prosebot, um sistema desenvolvido pela ZOS - uma empresa de comunicação social que gere uma grande base de dados de jogos, jogadores, e equipas, com um total aproximado de 5 milhões de entidades. O Prosebot é um sistema baseado em *templates*, que são construções textuais onde se podem interpolar variáveis e assim permitir a geração de frases. Há espaço para melhorias no sistema, no entanto o maior problema que este trabalho endereça é como acelerar a produção de conteúdo noticioso de forma a abranger um maior número de jogos.

Assim, este trabalho apresenta uma plataforma suportada na comunidade do *zerozero.pt* para a publicação de sínteses de jogos usando um texto gerado pelo Prosebot e pós-edição. A platforma fornece aos colaboradores do *zerozero.pt* um texto inicialmente gerado pelo Prosebot descrevendo informação geral sobre o jogo, os eventos, e o rescaldo. Este texto pode ser editado livremente, e é automaticamente publicado no *zerozero.pt* no ato da submissão. Esta plataforma foi desenhada com o contributo de jornalistas do *zerozero.pt* através de um inquérito, que tinha também o objetivo de estudar o impacto de ferramentas automáticas nas redações. Este inquérito foi também conduzido com jornalistas portugueses, de diferentes contextos, cujos resultados mostram uma perceção negativa sobre ferramentas automáticas, visto que os jornalistas não reconhecem vantagens como aumento de tempo livre ou aumento da cobertura, mas acreditam que podem vir a ser substituidos no futuro. Assim, os jornalistas não se mostram favoráveis à publicação de artigos gerados automaticamente sem supervisão. Por outro lado, a redação do *zerozero.pt* mostrou-se mais recetiva, que pode ser um resultado da sua experiência com o Prosebot.

Para a avaliação da plataforma, conduzimos avaliação humana ao apresentar um inquérito após a submissão de uma síntese, bem como avaliação automática através da comparação dos textos gerados pelo Prosebot e as versões submetidas pelos colaboradores. Os resultados mostram uma receção bastante positiva pelos utilizadores, e que grande quantidade do texto inicial é mantido na síntese final, mostrando que foi útil para os colaboradores.

Keywords: Geração de Linguagem Natural, Jornalismo Automático, Jornalismo Desportivo

Acknowledgements

Firstly, I'd like to thank my friends and family, who have always supported and accompanied me throughout the recent months, making this exceptional situation seem just a little bit easier.

Secondly, I want to thank the whole staff at *zerozero.pt*, and in particular Marco Sousa, Pedro Dias, Vasco Ribeiro, and Gonçalo Silva, for their commitment to the project, giving me the necessary conditions to develop my work.

Finally, I want to thank professors Luís Santos and Sérgio Nunes, for sharing their time, knowledge, and guidance.

Pedro Fernandes

“Everything in this world is magic,
except to the magician.”

Dr. Robert Ford, Westworld

Contents

1 Introduction **1**
1.1 Context . . . 1
1.2 Motivation . . . 1
1.3 Objectives . . . 2
1.4 Document Structure . . . 2

2 Automated Sports Journalism **3**
2.1 Literature Review Methodology . . . 3
2.2 Natural Language Generation . . . 4
2.3 NLG Tasks . . . 4
2.3.1 Content Determination . . . 5
2.3.2 Text Planning . . . 5
2.3.3 Sentence Aggregation . . . 5
2.3.4 Lexicalisation . . . 6
2.3.5 Referring Expression Generation . . . 6
2.3.6 Surface Realisation . . . 6
2.4 NLG Applications in Journalism . . . 7
2.5 Post-Editing and "Machine-in-the-loop" in NLG . . . 9
2.6 Impact of Automatic Tools in the Newsrooms . . . 10

3 Evaluation Methods **13**
3.1 NLG Evaluation . . . 13
3.2 Machine-in-the-loop Evaluation . . . 15

4 Prosebot platform **17**
4.1 NLG Module . . . 17
4.1.1 Previous Work . . . 17
4.1.2 Prosebot . . . 18
4.1.3 Data Source . . . 19
4.1.4 Template Design . . . 19
4.1.5 Content Selection and Document Structure . . . 20
4.1.6 Algorithm . . . 21
4.2 Improvements to the NLG Module . . . 22
4.2.1 Mention relevant players . . . 22
4.2.2 Extend compatibility to different formats . . . 23
4.2.3 Curiosities: streaks and records . . . 24
4.2.4 Extend support to other sports . . . 24
4.2.5 Head-to-head information . . . 25

4.2.6 Bug fixes . . . 25
4.2.7 Addition of templates . . . 26
4.2.8 Various changes . . . 26
4.3 Survey to the *zerozero.pt* Newsroom . . . 28
4.3.1 Planning of the Platform . . . 30
4.3.2 Impact of Automatic Tools in the Newsroom . . . 31
4.4 Prosebot Integration in *zerozero.pt* . . . 33

5 Evaluation **37**
5.1 Methodology . . . 37
5.1.1 Post-Submission Survey . . . 37
5.1.2 Automatic Evaluation . . . 37
5.2 Results . . . 40
5.2.1 Summaries Characterization . . . 40
5.2.2 Post-Submission Survey . . . 41
5.2.3 Text Comparison . . . 43

6 Survey on the Impact of Automatic Tools on the Newsroom **49**
6.1 General Characterization . . . 50
6.2 Impact of Automatic Tools on the Future of Journalism . . . 52
6.3 Further Analysis . . . 56

7 Conclusions and Future Work **59**
7.1 Conclusions . . . 59
7.2 Future Work . . . 60

A Software Requirements Specification **61**
A.1 Introduction . . . 61
A.1.1 Purpose . . . 61
A.1.2 Intended Audience and Intended Use . . . 61
A.1.3 Project Scope . . . 61
A.1.4 Definitions . . . 62
A.1.5 Risks . . . 62
A.2 Overall Description . . . 62
A.2.1 Product Perspective . . . 62
A.2.2 Product Functions . . . 62
A.2.3 User Classes and Characteristics . . . 63
A.2.4 Design and Implementation Constraints . . . 63
A.2.5 Assumptions and Dependencies . . . 64
A.3 System Features and Requirements . . . 64
A.3.1 User Interfaces . . . 64
A.3.2 Software Interfaces . . . 67
A.3.3 System Features . . . 67

B Surveys **71**
B.1 Introduction of Automatic Tools in *zerozero.pt*'s Newsroom . . . 71
B.2 Impact of Automatic Tools on the Future of Journalism . . . 78

References **87**

List of Figures

2.1 Systematic Literature Review method. 4

4.1 Sample of text generated by Prosebot in English. 21
4.2 Summary of results from the planning section of the survey. 29
4.3 Opinion on the name that should be used to refer to Prosebot's texts. 30
4.4 Perceived importance of automatic tools through the various stages of news production. 30
4.5 Summary of results from the survey. 32
4.6 Rich picture of the Prosebot platform. 33
4.7 Disclaimer shown on Prosebot's summaries, referencing Prosebot and the author. 34
4.8 The top section of a *zerozero*'s match page, including a banner for the associated summary. 35

5.1 Post-submission survey. 38
5.2 Sample of the results returned by the get_opcodes function. 39
5.3 Most represented competitions. 41
5.4 Distribution of visits to summaries. 41
5.5 Time to publish the summary after the match takes place. 42
5.6 Survey results summary. 42
5.7 USER metric distribution. 43
5.8 Dice coefficient distribution. 43
5.9 Length difference between original draft and published text. 44
5.10 Distribution of added text per type of word. 46
5.11 Distribution of removed text per type of word. 46
5.12 Distribution of types of words kept by the collaborators. 47

6.1 Age of the respondents. 49
6.2 Years in the field. 49
6.3 Contractual relationship. 50
6.4 The journalists' main editorials. 50
6.5 Journalists' mediums of communication. 51
6.6 Previous contact with automatic tools. 51
6.7 Sample of automatically generated finances article from Lusa. 52
6.8 Likelihood of applying automatic tools at the journalists' newsroom. 53
6.9 Importance attached to automatic tools in the various stages of news production. . 53
6.10 Results for Likert-scale questions. 54
6.11 How useful are automatic tools in sports, finances, elections and weather. 56
6.12 Response distribution according to the answer to the question regarding previous contact with automatic tools. 56

6.13 Response distribution according to the editorials of the journalists. 57
6.14 Response distribution according to the age of the journalists. 57

A.1 Rich picture of the platform. 63
A.2 Component diagram for the summary creation process. 64
A.3 Hub sample image. 64
A.4 Match page sample image. 65
A.5 Editor sample image. 65
A.6 Post-submission survey sample image. 65
A.7 Summary page sample image. 66
A.8 Moderator dashboard sample image. 66
A.9 Sitemap. 67

B.1 Survey on the impact of automatic tools *zerozero.pt*'s newsroom and the journalists' opinion on the Prosebot platform. 77
B.2 General survey on the impact of automatic tools on the future of journalism. . . . 86

List of Tables

2.1 List of Journalism NLG systems. 7

3.1 List of Machine-in-the-loop story generation platforms and their evaluation methods. 16

4.1 Data stored from the API request to *zerozero.pt*. 19
4.2 Available templates. 20
4.3 List of matches and their relevant players. 23
4.4 Final paragraphs of summaries generated for the Real Madrid vs Chelsea match. . 24
4.5 Currently available match curiosities. 25
4.6 Labels for the questions from the planning section of the survey. 29
4.7 Survey question labels. 32

5.1 Samples of values returned by USER and the DICE coefficient. 39
5.2 General indicators. 40
5.3 Pearson's correlation between user judgments and metrics. 44
5.4 Most frequently removed bigrams/trigrams. 45
5.5 Most frequently added bigrams/trigrams. 45

6.1 Survey question labels. 54
6.2 Most frequent *n-grams* for the question of ethical problems. 55
6.3 Most frequent *n-grams* on the question of characteristics impossible to replicate. . 55

Chapter 1

Introduction

1.1 Context

ZOS is a Portuguese media company responsible for the website zerozero.pt. It also holds websites in other languages, targeted at different countries, such as Brazilian Portuguese, English, and Italian. Through these websites, ZOS provides easy access to match information and statistics, along with news coverage for the most important competitions, provided by a team of journalists. ZOS also encourages community participation by allowing users to register on the website, answer to polls, comment on news and match pages, and even contributing information for matches, teams, and players.

ZOS has developed Prosebot, a tool to automatically generate a textual match report based on structured data about that match. Prosebot applies concepts and techniques from the area of Natural Language Generation, similarly to other tools that have helped journalists in the fields of finance, weather forecast, and sports. Currently, Prosebot is a working system, but it is only being used internally since until now it has been evaluated with the help of ZOS's journalists and automatic methods. Thus, we have decided to bring it to the public, since it is an opportunity to evaluate Prosebot with the involvement of its intended audience, and to increase the visibility of the teams.

1.2 Motivation

One of the main arguments for the development of Prosebot is the amount of structured data available. Every week, hundreds of matches are played and, for each one, *zerozero.pt* gathers data regarding players, goals, substitutions, fouls, among others. However, due to human resource limitations, only a small fraction of matches are covered with human-written articles. According to ZOS, from a total of approximately 700,000 complete match pages, there are only 10,000 with an associated article, i.e. only 1.4% of the matches have a news article. Therefore, there is certainly an opportunity to fill the gap by taking data from each match and automatically generating an article. Journalists generally cover the most relevant leagues (e.g. the "Primeira Liga"), however the community is still interested in lower-level leagues (e.g. district competitions), which are not covered.

Finally, Prosebot could potentially change the way journalists work. By significantly reducing the time spent studying a game's events and related statistics to create an article, Prosebot can help journalists by providing a competent draft for them to edit before publication. Additionally, they can focus on creating more diverse and in-depth content: previews, interviews, documentaries, opinion columns, or teams and players biographies.

1.3 Objectives

The main goal of this dissertation is to accelerate and improve the publishing process of news, using Prosebot and the community's collaboration, in order to generate articles for a greater number of matches. Additionally, we intend to create means to improve and diversify the text generated by Prosebot.

1.4 Document Structure

This report has five main chapters. Chapter 2 introduces the field of Natural Language Generation (NLG) and the advancements in the area of Automated Sports Journalism. Chapter 3 includes a review of existing evaluation methods for NLG systems. Chapter 4 describes the architecture of the Prosebot NLG module and the overall platform, and also presents the results of a survey conducted with *zerozero.pt*'s journalists. Chapter 5 explains our evaluation methodology and presents the results produced by the platform. Chapter 6 presents the results of a survey conducted with Portuguese journalists on the impact of automatic tools in the newsroom. Finally, Chapter 7 includes the conclusions for the work developed and possible avenues for future development prospects.

Chapter 2

Automated Sports Journalism

2.1 Literature Review Methodology

In order to study the field of Natural Language Generation, the most recent and cited review of the state of the art, by Gatt and Krahmer, was used as a reference [12]. Further, a systematic literature review in the area of automated sports journalism was conducted, for several reasons: to find more examples of NLG applications in the field of journalism, and more specifically sports journalism; to find common evaluation methods; to find systems that sought to include humans in the generation process. Our method is presented in Figure 2.1 and explained in the following paragraphs.

- **Keyword Selection** – By choosing the keywords listed in Figure 2.1 (“natural language generation”, “robot journalist”, “automatic news generation”, “machine-in-the-loop text generation”), we intended to find documents that describe applications of NLG in the fields of journalism and, more specifically sports journalism. In a separate goal, we wanted to find systems where both humans and machines are active agents.

- **Document Search** – We used popular academic search engines in our research process. While Google Scholar has a wider scope, Scopus includes peer-reviewed research literature, and ACL Digital Library is more focused on Computer Science. DBLP was used to retrieve more complete BibTeX citations.

- **Recursive Search** – After selecting a group of the most relevant documents for our goals, we analyzed their references and used search engines to find citations, i.e. other articles that cited these documents.

- **Filtering** – We limited documents to NLG systems for the field of journalism. When looking for articles where humans also actively collaborated, the former filter was lifted since we could not find any documents. However, we excluded works on systems that only generated small texts, such as slogans.

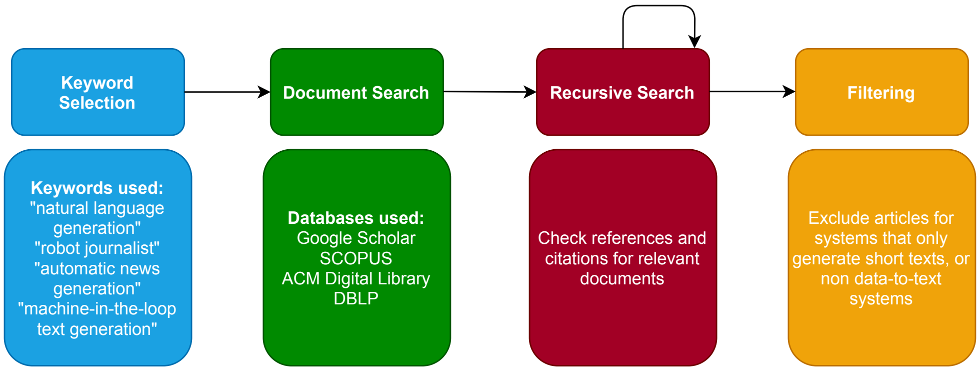

Keyword Selection
Document Search
Recursive Search
Filtering
Keywords used: "natural language generation" "robot journalist" "automatic news generation" "machine-in-the-loop text generation"
Databases used: Google Scholar SCOPUS ACM Digital Library DBLP
Check references and citations for relevant documents
Exclude articles for systems that only generate short texts, or non data-to-text systems

Figure 2.1: Systematic Literature Review method.

2.2 Natural Language Generation

Natural Language Generation has been described by Reiter and Dale [28] as *"the sub-field of artificial intelligence and computational linguistics that is concerned with the construction of computer systems than can produce understandable texts in English or other human languages from some underlying non-linguistic representation of information"*. This definition clearly suits data-to-text better than text-to-text systems. The latter take existing texts as input and output new text. One such application, and perhaps the most popular, is machine translation: the process of translating from one human language to another using a computer. Another very useful application is the automatic correction of spelling or grammar mistakes. On the contrary, data-to-text systems generate texts by taking as input information represented as structured data. Journalism applications, that use natural language generation technologies, are included in this group: soccer [42], weather [29], financial [23], and political [18] reports are all examples of these systems.

2.3 NLG Tasks

Reiter and Dale proposed, in 2000, an architecture comprised of tasks that can be found in most NLG systems [28]. Years later, in 2007, Reiter proposed an extension of this architecture for data-to-text systems whose input is raw data instead of knowledge bases [25]. Gatt and Krahmer [12] enumerate the most common tasks as follows:

1. **Content determination** – Deciding what information will be included in the text.
2. **Text structuring** – Determining where in the text will each information appear.
3. **Sentence aggregation** – Deciding how to build fluent sentences with the information.
4. **Lexicalisation** – Translating the existing information to proper words and terms.
5. **Referring expression generation** – Selecting the words and phrases to properly distinguish domain objects.

6. **Linguistic realisation** – Building well-formed sentences and paragraphs through the combination of words and phrases.

The following sections present each task in more detail.

2.3.1 Content Determination

The first step is to determine which information will or will not be included in the generated text. Usually, the input data contains a large amount of information, and in many cases it is not necessary to convey all of it through text. For example, football matches may have dozens of fouls and while some may be pivotal for the result (i.e. player commits a violent foul and is expelled), most of them are inconsequential and thus not worth reporting.

2.3.2 Text Planning

Having decided the content that will be featured in the text, the system needs to determine the order of presentation. In the case of soccer, the articles generated by PASS [42] start with a match preview and results, then continue with a chronological report of the most important events, and end with information for future matches.

2.3.3 Sentence Aggregation

In this step, we take one or more related sentences and transform them in order to improve readability and fluidity. Reiter presented the following techniques [28]:

- Conjunction and other aggregation. For example, transforming (1) into (2):

    1. Rúben Neves came on at minute 56'. Rúben Neves scored the second goal.
    2. Rúben Neves came on at minute 56' *and* scored the second goal.

- Pronominalization and other reference. For example, transforming (1) into (2):

    1. José Mourinho was at the press conference. José Mourinho seemed unhappy.
    2. José Mourinho was at the press conference. *He* seemed unhappy.

- Introducing discourse markers. For example, transforming (1) into (2):

    1. If FC Porto signs Mehdi Taremi, they should sign Marko Grujić.
    2. If FC Porto signs Mehdi Taremi, they should *also* sign Marko Grujić.

In every previous example, a common theme is how the information conveyed by the text didn't change. Only the text fluidity was improved.

2.3.4 Lexicalisation

With this task, the system starts converting information into natural language. The main concern is then figuring out what words or phrases to use to express the given information. The decisions made at this stage are highly dependant on the domain in question. In the case of sports, which is a well-defined domain and where articles may benefit from some variation, it would be feasible to associate phrases to an event (where an event could have several phrases), and randomly select a phrase for each event in the input. For example, there are multiple ways to express a goal: "to score", "to find the net", "slip past the keeper", so it would be appreciated to use different constructs if multiple goals are scored.

2.3.5 Referring Expression Generation

According to Reiter and Dale [28], Referring Expression Generation is "the task of selecting words or phrases to identify domain entities". There seems to be a similarity to lexicalization, however, they point out that the main issue is providing enough information to distinguish different entities. How an entity is referred to depends on, for example, how many times they have been mentioned throughout the text: if Cristiano Ronaldo is mentioned a second time in a sports article, the system may choose to use a pronoun or an alias (i.e "Juventus forward"). Additionally, the system should consider what is the right amount of information to convey: when reporting a Ronaldo goal, it would be unnecessary to indicate his hairstyle, however unique it may be.

2.3.6 Surface Realisation

The last step pertains to combining all the relevant words and phrases into a well-formed sentence. Some details that require attention are ordering sentence constituents and generating the right morphological forms (including verb conjugation and agreement, which are relevant in languages such as Portuguese). There are several alternatives for this step, and Gatt and Krahmer [12] have discussed the following:

- Human-crafted templates;
- Hand-coded grammar-based systems;
- Statistical approaches.

With the first method, sentences are generated by associating a humanly crafted template with an information type (e.g. events), and replacing the variables with the given data. In the case of soccer, a commonly reported event is a goal – if we have the following template:

> *$player* scored for *$team* in the *$minute* minute.

Table 2.1: List of Journalism NLG systems.

Name	Year	Domain	Surface Realization	Evaluation
PASS [42]	2017	Dutch Football	Templates	Human-based
Chen and Mooney [7]	2008	Sports	Machine-learning	Human-based
Leppänen et al. [15]	2017	Finnish Elections	Templates	Brief remarks
Plachouras et al. [23]	2016	Finance	Templates	Not used
SumTime-Mousam [29]	2005	Weather Forecast	Grammar-based	Human-based

Then the three variables can be replaced with a player name, the player's team name, and the minute in which the player scored the goal. Thus, a possible result of applying this template could be:

Kelvin scored for *FC Porto* in the *92nd* minute.

Templates give full control to the crafters over the quality of the output, and help avoid the generation of ungrammatical structures. Additionally, despite being considered less sophisticated than other alternatives, it has been claimed that template-based systems are not necessarily inferior to other approaches with respect to maintainability and quality of output [40]. It is important to note that this is the approach followed by Prosebot.

Grammar-based systems are a domain-independent alternative to templates, that make their choices based on the grammar of the language being studied. One difficulty arises when choosing among related options, as creating rules with the right sensitivity to context may be difficult to design. More recent approaches make use of probabilistic grammars from large corpora, with the advantage of lowering the amount of manual labor required and increasing coverage.

2.4 NLG Applications in Journalism

In this section, we will take a look at some NLG systems in the field of journalism, presented in Table 2.1 and explained in the following paragraphs.

PASS [42] is a data-to-text template-based system that generates dutch sports reports. One interesting particularity is that it generates two reports per game, with a different tone depending on the team of the reader. It is a difficult task to be performed by a human journalist, but to a computer, it can be done in a matter of seconds, allowing fans to receive a more personally appropriate report. The input data is scrapped from the website *Goal.com*, and the system stores information about participating teams, final score, goal scorers, fouls, and even more sizable data, like past games results, and previous match-ups between the two teams. To design the templates, the MeMo FC corpus [5] was used. It contains match reports created directly for the teams that participated in the match, and therefore are intended towards the supporters of their respective club. This makes it suitable for PASS, particularly because of the intended text personalization aspect. PASS extracted a large number of event categories, and templates per category. Therefore, the authors argue the system produced text with a similar amount of variation to GoalGetter [37].

Chen and Mooney [7] present a commentary system that describes the events in a given match. This already constitutes a big difference when compared to summary generator systems like Prosebot, since in the former almost every event will appear in the text, while in the latter only the most important are considered relevant to be included. Another distinction is that Chen and Mooney [7] use machine-learning techniques, instead of templates, to generate the text. The system has been trained using human-commentated games from the Robocup simulation league[1] and introduced three algorithms to generate commentaries for unseen games. For the content-selection task, it uses a probabilistic approach to understand what type of events (e.g. pass or goal) are most likely to be reported on by human commentators.

Leppänen et al. [15] built an NLG template-based system that generated news articles for the 2017 Finnish municipal elections in Finnish, Swedish, and English. It used simple templates such as "*$entity* won *$value* new seats in *$location*", along with other techniques which have been previously mentioned, such as aggregating sentences with common prefixes, using a referring expression generator to avoid mentioning the same entity multiple times by the same name, and using language-specific morphological rules to assure the sentences are correctly constructed.

Plachouras et al. [23] implemented a system for searching financial data using keywords or natural language. The system analyzes the given query, retrieves the related information, and generates the answer using NLG. So, for example, for the query "India's GDP 2010", the system finds the record that holds the GDP of India in 2010, and returns a text answer, providing other contextual information. Similar to other systems, it contains a module that generates parts of the answer using templates, and for each use case there may be several templates.

SumTime-Mousam [29] generates weather forecast reports from numerical weather prediction data. Its architecture contains much of the tasks presented, including document planning, aggregation, lexicalization, referring expression generation, and finally surface realization. Instead of using templates, special grammar rules were built, since the output would not exactly be in conventional English, but instead in a weather sublanguage. The article focuses primarily on word choice between humans and the system, for instance, choosing between "backing" or "becoming" to describe the change in wind direction.

Finally, there have been instances of popular newspapers using machine-generated news articles, however available information on the systems is scarce. In 2007, the Los Angeles Times launched a blog about homicide reports that would eventually feature machine-written texts following a simple template [45]. In 2014, the same newspaper used *Quakebot* to generate the first article for an earthquake [20]. In 2019, BBC managed to publish a news story for every constituency (a total of 650) in the UK on the general election night, using an NLG tool [18]. Commercially, the most relevant companies are Automated Insights[2] and Narrative Science[3], which have built, respectively, WordSmith and Quill.

[1] https://robocup.org

[2] https://automatedinsights.com

[3] https://narrativescience.com

2.5 Post-Editing and "Machine-in-the-loop" in NLG

The idea of humans and machines jointly writing a sports article is not something that has been done before, according to our findings. However, we did find two related concepts: post-editing and machine-in-the-loop story generation.

Ehud Reiter wrote a blog article [26] based on the question of editing an automatically generated text before releasing it. He presented the technique of post-editing used primarily in the machine-translation field, along with an experiment made with a weather forecast generator [35] where experts would edit the texts before they were sent to clients. The most interesting insights were that analyzing the post-edits helped understand how to improve the system, but at the same time, the post-editing process did not significantly improve the texts, in Reiter's opinion. Finally, he recommended that the best way for a human and machine to jointly write a text would be for the computer to generate a readable and accurate text, and then for the person to make more complex changes.

Additionally, as previously mentioned, there are various text-to-text NLG systems with the purpose of supporting individuals, in areas including machine translation and automatic grammar checking [44, 33, 3]. Another instance of this type of system is machine-in-the-loop story generation. *Machine-in-the-loop* is the reverse of another popular term, *human-in-the-loop*. While in the former a machine acts in a supporting role with the goal of improving Human ability, in the latter humans are actively included in the training of machine-learning models by providing feedback such as labeling examples or suggesting features [8].

The general idea is the following: creative writers that are affected with "writer's block", a condition that affects their creativity and limits their ability to further their stories, use NLG systems that automatically generate sentences from a given input text. Examples of such systems are STORIUM [2], Writing Buddy [32] and Creative Help [31].

STORIUM [2] is based on a gamified collaborative story writing platform with the same name, where the writing process is turned into a game: in each game there is one player that takes the role of narrator, and the rest will play individual characters. The story is composed of high-level scenes that consist of multiple scene entries, where each is written from the perspective of a character, or the narrator. Scenes commonly revolve around challenges, and in order to face them, characters have access to a set of cards that convey properties such as strengths, items, or goals. The story moves forward by introducing new challenges, locations, characters, and cards. To introduce "machine-in-the-loop" story generation, the authors implemented a GPT-2 language model [24] trained on approximately five thousand stories from the STORIUM platform. To evaluate the stories, a web service was created where the model outputs are served to the STORIUM platform. For the user, he only has to press a button to obtain a generated scene entry, that was created using the context of the previous text. He is then able to edit the generated text, by adding or deleting as he may wish.

Writing Buddy [32] is an application where the writer and the system take turns at writing a story. The authors present an experiment where the participant is asked to write a ten-sentence story based on an unlabeled cartoon. Using a simple web interface, the user submits a sentence through a text input. Then, the application will generate a sentence using a language model given the preceding text, and display it below the previous sentence. The user is allowed to edit as he wills before he submits the new sentence, and the third sentence will once again be written by the author. This cycle will continue until the ten sentences have been reached. Note that, after submitting each sentence, it will become locked for edition, and be added to a section that displays the story written until that point.

Creative Help [31] is a web application for writing stories. The user interface is very simple, as users only need to interact with a text area where they will write the story. When the user inputs the phrase "/help/", the application will generate a sentence as a suggestion to continue the story, replacing the "/help/" string. The generated text can then be fully edited, just like the rest of the story. The system will track the changes made to the suggestions to evaluate their contributions to the story. The suggestion generator uses a corpus of approximately twenty million English-language stories, and following a strategy similar to SayAnything [36], will find the most similar match in the corpus for the sentence that appears directly before the "/help/" command and returns the sentence that immediately follows it in the corresponding story. To track the edits made to the suggestions, the application listens for keystroke events in the text area the suggestion appears in, and if it detects that most of the text characters have been removed, then considers it to have been deleted. If not, it will continue tracking it and, if the suggestion remains unchanged for at least one minute, then the system considers it to have been accepted. For the case when the user asks for a suggestion without having written any text, then, since there is no previous sentence for the system to use, it will return a random sentence from a manually written list of introductory sentences.

There are many differences between Prosebot and the former systems. While Prosebot fits the category data-to-text, and is a template-based system, the aforementioned story generation systems are described as text-to-text, and generally use machine-learning models. Perhaps the most important distinction is the goal of the system: since, contrary to creative writing systems, Prosebot aims to inform the reader, creativity plays a lesser role when compared with factual correctness. Even still, we are interested in the evaluation aspect: since the user is able to edit the generated text, there need to be methods to determine the quality of the tool in the eyes of the user. This will be explored in Section 3.2.

2.6 Impact of Automatic Tools in the Newsrooms

The theme of the perception of news journalists towards automatic tools has also been previously explored. van der Kaa and Krahmer [41] conducted a survey on the journalists and news consumers perceived trustworthiness and expertise of automatically generated articles when compared to human-written ones. The authors concluded that journalists perceived their trustworthiness to be much higher than that of a computer, and that they recognized the computers expertise to be higher than the news consumers perceived it to be. The respondents also perceived finance articles to be more trustworthy than sports articles. van Dalen [39] analyzed a collection of blog posts and articles to understand how journalists see their future impacted by automatic tools and presented their conclusions in the form of a SWOT analysis (Strengths, Weaknesses, Opportunities and Threats), which showed that journalists highlighted their analytical skills, creativity and flexibility as important skills defining journalism, but also recognized the automation of routine tasks as an opportunity to perform more in-depth reporting. Kunert [14] has conducted interviews with german software providers and sports journalists regarding the usage of automatic tools and the possible changes in sports reporting. Results show that media outlets believe automatic tools to be helpful for their work, as they can be used to automatically create articles and allow for in-depth reporting. Other outlets edit automated content before publication in order to increase the quality of the articles, although this is done on a small scale. We have based ourselves upon these studies to create our own survey targeting Portuguese journalists, and will compare their results with ours in Section 4.3 and Chapter 6.

Chapter 3

Evaluation Methods

In this chapter, we'll go over methods used to evaluate NLG systems, which we may use to assess the changes we will implement to Prosebot, namely, by addition of templates. Additionally, since the Prosebot platform generates summaries using an NLG system and human post-editing, we will explore how systems that use a similar approach have been evaluated.

3.1 NLG Evaluation

Intrinsic methods evaluate the performance of a system without considering other aspects of the setup. Details such as text quality, fluidity, and readability belong to this category. On the other hand, extrinsic methods evaluate the effectiveness of a system in achieving the desired goal.

Intrinsic evaluation can be divided into two main categories: human-based and automatic. Although automatic metrics such as BLEU [21], METEOR [4] and ROUGE [16] are increasingly popular [43], they are also the target of criticism. van der Lee et al. [43] point out that automatic metrics are uninterpretable and do not correlate with human judgments, but they still note that due to their cheapness and quickness when compared with human-based methods, their usage should not be dropped. On the other hand, human-based methods are also not free of criticism: Gatt and Krahmer [12] noted that there may be high variance in judgments made by different evaluators, thus reducing reliability.

Intrinsic human-based methods mostly consist of presenting a group of people (that could be mere readers or experts in the matter) a collection of texts that may have been generated from the NLG system or by other humans, and making several questions regarding fluidity, readability, accuracy, adequacy, relevance, or correctness [12]. van der Lee et al. [43] found that the most popular rating method is a 5-point Likert scale question. For instance, the statement "The text is easy to read" would have five possible answers, ranging from a positive to a negative stance: "strongly agree", "agree", "neutral", "disagree", and "strongly disagree".

As for automatic metrics, they are used to evaluate the quality of the system's output by comparing it to human output made in the same context, which is essentially evaluating if the machine is able to write like a human. Therefore, a good result happens when the metric reveals a high similarity between the machine and human texts. The popular metrics mentioned above are based on n-gram overlap. N-grams are usually considered as a contiguous sequence of n items in a text, which could be characters, syllables, or even words. We can consider different names depending on the value of n: unigrams, bigrams, or trigrams. For instance, considering bigrams where items are words, the sentence “Lionel Messi scored a brace” can generate the following bigrams: “Lionel Messi”, “Messi scored”, “scored a”, and “a brace”.

On the issue of evaluating the accuracy, Thomson and Reiter [38] have presented a method based on the annotation of errors, where a group of collaborators would analyze a given text, look for errors and categorize them according to the following categories: incorrect number, incorrect named entity, incorrect word, context error, not checkable and other. For instance, in the sentence “Cristiano Ronaldo scored on minute 110 against Italy in the finals of Euro 2014” there are three incorrect named entities (Ronaldo, Italy, and Euro 2014) and one incorrect number (minute 110) – the correct sentence would be “Éder scored on minute 109 against France in the finals of Euro 2016”.

Extrinsic evaluation is the most expensive in terms of time and cost, so, despite claims that it is the most meaningful kind of evaluation [27], it is also rarely used [12, 43]. One of the first extrinsic evaluations of an NLG system was performed by Young [46], who compared multiple algorithms in the task of generating instructional texts, and then asked users to follow the instructions, and measured how many mistakes they committed – this is an example of using task performance as a measure. In a different setting, the GIVE Challenge [13], where NLG systems generated instructions to help users navigate through a virtual world, extrinsic evaluation was carried out by having users play the GIVE game and registering metrics such as the time it took a user to complete the game. Having presented the main categories of NLG evaluation, we will now recall works presented in Section 2.4, and see what evaluation methods they have conducted.

PASS [42] presents users with twenty articles generated by the system for ten soccer matches played in the 2015/2016 season of the Dutch second league (two for each match), and makes simple seven-point Likert-scale questions in order to assess text clarity and fluidity. Text clarity was measured with two questions: “The message of this text is completely clear to me”, and “While reading, I immediately understood the text”. On the other hand, fluidity was measured with two other questions: “This text is written in proper Dutch” and “This text is easily readable”. Additionally, to evaluate the text personalization aspect, the reader is also asked to identify the fans at which the text was tailored to. The results show that in 91% of cases, the readers could correctly make this identification. Furthermore, readers also demonstrated a positive perception of the text quality, with respect to clarity and fluidity.

Chen and Mooney [7], in their sportscasting system, asked human judges to assess the commentary with regards to fluidity, semantic correctness and sportscasting ability, after showing them clips from Robocup games, along with the commentary created by humans and the system. Although, as to be expected, the human commentary received better scores, they still concluded the system commentary to be of reasonable quality.

Plachouras et al. [23] and Leppänen et al. [15] did not conduct any formal evaluation, however in the latter authors argue that overall the generated texts were clear and fluid, despite not always being as fluid as human-written ones. They present two reasons to justify this: fact-ordering is sometimes suboptimal, and the aggregation step may generate unnatural or misleading sentences.

SumTime-Mousam [29] presented participants with texts generated by humans, the system, and a hybrid approach (where a human-written text would be edited to use words that the system would have used), and made several questions regarding background information (i.e how many weather forecasts they read in the space of a year), comprehension (i.e. at various time points, what was the wind speed according to the text) and preference (i.e. which of the articles were the easiest to read, the most accurate and most appropriate). An interesting conclusion was that the participants preferred machine-generated texts over human-written ones.

3.2 Machine-in-the-loop Evaluation

In this section, we will recall previous works in the machine-in-the-loop story generation field presented in Section 2.5, and study how they have evaluated their platforms. Table 3.1 presents the evaluation methods used, and we describe them in the following paragraphs.

STORIUM [2] developed an automatic evaluation metric called User Story Edit Ratings (USER), based on the Longest Common Subsequence (LCS) variant of ROUGE [16]. The idea is that if the user preserves a large amount of the generated text, then he found it useful. They also asked users, at the time of story publication, to fill out a form with 5-point Likert-scale questions to assess relevance, fluidity, coherence, and likability. Additionally, they also performed interviews with 10 STORIUM users to find out the strengths and weaknesses of their model. They concluded that the users are most likely to keep generated text when it is more relevant to the overall story, and that the users found the text to be fluent (which, according to them, is a consequence of GPT-2's pretraining). Additionally, while trying to understand what kind of generated text is preserved by the users, they found that approximately 30% of proper nouns are preserved, which may be a consequence of the suggestions maintaining, for instance, character names which are not usually renamed. However, some names were also created by the system, and user interviews revealed they enjoyed this ability of generating new names.

Creative Help [31] used edit distance, a popular method in computer science to measure the difference between two strings. It measures the number of operations needed to transform one string into another. Intuitively, texts with lower edit distance have higher similarity, which indicates that the generated text was considered useful. They didn't use any human-based evaluation methods. In their experiments, they compared different models for the generation of suggestions that differed on the size of the corpus, and an additional model that did not take the story context into account, thus returning a random sentence. As expected, they found that the random model performed the worst by having a higher rate of edits and deletions, which validates the use of the application as an evaluation platform.

Clark et al. [8] didn't use any automatic evaluation methods. Instead, they performed experiments by asking participants to fulfill three tasks of writing a short story. After each task,

Table 3.1: List of Machine-in-the-loop story generation platforms and their evaluation methods.

Name	**Year**	**Automatic evaluation**	**Human-based evaluation**
STORIUM [2]	2020	Measure string difference	User surveys and interviews
Creative Help [31]	2015	Edit distance	Not conducted
Clark et al. [8]	2018	Not conducted	User surveys and interviews

they were asked to fill a form with 7-point Likert-scale questions, in order to assess creativity, coherence, entertainment, and grammatical correctness in the final product. After completing all tasks, a final survey was presented with Likert-scale questions to assess if the suggestions were surprising, creative, and grammatically correct. Additionally, they conducted an open-ended interview with the participants to find out their experiences and thoughts for future improvements. The participants conveyed mixed reactions to the tool. While some considered the non-editable, sentence-by-sentence structure helped them move the story forwards, others complained they were not allowed to follow their usual writing process. Regarding the usefulness of the suggestions, all of the participants noted they were very random, and while two participants said it helped them write more silly and creative stories, most decided to disregard the suggestions. Additionally, in some cases participants did not consider the suggestions helpful since they would clash with their already defined idea for the story.

Chapter 4

Prosebot platform

4.1 NLG Module

In this chapter we present the architecture of the Prosebot system, which has been in development for approximately five years, undergoing multiple iterations which have been documented at previous dissertations at the University of Porto [1, 34, 30, 9].

4.1.1 Previous Work

4.1.1.1 GameRecapper

Aires [1] developed GameRecapper, a data-to-text, template-based system that generates match summaries from structured data, in Portuguese. The author used a corpus written by *zerozero.pt*'s journalists to manually develop the templates, and analyzed a collection of match reports to design the structure of the summary. It includes an introduction that presents the teams and the match result, a paragraph that narrates the goals of the match, and a conclusion that mentions the classifications for each team after the game. The system achieved good scores with regards to intelligibility and fluidity, although in some cases where matches had a large number of goals, the text would seem repetitive due to a low variety of goal templates. It was also made a comparison to human-authored texts regarding completeness and readiness, and it was concluded that the system would need to report on other relevant events besides goals (such as sent-offs), and that it would be appropriate to be used as a draft generator for journalists to complete.

4.1.1.2 Statistical Models

Soares [34] developed a data-to-text system for the generation of match summaries that, unlike GameRecapper, used a statistical approach for the surface realization step. The author created a corpus after analyzing a collection of news reports, extracting sentences and delexicalizing them in order to generalize them for other matches. The sentences were grouped by four categories: introduction, goals, sent-offs and conclusion, and were also attributed a sub-type (e.g. one introduction sentence could have a “home team wins” type and be used only on that occasion). Then, language models were trained on this corpus and generated delexicalized sentences. A user interface was developed, that allowed users to get sentences regarding a match after selecting the category, after which the system would lexicalize the sentences with the match information and return a complete phrase. The system was evaluated by asking users to rate sentences with regards to intelligibility and completeness, and although the scores were overall positive, there were also some poor results.

4.1.2 Prosebot

Ribeiro [30] presents improvements made to Prosebot, a template-based data-to-text system that was adapted from GameRecapper in the PHP programming language. The system can generate football match summaries in four languages, using a collection of templates that were written after analyzing news articles created by journalists. The system makes use of domain data to provide information regarding the teams and players, as well as linguistic and grammatical functions which may, for instance, convert numbers into their cardinal or ordinal form, and assert the coherence of the generated text. The structure of the texts is similar to that of GameRecapper, including an introduction that presents the teams and match result, a paragraph that describes relevant events from the match (goals, penalties, sent-offs and occasionally substitutions), and a concluding paragraph that mentions changes in classification for both teams. Prosebot can also attribute a score to the texts it generates with respect to three metrics: average sentence length, lexical diversity, and variability of used entity names. Finally, Prosebot was evaluated by presenting participants with a questionnaire regarding the fluidity and intelligibility of the text, and another questionnaire that asked respondents to evaluate the completeness and readiness of the text (which could be written by journalists or Prosebot). Results showed that Prosebot could generate correct and fluent texts, but when compared to journalist-produced articles, scored lower on readiness to be published online.

4.1.2.1 Evaluation Methods for Text

Correia [9] focused his work on developing a system that could evaluate both human-authored news reports, as well as computer-generated match summaries. This system has an API that can retrieve information, metrics and readability scorings about a text. The author also implemented an interface that lets users create a match summary by selecting events and information from a football match and retrieving Prosebot's generated text, which he could then post-edit. This interface also includes a metrics report using the API mentioned above. The journalists from *zerozero.pt* were inquired about the developed systems, and classified the implemented metrics as important for their work, and also gave positive feedback regarding the user interface, finding it helpful for the production of match reports.

Type	**Information**
General	Competition, fixture, datetime, stadium, city, final score, teams, goal scorers
Match events	Assists, regular goals, own goals, free kick goals, penalty goals, penalty misses, yellow cards, red cards, substitutions
Previous results	Number of consecutive matches each team won/lost/drew
Players and managers	Name, position, number of goals scored and assists given in the match, team, number of goals scored in the season, country, number of consecutive matches with goals scored
Match and team statistics	Total shots, shots on target, passes, fouls suffered and committed, corners, offsides, context for goals (zone, type of play and body part)
Season statistics	Best and worst values obtained in a current season's match for the statistics mentioned above
Team streaks	Consecutive matches without losing, without winning, without conceding goals, with goals scored
Post match	Team classification, next games for each team

Table 4.1: Data stored from the API request to *zerozero.pt*.

4.1.3 Data Source

Unlike previous systems that scraped data from *Teletext* or *Goal.com*[1], namely Theune et al. [37] and van der Lee et al. [42], Prosebot relies on match data from *zerozero.pt*'s database, served through an API. This API returns information about the match, including the participating teams, competition name, final score, goal scorers, previous results, and players that were carded. In addition, *zerozero.pt* also supplies data about the game's events and players' and teams' statistics which are listed in Table 4.1. This allows to include interesting remarks along the text, referring for example if a player has scored a significant amount of goals, or if he is in a "hot streak" of matches with goals scored. However, there is still data that is not being used at the moment, which could be included in future versions of the system and make the generated text more complete.

4.1.4 Template Design

In the latest version of Prosebot, the templates have been designed in collaboration with *zerozero.pt*'s engineers and journalists. Templates are stored in JSON files, grouped by categories, and for each template there is the content and a condition. The condition is written in a way that can be parsed and evaluated by Prosebot, and will determine if the content is included in the text or not. This approach has been particularly useful in the effort to increase the variety of the templates and make the texts feel less repetitive. Using a human-friendly interface, one of *zerozero.pt*'s journalists has been editing and adding new templates to the system, in an almost fully autonomous way. Table 4.2 includes the most relevant templates used by the system.

[1] https://goal.com

Section	Category	Variants
Title		All-purpose, 2+ goal difference, 4+ goal difference, home team won, away team won, game ended in draw, match ended with penalties
Subtitle		All-purpose, favourite team lost, decisive goal in the last minutes, no goals scored, many goals scored
Small text		Final score, starter and benched relevant players
Introduction		Final score, previous results, best player
Events	Goal	All-purpose, first goal, only goal, own goal, hat-trick, poker, second goal, last goal, goal drew the match, goal increased/decreased the goal difference
	Substitution	Gamechanger player is subbed in, relevant player is subbed in/out
	Missed penalty	Goalkeeper saved, penalty taker missed
	Red card	Direct red card, accumulation of yellow cards
Debriefing		Post-match classification, next games, match stats
Curiosities	Stats	Best/worst result of the season for the team, best/worst overall result of the season
	Streaks	Increased or broke a sequence of matches

Table 4.2: Available templates.

4.1.5 Content Selection and Document Structure

The document is mostly structured in a way that is similar to the structure of news articles that are published in *zerozero.pt*, with some exceptions. There are seven main sections:

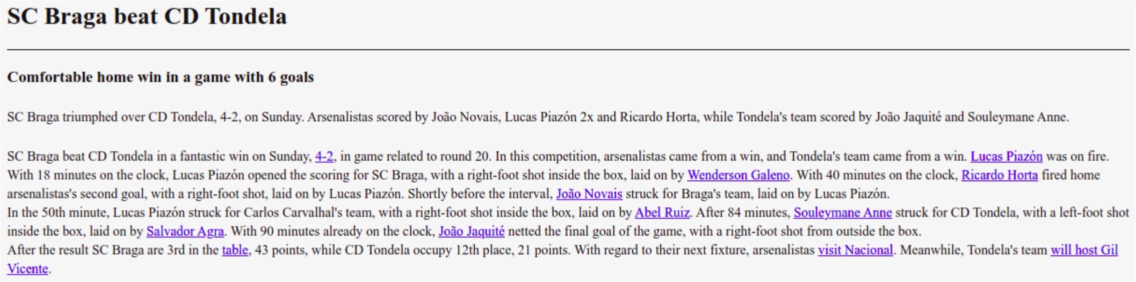
Title Sums up the result of the match (win or draw)

Subtitle Briefly characterizes the result depending on the goal difference, number and context of goals scored and participating teams

Small text Informs on the match result, the goal scorers, the number of red cards shown and relevant players

Introduction Informs on the match result, the teams' form prior to the match (e.g. consecutive wins, losses or draws), the best player of the match and a relevant curiosity about the match. This curiosity will appear if some relevant record was broken, including the number of shots, shots on goal, or corners taken.

Events Informs on the most relevant events that happened during the game, most notably goals, red cards, substitutions, and missed penalties.



SC Braga beat CD Tondela

Comfortable home win in a game with 6 goals

SC Braga triumphed over CD Tondela, 4-2, on Sunday. Arsenalistas scored by João Novais, Lucas Piazón 2x and Ricardo Horta, while Tondela's team scored by João Jaquité and Souleymane Anne.

SC Braga beat CD Tondela in a fantastic win on Sunday, 4-2, in game related to round 20. In this competition, arsenalistas came from a win, and Tondela's team came from a win. Lucas Piazón was on fire. With 18 minutes on the clock, Lucas Piazón opened the scoring for SC Braga, with a right-foot shot inside the box, laid on by Wenderson Galeno. With 40 minutes on the clock, Ricardo Horta fired home arsenalistas's second goal, with a right-foot shot, laid on by Lucas Piazón. Shortly before the interval, João Novais struck for Braga's team, laid on by Lucas Piazón.
In the 50th minute, Lucas Piazón struck for Carlos Carvalhal's team, with a right-foot shot inside the box, laid on by Abel Ruiz. After 84 minutes, Souleymane Anne struck for CD Tondela, with a left-foot shot inside the box, laid on by Salvador Agra. With 90 minutes already on the clock, João Jaquité netted the final goal of the game, with a right-foot shot from outside the box.
After the result SC Braga are 3rd in the table, 43 points, while CD Tondela occupy 12th place, 21 points. With regard to their next fixture, arsenalistas visit Nacional. Meanwhile, Tondela's team will host Gil Vicente.

Figure 4.1: Sample of text generated by Prosebot in English.

Debriefing Shows the classification of each team after the match, the next match for each team, and may also include a relevant curiosity for each team.

Curiosities This section appears separately from the text, and simply displays a list of curiosities about the match or participating teams. This list includes sequences, such as winning streaks or sequences of matches with goals scored (e.g. “Team A broke a sequence of 5 winning games”). The list also contains statistics about the match or the teams when it comes to beating records for certain types of events, including, for example, beating the record of shots taken in a single match in the competition.

Each section, excluding the last that is kept separate, is shown in a different paragraph. Every text generated by Prosebot will share this structure, however the type of content may be different depending on the match and the teams. For example, if a match is for a competition such as the playoffs of the Champions League, then the text will not include the classification of the teams because it wouldn’t make sense in the context of the competition.

4.1.6 Algorithm

The system starts by parsing the template, grammar, and entity manager files according to the selected language. Currently, Prosebot supports four languages: Portuguese, Brazilian Portuguese, English and Spanish, and a sample for a text generated in English is included in Figure 4.1. As previously said, the template files contain the textual content that will appear in the final result, according to the validity of their conditions. The grammars include language specific utility functions to help print numbers in the ordinal or cardinal form, and also functions to include articles depending on the gender and number of the subject, which is particularly important in the Portuguese language. As for the entities managers, they are mostly used to return names for entities such as players and teams, depending on the information that is stored and the names that have already been used. For example, depending on the team, we may know its name, its city, its coach and its nickname. So, throughout the text, the manager may return different results for the team’s name - in the case of FC Porto, it could return “Sérgio Conceição’s team” or “Porto’s team”.

After loading the necessary files, it collects the data from the API request into the appropriate classes. There is a central class representing the match, which is the gateway for everything related to the match, teams, and players. Once everything is stored, it will generate the text for each of the paragraphs presented in the previous section using the same strategy, with some exceptions for the events paragraph. This process is mostly recursive, due to the nature of the templates: a template may include not only text but also may interpolate functions and variables, as well as *other* templates. The first step is to select the root category to start the process. From this category, a subset of templates will be filtered depending on the validity of their conditions. Additionally, since a condition has an associated weight, the templates will also be filtered according to the weight, so that only the most specific templates may be used. For example, looking at two descriptions of a goal: "After 20 minutes, Cristiano Ronaldo struck for Juventus" and "After 20 minutes, Cristiano Ronaldo scored Juventus's only goal of the game". Since the first template can be used for any kind of goal, it has no condition, while the second template is only used if the player's team scored a single goal, then the latter template has more weight and the former will be filtered out. After having filtered the templates, the system only needs to pick one of them to use. Finally, a recursive function is called that will parse the template's text, interpolates the variables and repeats the same process if other templates are found.

As previously mentioned, the process for the events paragraph is slightly different: since we want to narrate the events of the match, there doesn't exist a root template, therefore the system loops through the match events, and using the process described above generates the text for each event and appends it to the result.

4.2 Improvements to the NLG Module

In this section we present improvements we have made to the Prosebot system. The most relevant changes are described in each subsection, which may use illustrative tables or snapshots from the text generated by Prosebot. We have focused on fixing bugs that prevented the generation of a summary or caused parts of the summary to be nonsensical, but also strived to include new and interesting information to the text, while working closely with *zerozero.pt*'s journalists to improve the diversity of the templates.

4.2.1 Mention relevant players

First, we need to define what relevant players are. This is dependant on two main factors: the type of teams that are playing (national teams or clubs) and the country to which the text is directed at. To better understand this concept, let's look at Table 4.3.

The first match is between English clubs, was generated in Portuguese and is therefore directed towards the Portuguese audience. So, the system will tag Portuguese players as relevant. As for the second match, it involves German teams, was generated in English which means it is direct towards the English audience. As such, any English players will be considered relevant. In the third match there is only one relevant player: Seferovic. Since the game is between national teams

Table 4.3: List of matches and their relevant players.

Match	Language	Country	Type	Relevant Players
Manchester City vs Wolverhampton Wanderers	Portuguese	Portugal	Clubs	João Cancelo, Rúben Dias, Bernardo Silva, Rui Patrício, Nélson Semedo, Rúben Neves, João Moutinho, Pedro Neto, Vitinha
Borussia Dortmund vs RB Leipzig	English	England	Clubs	Jadon Sancho
Switzerland vs Finland	Portuguese	Portugal	National Teams	Haris Seferovic
Belgium vs Denmark	Spanish	Spain	National Teams	Eden Hazard, Thibaut Courtois

and the text was generated in Portuguese, then the system will try to find any players that play for Portuguese teams, and in this case Seferovic plays for SL Benfica. In the last match which was generated for the Spanish audience, there are two players that play for Spanish teams: Hazard and Courtois, that both play for Real Madrid.

Thus, we have introduced the following rule: when the game involves clubs, the system will look for players whose nationality is the same as the audience's, but play for a foreign team. Regarding national teams, the system will tag players that have a different nationality than the audience but play for a team from their country. At the moment, these players are mentioned in the introduction paragraph, in a section that indicates the starter and benched players. They will also be mentioned in the event of a substitution. The following text is extracted from a summary generated in Portuguese for the Manchester City vs Wolverhampton Wanderers match:

> Entre os titulares estiveram João Cancelo, Rúben Dias, Bernardo Silva, Rui Patrício, Nélson Semedo, Rúben Neves, João Moutinho e Pedro Neto. Vitinha não saiu do banco.

4.2.2 Extend compatibility to different formats

The previous version of Prosebot assumed that every given match would be for league competition, such as the English Premier League and would generate non-sensical text for matches of different competitions, such as the knockouts phase of the UEFA Champions League. Table 4.4 displays a comparison between the previous and current version of Prosebot when asked to generate a summary for a semi-final match of the Champions League. As we can see, the previous version makes two wrong assumptions: the first is that there is a general classification for the competition, and the second is that both teams will play again for the same competition after this match. Since the given information does not correspond with the template, the resulting text is not correct. The

Table 4.4: Final paragraphs of summaries generated for the Real Madrid vs Chelsea match.

Previous Version	**Current Version**
After the result Chelsea are 0th in the table, 0 points, while Real Madrid occupy 0th place, 0 points. In their next fixture, blues visit Manchester City. Meanwhile, merengues .	With this result, Chelsea see Real Madrid off the competition. In relation to forthcoming matches, blues visit Manchester City.

current version fixes these issues by displaying what team advances to the next stage, and only showing the next match for that same team.

4.2.3 Curiosities: streaks and records

Statistics were not particularly explored in the previous version of Prosebot. It would only store a player's number of goals scored and number of consecutive games with goals scored. So, to take advantage of the amount of statistics stored by *zerozero.pt* we have decided to follow two avenues: the first is to display when a record related to a certain statistic (e.g. shots or corners taken) is broken for the current season. There are multiple types of records to consider: they may be positive (e.g. most shots taken) or negative (e.g. least corners taken); they may be related to the match as a whole (e.g. match with the most fouls) or only to a team (e.g. team had the biggest ball possession in the season); and the record may be personal or global, depending on if other teams have better records.

The second avenue is to display streaks that the current match may have extended, or broken. In this case, streaks are sequences of consecutive matches where a team has, or has not, achieved a certain feat such as scoring a goal.

Both the records and streaks are displayed in a separate section to the text, in the form of a list. Some of these curiosities may also be inserted in the text, depending on the fixture of the match. For instance, a match played on the second fixture of a league will likely generate a long list of curiosities, which may not be worth reporting. However, a curiosity generated on the tenth fixture will be interesting to report. Table 4.5 lists the currently available curiosities.

4.2.4 Extend support to other sports

The Prosebot system was originally developed for the sport of football. However, since *zerozero.pt* also provides coverage for other sports such as futsal and hockey, we decided to investigate its behaviour when given a game for one of these sports. Unsurprisingly, when given a futsal match, it failed to generate a text. The issue arises from the fact that Prosebot uses the player's position to refer to him: for instance, if Cristiano Ronaldo scores two consecutive goals, in the first goal he will be referred to by his name, and in the second he will be referred to by his position of a forward. To achieve this, Prosebot needs to have the positions locally available, and since futsal

Table 4.5: Currently available match curiosities.

Records	Streaks
Ball possesssion	Games won
Shots taken	Games lost
Total passes	Games drawn
Fouls commited	Games without winning
Fouls suffered	Games without losing
Corners taken	Games with goals scored
Goalkeeper saves	Games without conceding goals
Shots on goal	
Offsides	

has a different set of positions than football, it failed. After this information was added, the text was successfully generated.

Since the sports of football, futsal and hockey share many similarities, we didn't notice any problems with the text, at first. However as we tested with more matches we noticed two further problems: when reporting on events, the templates may say that, for instance, a goal happened at the end of the first half (by checking if the associated minute is bigger than 40). This works for football but these other sports do not have the same half duration, thus this kind of templates were being wrongly used. To fix this, we have introduced `minute_ratio`, a variable that puts the minutes in relation to the duration of the match, regardless of the sport. Another issue is that in hockey there are tournaments which are played in a best-of-N format, and so a team advances when it has won $(N+1)/2$ matches. Since the API was not compatible, the text would incorrectly report that one of the teams advanced. While the API is not fixed, this sentence has been temporarily removed.

4.2.5 Head-to-head information

One of the principal talking points for match previews is the head-to-head results between the teams. For instance, if two teams meet and one has a very good track record against the other, then the former will in most cases be more favored to win the match. We thought it would be interesting to include head-to-head information in the text, since it can provide more context and importance to a match result. In a first approach, we have added two templates to the curiosities section, that will display if the winning team has won its first game against the opponent, or if the losing team has never won against its adversary. Of course, in the future there may be more templates added, and the text may also be included in the main section of the summary.

4.2.6 Bug fixes

In an effort to prepare Prosebot for production, *zerozero.pt* developed a script to automatically publish summaries on the website, using matches from past seasons, different competitions and even different sports such as Futsal. At the moment, approximately a hundred and fifteen thousand summaries have been published using this script. Throughout this process, Prosebot has failed to generate summaries for multiple matches, mostly due to misinterpretations of the data returned from the API, assuming that certain attributes are always present, when sometimes they are not. These issues have been promptly fixed, allowing Prosebot to generate summaries for as many matches as possible.

4.2.7 Addition of templates

One of the main aspects that required attention was the diversity of the templates. Since there were a low number of templates for each category, Prosebot generated articles that were very similar to each other, and reading them would start to feel repetitive. To tackle this problem, we have collaborated with journalists from *zerozero.pt*, who despite having minimal experience with programming languages, can contribute to the templates using an editor that includes annotations for the existing conditions, which helps decrease the learning curve for the journalists so that they contribute in an almost fully autonomous manner. For example, one of the templates for the event of a goal is the following:

> **Text**: {template.time}, {scorer.name} opened the scoring for {team.name}, {template.goal_type}, {template.assisted}.
>
> **Condition**: match_goal==1 && match_goals>1
>
> **Annotation**: First goal in a match with several goals

4.2.8 Various changes

We have presented the most relevant improvements made to the Prosebot system, however there were other smaller changes we would like to mention and will go over in this section.

Coaches sent off Since the system already reports when a player sees a red card, and it also stores the coaches for both teams, a small tweak was added to also report when a coach is sent off.

Matches on neutral ground The vast majority of matches are played in stadiums belonging to one of the playing teams, and as such the system has adopted the terms “home team” and “away team”. Thus, in some parts of the text, we may encounter phrases such as “Comfortable home win” or “In the next fixture, Manchester City will host Chelsea”. However, there are some games that break this rule, with the most popular being the Champions League final, which is usually played in a neutral field. In these cases, while the underlying system may still use the terms “home” and “away” to differentiate the teams, it doesn’t make sense to use them in the text, and we have corrected this error.

Crowd Another trivial curiosity to add to the text is the number of spectators that watched the game at the stadium. We have stored the crowd attendance and the stadium capacity, so that we can have phrases like “20456 spectators attended the match” or “Porto beat Benfica, 2-1, in front of a sold-out stadium”. For the latter phrase, we only need a condition that checks on the ratio of spectators and capacity, such as `crowd_ratio >= 0.9`.

Team dominance One of the templates for the title is the following:

> **Text**: {winner.name} with absolute dominance.
>
> **Condition**: final_score_diff>2

The condition says that the template would be valid if the winning team scored at least 3 goals more than their opponent. However, the amount of goals can not be the only factor taken in consideration to decide if one team dominated. Thus, we have introduced a condition that will check if the winning team had more than 65% ball possession, as well as more than 30% shots taken than the other team.

Decisive goal in stoppage time We have introduced a condition that will check if a team was able to clinch a result in the stoppage time of the second half of the match. This can mean two things: the team was losing by one goal and was able to draw, or the team won the game by breaking the tie, both in the final minutes of the game. This condition may be used, for instance, to create titles such as “Porto beat Benfica in the final moments” or “Porto clinched a draw in stoppage time”.

Contribution from the best player When Prosebot finds one player to have had a great exhibition, which it does by computing the players’ impact score using their goals and assists and checking if the score is high enough, it will highlight that player in the first paragraph, with phrases such as “Cristiano Ronaldo was on fire”. In order to add more context to this, we have included a template that will explain the player’s contribution. Thus, the previous phrase would transform to, for example, “Cristiano Ronaldo was on fire, after scoring 3 goals and 1 assist”.

Strategy for mentioning a team As mentioned in Section 4.1.6, Prosebot will avoid repetition by mentioning a team using different phrases. The initial strategy was to store a list of the team’s name, nickname, city and coach, and sequentially return a new name each time the team is referenced in the text. The system will store in cache the last used value for each team, and this cache is reset when the system generates a new section of the text (e.g. moving from the subtitle to the introduction), so that in the beginning of each section the teams are always referred to by their proper name. The problem with this approach is that sometimes some of the available names would not be used. To fix this issue, we have introduce randomization in the aforementioned list: each time the system generates a new section of the text, the list is randomized with exception of the team name which is always first.

Teams from the same town Regarding team references, there was also an issue with mentioning the teams by their city name. In games such as Benfica facing Sporting, the system could use the term "Lisboa's team" for both of the teams, thus making it difficult to distinguish between them. In these cases, where both teams are from the same town, this name will now be disregarded.

Number of goals scored in a season Prosebot included a template that was meant to be displayed after reporting a goal, and tell the amount of goals scored by the goal scorer in the season if it was a multiple of 5. However, due to a bug, this template was always being included, making the text seem too repetitive. We have introduced a condition that fixes this issue and adds an additional condition: if a player scored 3 or more goals, the system will report the number of season goals after the last one.

Grammatical agreement in number When generating a Portuguese summary for a SC Braga match, also known as "arsenalistas", we noticed the following sentence: "os arsenalistas recebe o FC Porto". The problem lies with the fact that "arsenalistas" is plural and "recebe" is singular, thus the two terms are not in agreement. This bug is due to the fact that Prosebot uses different names to refer to the same team, therefore when choosing between the singular or plural of the "receber" verb, the system was using a different name than "arsenalistas". The system used an entity cache to deal with this problem, however that cache was only used inside a template, and since the previous phrase is the result of two templates, the cache did not work. Thus, we have made it so that, in situations like these, the templates will share the same cache, and the issue was fixed.

Matches that end on penalty shootout Initially, Prosebot was disregarding matches that end on penalties, because it always reported the result at full-time. This would generate incomplete summaries in instances where a team may win in full-time, but lose after a penalty shootout. So, we are now storing the penalties score, if it exists, and will display it alongside the full-time score.

Teams play each other again In the final sentence of the main section of the summary, Prosebot reports on the next matches for each of the teams. In some situations, were the teams will face against each other again immediately, the text would be redundant: "With regard to their next fixture, Manchester's team will host Paris SG. Meanwhile, Paris's team visit Manchester City". This redudancy has been fixed, and now Prosebot would only say "the teams will meet again on Manchester City's field".

4.3 Survey to the *zerozero.pt* Newsroom

To design the platform, we conducted a survey inside *zerozero.pt*'s newsroom in order to receive contributions, and we took the opportunity to understand how the journalists perceive the impact

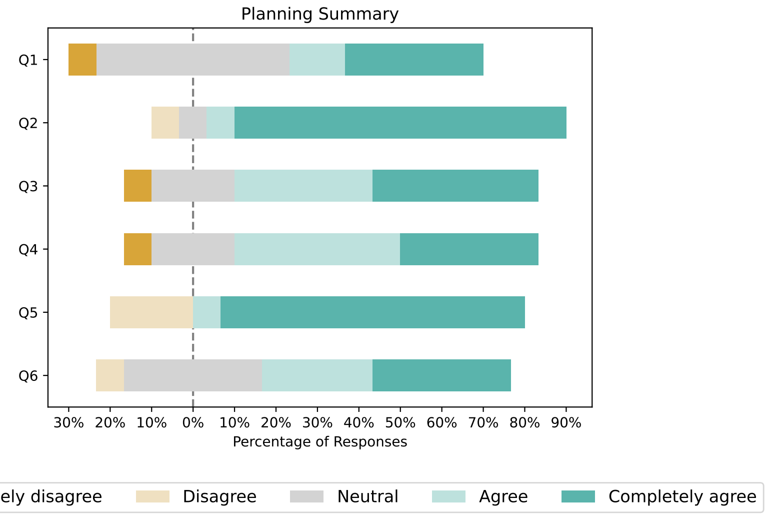


Figure 4.2: Summary of results from the planning section of the survey.

of automatic content generation tools in their work. A total of fifteen (15) journalists responded, and no personal information was collected as the survey was fully anonymous.

The survey consists of 5-point Likert scale and open text questions, and includes two sections: the first section aims to understand the journalists' opinion on the new platform, and therefore asks questions such as "Should the published texts reference Prosebot's envolvement?" or "What name should be attributed to the texts published in the platform?". The second section intends to find the journalist's perception towards the appearance of automatic tools in the newsroom, and thus includes questions such as "How important is Prosebot in the multiple steps of writing a news article?" or "How does the role of sports journalists change with the increased use of automation software?". The full survey can be consulted in Annex B.1.

Table 4.6: Labels for the questions from the planning section of the survey.

ID	Question
Q1	Allowing the community to create sports content is beneficial.
Q2	The texts should be approved before being published.
Q3	The published texts should reference the involvement of Prosebot in its generation.
Q4	The published texts should mention the authoring collaborator.
Q5	The news articles published by *zerozero.pt* and the texts published in the new platform should be differentiated, that is, presented in different ways.
Q6	The possible increase in published texts influenced by this platform would have a positive impact in the community.

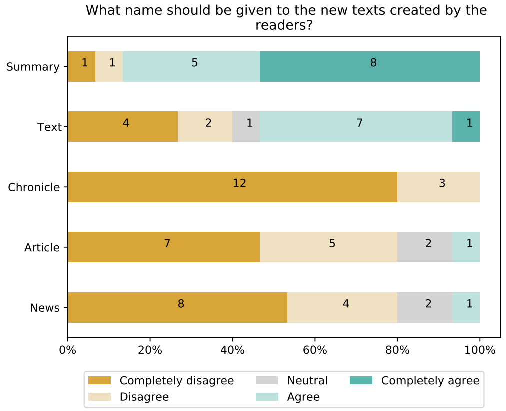


Figure 4.3: Opinion on the name that should be used to refer to Prosebot's texts.

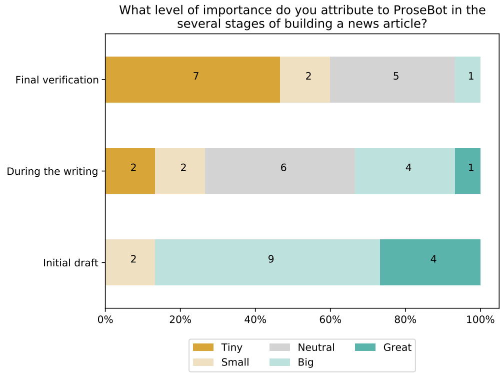


Figure 4.4: Perceived importance of automatic tools through the various stages of news production.

4.3.1 Planning of the Platform

Regarding the platform, Figure 4.2 displays a summary of the results of the survey, with the labels presented in Table 4.6. In the first question, we wanted to see if the journalists agreed with the core concept of the platform: letting the community participate in the creation of sports content. While 46% of the respondents agree that the concept is beneficial, the same percentage seems to have a neutral standpoint.

Regarding the publishing of the summaries, 86.7% of the journalists agree that the summaries should be approved before being published [Q2] – however, the product managers have the opposite opinion. As seen in Section 4.4, *zerozero.pt* has a collaborative spirit and relies on contributions from the community to complete various types of information. Therefore, we intend to keep this approach with the Prosebot platform, while still keeping in mind the warnings of the journalists with regards to journalistic integrity, which is why the content will be moderated after it is published, by removing texts that are deemed unfit to be displayed on the website.

We asked how the summary should be signed [Q3, Q4], and 73,3% of the journalists agree that both the collaborator and Prosebot should be referenced as co-authors of the summary. This way, transparency is preserved and readers are fully aware of who was involved in the writing of the text.

Another relevant topic is the presentation of the summaries published by the collaborators in comparison with the news articles created by the newsroom, and 80% of the respondents agree that they should be differentiated [Q5]. In a follow-up question, the respondents again underlined that there should be a disclaimer in Prosebot's summaries with the identification of the tool and the author, but also suggested design differences such as using another text font, colors, or framing. Additionally, it was also suggested to display Prosebot's summaries in a different section of the website.

Before working on the survey, we were considering the term "chronicle" to refer to the new texts. However, we thought it would be relevant to ask for the journalists' opinion in this regard and we found a surprising conclusion, as seen in Figure 4.3: “chronicle” (“crónica”) was their least favourite term while “summary” (“resumo”) was found the most adequate. After further inquiry, we understood that chronicles are usually more opinion-based, and are actually already part of the content that *zerozero.pt* publishes. Since the texts, despite not being published by journalists, are still meant to be non-biased and factually correct, this term was seen as unfit. One of the respondents also suggested the term “rescaldo”, which could be translated as “after-game”.

Finally, we wanted to ask if the possible increase in published texts inclueneced by the platform would have a positive impact on the community, and once again the journalists gave a positive reaction, with 70% of respondents agreeing with the sentiment [Q6].

4.3.2 Impact of Automatic Tools in the Newsroom

In the second part of the survey, we asked questions regarding the usage and impact of automatic tools such as Prosebot in the newsroom. The results are summarized in Figure 4.5, and the labels are presented in Table 4.7. Regarding the comparison in qualities between journalists and automatic tools, we have concluded that most respondents see themselves as more flexible and analytical than automatic tools, however they do realize that automated content can be produced faster, and for a wider audience (in the case of football, allowing smaller competitions to receive textual content). Although some journalists showed their concern towards the possibility of fewer journalists being needed in the future, still 66.6% of the respondents believe that is not going to happen. These results are generally in line with the conclusions from van Dalen [39] – we only found a contradiction, since unlike what’s underlined in the previous study, most of *zerozero.pt*’s journalists do not believe they will be replaced. This may be a result of the newsroom’s previous contact with Prosebot, since the tool has been in development for approximately five years, and in this period the journalists have been asked to collaborate multiple times. Additionally, *zerozero.pt*’s founders have told us that their goal with Prosebot is strictly to help journalists, not replace them.

We have also asked on how important automatic tools are in the multiple stages of news production, and it seems that as the process of writing the article advances, the perceived importance of Prosebot and other automatic tools decreases, as seen in Figure 4.4. This actually goes in line with the usage we intend to give Prosebot in our platform. It will provide a draft with the most relevant information regarding the match, saving the author time trying to analyze data.

The topic of possible ethical problems stemming from the use of automatic tools was also inquired upon, and we can notice that 40% of the respondents agree these problems may exist [Q10]. In a follow-up question, the journalists mention the issues of eventual bias, truthfulness and independence. With regards to the veracity of the content, we have indeed verified that at times, the summary generated by Prosebot may include content that is not factually correct, which is due to wrong information being sent from the API.

In another open-ended question, we asked how the introduction of these tools could change the role of a journalist, and most respondents highlighted the time freedom to work in other types of content such as investigative work, although 3 answers don’t foresee any significant change. In

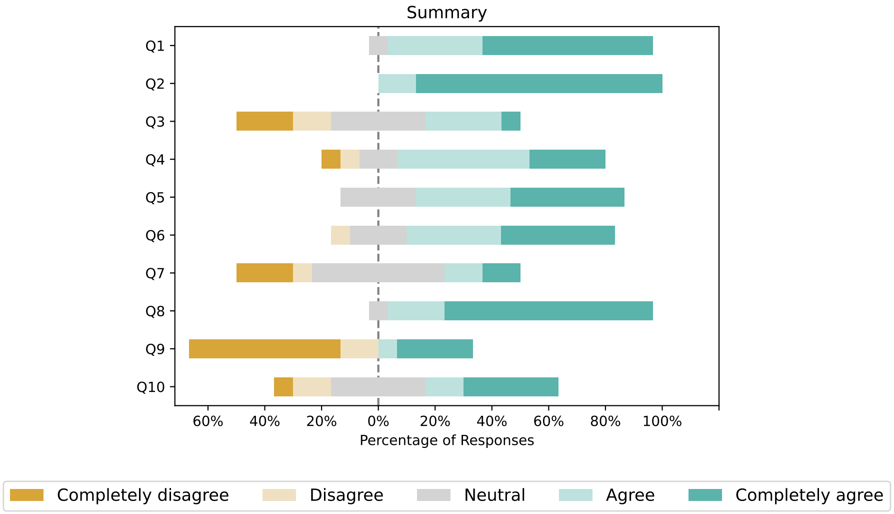
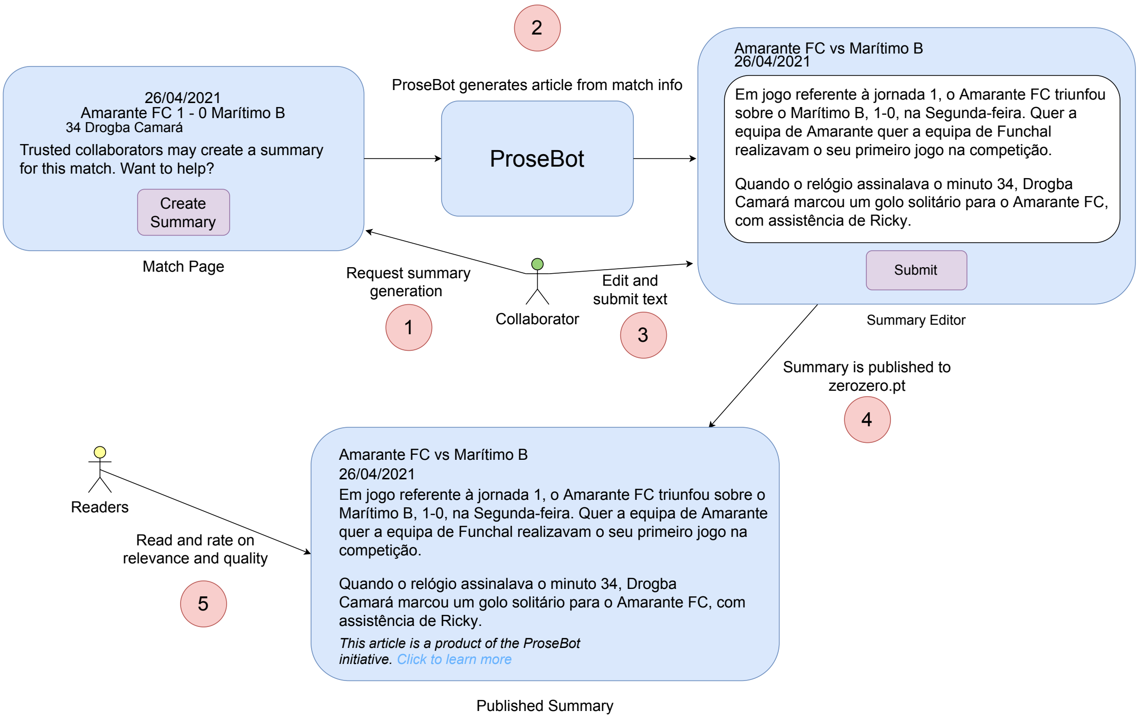


Figure 4.5: Summary of results from the survey.

Table 4.7: Survey question labels.

ID	Question
Q1	The flexibility of journalists, in contrast with automatic tools, allows them to cover more personalized content, such as breaking news.
Q2	The analytical skills of journalists allows them to provide a deeper coverage than automatic tools.
Q3	Automatic tools can provide a wider coverage than journalists.
Q4	The content produced exclusively by journalists has a greater cost than the one created by automatic tools.
Q5	Automatic tools can produce content faster than journalists.
Q6	The automatization of news articles will free journalists to investigate and produce more detailed news articles.
Q7	The competition of automatic tools will result in the improvement of the content produced by journalists.
Q8	Automatic tools allow the coverage of smaller audiences, such as matches for local leagues.
Q9	The increase in usage of automatic tools will lead to fewer journalists being needed.
Q10	The usage of automatic tools brings ethical questions, such as transparency and copyright.


2
ProseBot generates article from match info
26/04/2021
Amarante FC 1 - 0 Marítimo B
34 Drogba Camará
Trusted collaborators may create a summary for this match. Want to help?
Create Summary
Match Page
ProseBot
Amarante FC vs Marítimo B
26/04/2021
Submit
Summary Editor
Request summary generation
1
Collaborator
Edit and submit text
3
Summary is published to zerozero.pt
4
Readers
Read and rate on relevance and quality
5
Amarante FC vs Marítimo B
26/04/2021
Published Summary

Em jogo referente à jornada 1, o Amarante FC triunfou sobre o Marítimo B, 1-0, na Segunda-feira. Quer a equipa de Amarante quer a equipa de Funchal realizavam o seu primeiro jogo na competição.

Quando o relógio assinalava o minuto 34, Drogba Camará marcou um golo solitário para o Amarante FC, com assistência de Ricky.

This article is a product of the ProseBot initiative. *Click to learn more*

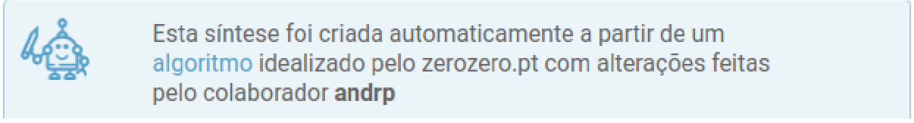
Figure 4.6: Rich picture of the Prosebot platform.

the last question, we inquired about the characteristics of the journalistic work that automatic tools would be unable to replicate. The respondents understand that these tools do not have access to the full context of a match, and do not have the sensitivity, creativity, and subjectivity to deal with the game's nuances. One particular answer even states that "*the sensitivity, in-depth analysis and interpretation of each journalist is irreplaceable*".

4.4 Prosebot Integration in *zerozero.pt*

zerozero.pt encourages community participation, by allowing users to register on the website, answer to polls and comment on news and match pages. Members can also become collaborators by submitting information about teams, players, coaches or stadiums. There is also a particular set of members called "edition collaborators", which are able to submit match results for certain competitions. These contributions are particularly useful for *zerozero.pt* to cover matches for a great number of competitions, ranging from the upper echelon to local leagues. To become one of these collaborators, one needs to submit an application to *zerozero.pt* by explaining how knowledgeable they are about the respective competition, and await for a manual approval. There are currently approximately thirty thousand "edition collaborators", which may, through the Prosebot platform, participate in the production of match summaries using an initial draft generated by Prosebot.

The workflow of the platform, which is presented in Figure 4.6, is straightforward – when a collaborator visits a match page, he may choose to create a summary by clicking the respective



Esta síntese foi criada automaticamente a partir de um algoritmo idealizado pelo zerozero.pt com alterações feitas pelo colaborador **andrp**

Figure 4.7: Disclaimer shown on Prosebot's summaries, referencing Prosebot and the author.

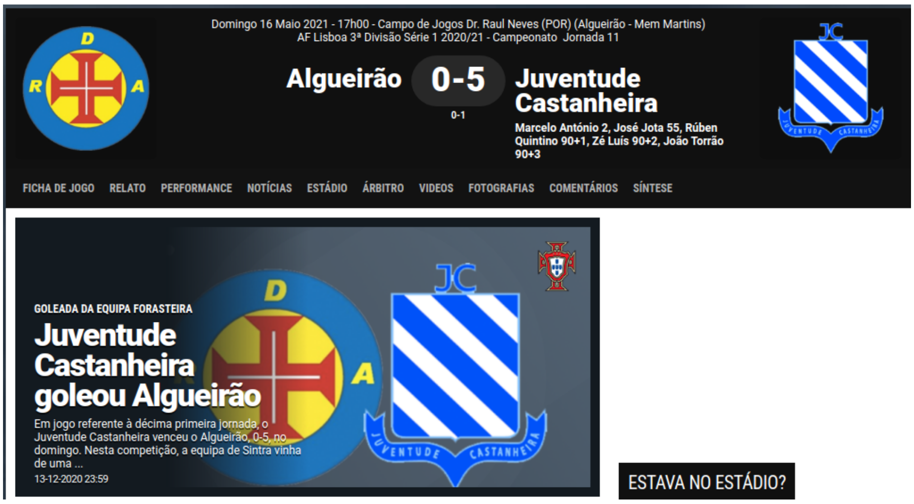
button and access a dedicated page (1). This page contains a text editor pre-filled with the text generated by Prosebot's NLG module (2) and a section that contains this same text so that the user can come back to it if needed. A table with curiosities is also included, as well as information regarding the match events. Additionally, there is a brief presentation of the initiative, including some warnings: for example, *zerozero.pt* discourages the inclusion of personal opinions or value judgments.

After submitting the summary (3), the collaborator may fill a short survey to convey his opinions regarding user experience. The summary will then be published on the website and become accessible at the top of the match page (4). To maintain transparency, the summary will show an alert referring both Prosebot and the collaborator that worked on it, identical to the one shown in Figure 4.7. Other collaborators may also leave comments and rate the summary with respect to the quality and relevancy (5), aspects that may be used by the moderation team to remove biased summaries, for example.

Additionally, taking in consideration feedback from the journalists and the product managers (Section 4.3.2), it was decided that Prosebot's summaries were listed in a dedicated page[2], which also presents the platform to the public. This was made to distinguish these summaries from *zerozero.pt*'s news articles, since only the latter will appear in the homepage. However, one detail that was not changed was in regards to the match page, which usually includes a banner linking to the news article. Due to infrastructure constraints, this banner was also kept for Prosebot's summaries, which is similar to the one shown in Figure 4.8. Regarding the name for Prosebot's texts, we have adopted the term "síntese", which in English may translate to "summary". Once again, it is important to clearly distinguish between these texts and news articles, since journalists are not directly involved with the writing of Prosebot's summaries.

Finally, a Software Requirements Specification document can be found in Annex A. It was written after multiple requirements gathering meetings with *zerozero.pt*'s product managers and also considering the results of the survey to *zerozero.pt* journalists. It includes an overall description of the platform, relevant definitions, user classes, user and software interfaces, as well as system features and requirements.

[2] https://zerozero.pt/prosebot.php



Domingo 16 Maio 2021 - 17h00 - Campo de Jogos Dr. Raul Neves (POR) (Algueirão - Mem Martins)
AF Lisboa 3ª Divisão Série 1 2020/21 - Campeonato Jornada 11

Algueirão 0-5 Juventude Castanheira

0-1

Marcelo António 2, José Jota 55, Rúben Quintino 90+1, Zé Luís 90+2, João Torrão 90+3

GOLEADA DA EQUIPA FORASTEIRA

Juventude Castanheira goleou Algueirão

Em jogo referente à décima primeira jornada, o Juventude Castanheira venceu o Algueirão, 0-5, no domingo. Nesta competição, a equipa de Sintra vinha de uma ...

13-12-2020 23:59

Figure 4.8: The top section of a *zerozero*'s match page, including a banner for the associated summary.

Chapter 5

Evaluation

5.1 Methodology

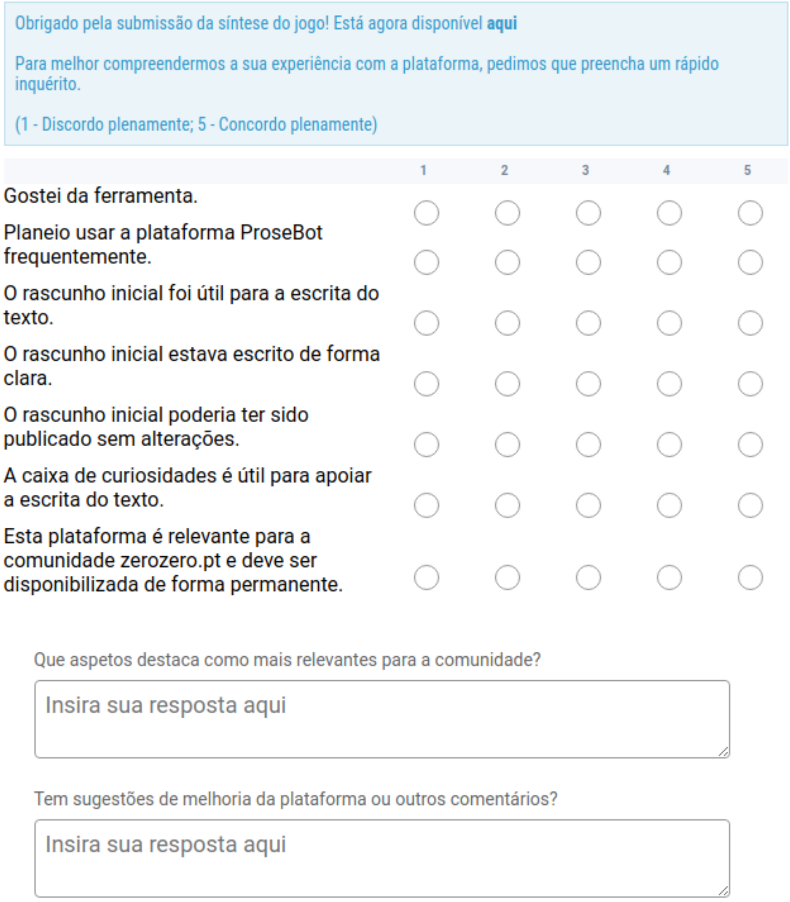
5.1.1 Post-Submission Survey

In order to more efficiently obtain feedback with regards to user's experience with the platform, we presented a short, quick survey after the submission of an article. The same user could answer it multiple times, or choose to skip if, for example, he submitted multiple articles in a short span of time. The survey contains seven 5-point Likert scale questions, partially based on the SUS (System Usability Scale) survey [6]: these questions are related to the user's overall opinion on the platform ("This platform is relevant for the *zerozero.pt* community and should be permanently available."), but also with his opinion regarding the quality and usefulness of the text ("The initial draft was written in a clear manner", "The initial draft could be published without any editions"). There are also two open-ended questions, where the user can express his opinions on the platform's strengths and points to improve. Figure 5.1 shows the survey as it was displayed to the users.

5.1.2 Automatic Evaluation

The main goal of our automatic evaluation is to understand how the collaborator edits the generated text: how much text does the collaborator keep, what kind of text does he keep, add or remove, and how similar the initial and final texts are. This may help us find patterns and devise action points to improve Prosebot in the future, by for example, updating the templates or the information that is reported. To start, we have applied two metrics when comparing the drafts and published texts: the Dice coefficient [11], and a metric identical to the one used in STORIUM [2].

The first step is to preprocess the generated and final texts. We unescape HTML entities and remove HTML elements such as paragraphs and anchors (which Prosebot uses to link to player and match pages), as well as remove any non-textual characters.



Obrigado pela submissão da síntese do jogo! Está agora disponível **aqui**

Para melhor compreendermos a sua experiência com a plataforma, pedimos que preencha um rápido inquérito.

(1 - Discordo plenamente; 5 - Concordo plenamente)

	1	2	3	4	5
Gostei da ferramenta.	○	○	○	○	○
Planeio usar a plataforma ProseBot frequentemente.	○	○	○	○	○
O rascunho inicial foi útil para a escrita do texto.	○	○	○	○	○
O rascunho inicial estava escrito de forma clara.	○	○	○	○	○
O rascunho inicial poderia ter sido publicado sem alterações.	○	○	○	○	○
A caixa de curiosidades é útil para apoiar a escrita do texto.	○	○	○	○	○
Esta plataforma é relevante para a comunidade zerozero.pt e deve ser disponibilizada de forma permanente.	○	○	○	○	○

Que aspetos destaca como mais relevantes para a comunidade?

Insira sua resposta aqui

Tem sugestões de melhoria da plataforma ou outros comentários?

Insira sua resposta aqui

Figure 5.1: Post-submission survey.

The metric uses Python's *SequenceMatcher*[1], a class for comparing pairs of sequences. It employs an algorithm that finds the longest contiguous matching subsequence that does not contain elements such as whitespace. The same process is applied recursively to the pieces of sequences to the left and right of the matching subsequence. One particular method of this class, `get_opcodes()`[2], returns a list of tuples describing how to turn one document into another, through operations of adding, deleting, replacing, or maintaining a sequence of text. Figure 5.2 shows an example of applying this function taken from one of the summaries submitted on the platform. Using this method, the function saves the sequences of text that the user maintained, and later divides the cumulative length of these sequences with the length of the initial text, to find the percentage of the text that was kept from the initial draft. Therefore, the formula for USER is the following:

$$USER(x,y) = \frac{|Match(x,y)|}{|x|} \tag{5.1}$$

[1] https://docs.python.org/3/library/difflib.html

[2] https://docs.python.org/3/library/difflib.html#difflib.SequenceMatcher.get_opcodes

Initial Text	Final Text	USER	Dice
Amarante FC venceu Marítimo B	Amarante FC venceu Marítimo B	1.00	1.00
Os Montelavarenses recebeu e derrotou o SC Frielas	Os Montelavarenses recebeu e venceu o SC Frielas	0.88	0.71
Vila Pouca vence UDC Sabrosa com goleada	SC Vila Pouca vence sem contestação	0.43	0.36
Desportivo O. Moscavide e Águias Musgueira não foram além de um empate	Empate caseiro coloca Desportivo O. Moscavide na primeira divisão da AFL	0.25	0.19
GD Resende recebeu e derrotou o Sátão	Vitória de mão cheia	0.00	0.00

Table 5.1: Samples of values returned by USER and the DICE coefficient.

Where $Match(x,y)$ represents the sequences of texts that were found in text x and kept in text y. Additionally, the program also stores other information, including the content added or removed by the user, and its total length. This process is repeated for each section of the summary: title, subtitle, and large text.

One problem with this approach is that in a hypothetical case where a user only changes the order of the paragraphs but maintains the overall text, the `get_opcodes()` function will use operations of addition and removal. Therefore, we also employed the Dice coefficient, which calculates the bigram similarity between two texts, disregarding its order of appearance. The formula for the Dice coefficient is the following:

$$Dice(x,y) = \frac{2n_t}{(n_x + n_y)} \tag{5.2}$$

Where n_x is the number of bigrams for text x, n_y is the number of bigrams for text y and n_t is the number of bigrams found in both texts. Table 5.1 shows values of the USER metric and Dice coefficient, when applied to titles of summaries retrieved from the Prosebot platform.

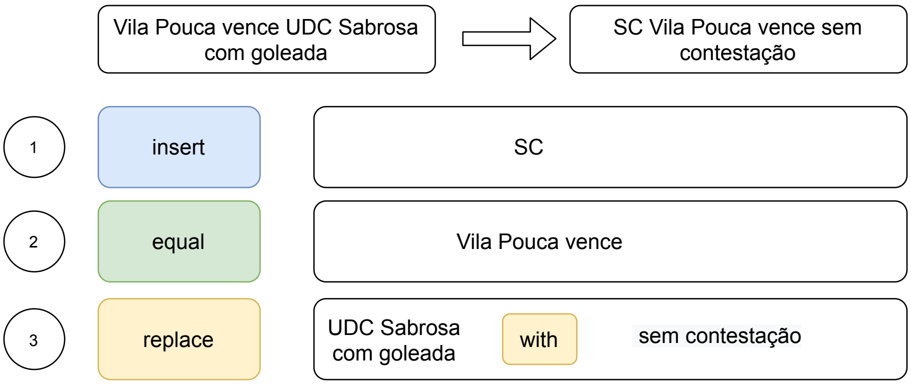


Figure 5.2: Sample of the results returned by the get_opcodes function.

Indicator	Value
Number of summaries	137
Number of authors	37
Number of teams covered	175
Number of competitions covered	52
Mean summaries per author	3.7
Mean summaries per team	1.6
Mean summaries per competition	2.6
Mean visits per summary	48.8

Table 5.2: General indicators.

5.2 Results

5.2.1 Summaries Characterization

The platform was launched in May 2021, as *zerozero.pt* gave permissions to the previously mentioned "edition collaborators" to start publishing match summaries. Despite there not being an official announcement, the collaborators were made aware of the platform via messages presented in competition and match pages, indicating they could contribute to the respective competitions and matches using Prosebot. This was decided by *zerozero.pt*'s product managers with the purpose of not overflowing the platform from the beginning, and to see how the platform would organically evolve.

The results hereby presented correspond to 30 days of usage and Table 5.2 presents some general indicators. Regarding platform usage, we can see that most authors tend to generate between 1 to 4 summaries, but there are also more engaged collaborators, as the five most participative account for 39% of the summaries. While trying to understand the usage patterns of these more active users, we found that one of them mostly generated summaries for matches involving one particular team which we can only assume is one of his favorites. Another user has the opposite behavior, never generating more than one summary per team. We also noticed two users that focused their attention on competitions from a certain location (e.g. Lisbon), while another collaborator only generated summaries for women's football matches.

As for the activity, Figure 5.5 shows that 53% of the summaries have been published up to two days after the match took place, while approximately 19% of the summaries were related to matches that took place more than 1 month prior to the summary being published. One possible explanation for this is that the collaborators may have wanted to experiment with the platform by generating summaries for older matches, while their preferred competitions/teams did not have new matches. Nonetheless, it makes sense that most summaries are generated near a match's occurrence, since it is when there is the most interest for that match.

The most represented competitions are displayed in Figure 5.3 and we can find many district leagues (denoted by the "AF"), as well as two women's football leagues. Regarding the visits to the summaries, Figure 5.4 show that while many summaries did not surpass the barrier of 50 views (70%), there were still some summaries that achieved the marks of 200 or 300 views – approximately 10% had more than 100 visits. In our opinion, the fact that there are summaries that can achieve this kind of viewership is very important, since it may influence the readers and collaborators to read more about the platform and help increase its relevance in the future. In total, more than 9200 visits to collaborator-generated summaries were recorded.

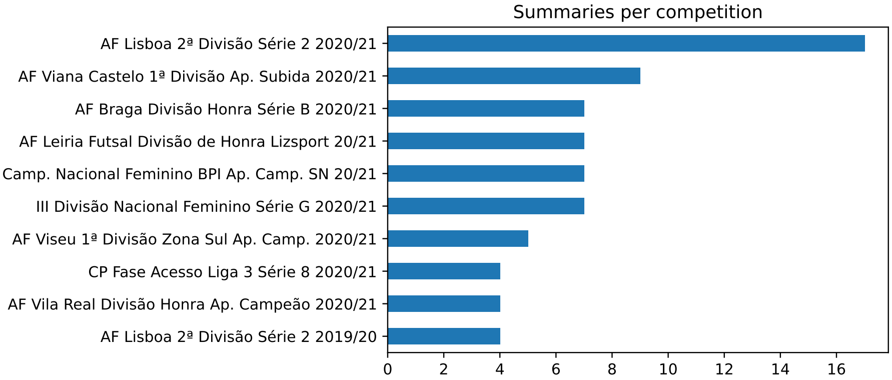


Figure 5.3: Most represented competitions.

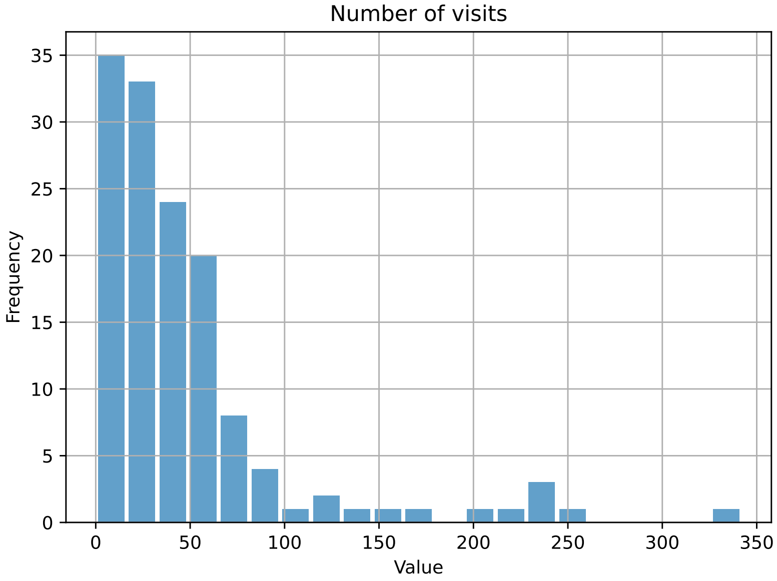


Figure 5.4: Distribution of visits to summaries.

5.2.2 Post-Submission Survey

A total of 42 answers have been collected: overall, they were very positive, as seen in Figure 5.6, as the mean for every question is above 4. Regarding the question of planning on using the platform frequently, we have noticed that 73% of the respondents have generated more than one text.

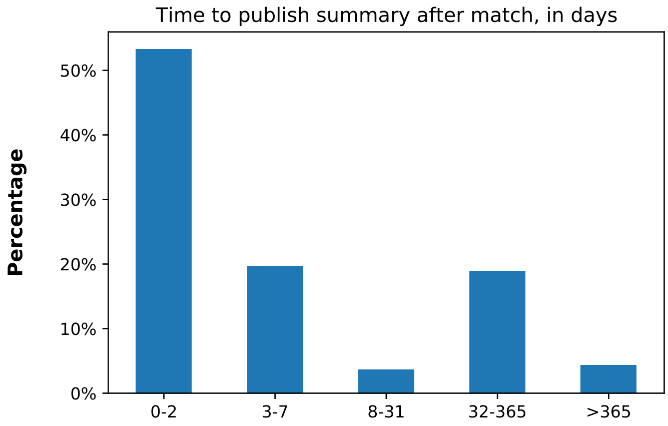


Figure 5.5: Time to publish the summary after the match takes place.

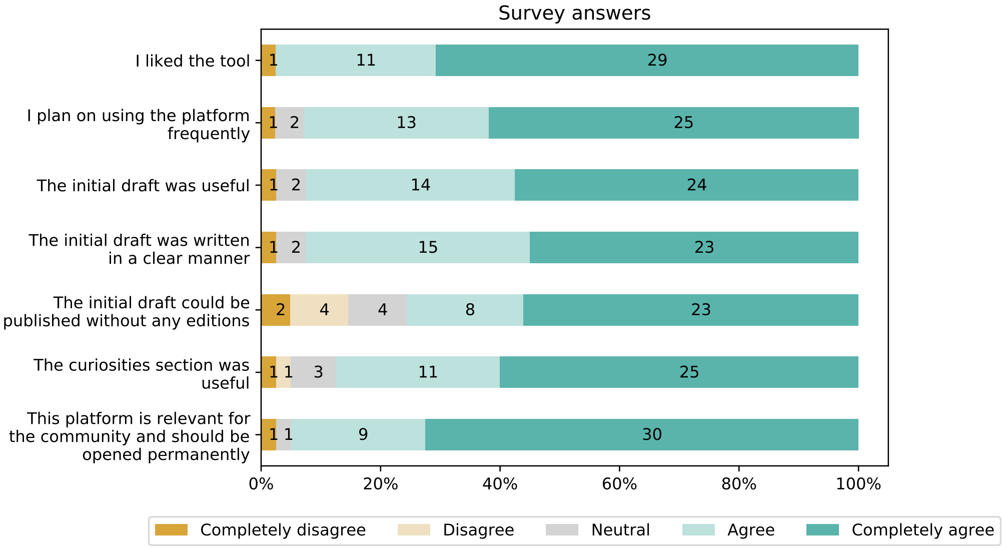


Figure 5.6: Survey results summary.

In the open-ended questions, the participants left suggestions regarding a visual bug and the design of the platform, which due to time and resource constraints was not in conformity with the rest of the website. Another user suggested allowing the generation of a summary without having the match information completed, which would most likely be unfeasible due to the fact that the draft would also be incomplete.

We wonder if the very positive sentiment towards the platform is caused by the excitement of the collaborators for being able to provide a more meaningful contribution to the community and their preferred teams – like the last question shows, almost 100% of the respondents think the platform should be permanently available.

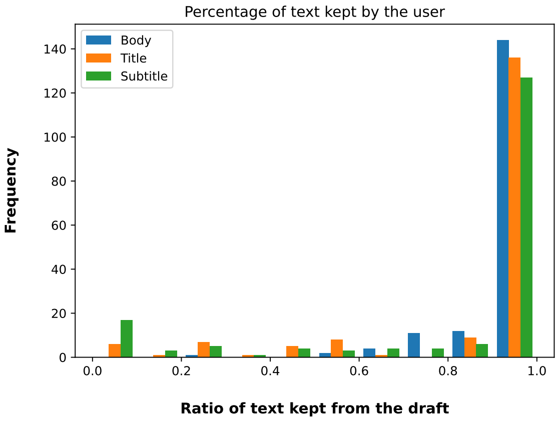


Figure 5.7: USER metric distribution.

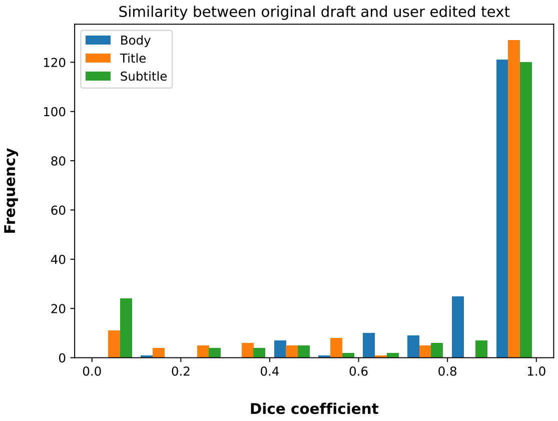


Figure 5.8: Dice coefficient distribution.

5.2.3 Text Comparison

In this section we will present our experiments made with the drafts generated by Prosebot and the final texts submitted by the collaborators. To compare the similarity between the texts, we apply the USER and Dice metrics, and calculate the difference in length. We have applied the Pearson's correlation between the metrics and user judgments to find eventual connections. Then, to study what kind of text the collaborators keep, add or remove, we have used an automatic word frequency analysis, as well as part-of-speech tagging. The evaluations were made with the body of the summary, since it is its most important part. Although we have also made calculations for the title and subtitle, due to their small length we do not believe we can extract great conclusions.

In a majority of cases (82% of the summaries), more than 90% of text generated in the initial draft is kept for the final version (Figure 5.7). With regards to the Dice coefficient (Figure 5.8), there is a mean of 89%, with 70% of the summaries having a value higher than 90%. Actually, approximately 40% of the texts had the maximum Dice coefficient, which means the drafts were kept unchanged. These values are surprisingly high, but we believe a possible explanation for this is the impact of the COVID-19 pandemic. Since while our experiment took place, the public was not allowed to attend football matches, we believe that a possible factor for the low amount of edits is that the collaborator may not have watched the game. In the future, the post-submission survey should be updated with this question, in order to better understand how it impacts the writing of the summary. In both figures, we can see how prevalent it is to maintain a high portion of the original draft, and only in the title and subtitle we can see more pronounced occurrences of collaborators discarding most of the text. In these sections in particular, it is very easy to have a low or high match depending on if the users use the same names to refer to the teams, since these texts tend to be very small. In Figure 5.9 we can also see a distribution of the difference in length between the original and final body of the texts and, in line with the previous results, in 80% of the cases the texts differ by less than 20 tokens, with 52% having the exact same length. There is also a tendency for users to add content instead of reducing the text.

We have applied Pearson's correlation [22] between the USER and Dice metrics as well as two

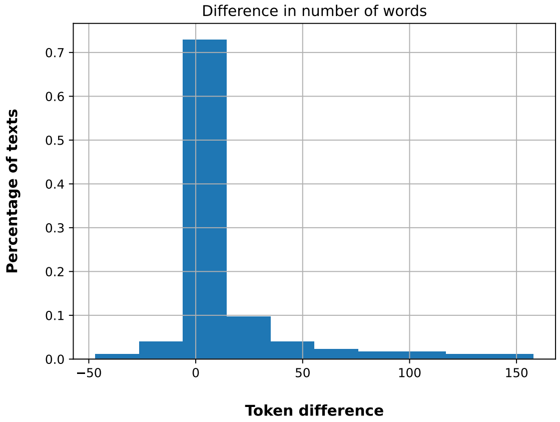


Figure 5.9: Length difference between original draft and published text.

questions from the post-submission survey related to the quality of the text: clarity and necessity of changes before submission, whose results are found in Table 5.3. We can find a high positive correlation (>0.7) between the two survey questions, which is expected since they have similar results. There is also a very high correlation between Dice and USER, which indicates that the hypothetical case we mentioned in the previous section (change in the order of paragraphs/sentences) was not a frequent occurrence. Regarding correlations between the metrics and survey questions we can find weaker values, but they are very similar between each other. These values indicate that even if the user found the texts clear, and that they could be published without changes, he may still want to make editions to provide further information, or vice-versa. To confirm this, we have manually checked the results for the collaborators that replied to the post-submission survey, and have concluded that despite there being instances where the metrics and the survey answers are equally high, there are also situations where a lower score on the survey does not equate to lower metric results. Finally, when looking at these results we have to keep in mind that not every collaborator filled the post-submission survey after submitting a summary, which does not allow us to fully study these correlations.

Additionally, we have applied a manual and automatic word frequency analysis (not including stopwords) to understand what kind of text the collaborator usually adds or removes, and we have included the most relevant results in Tables 5.4 and 5.5.

The first result in table 5.4 can be easily linked with this template:

Table 5.3: Pearson's correlation between user judgments and metrics.

	Needlessness for changes	USER	Dice
Clarity	0.71	0.24	0.22
Needlessness for changes	-	0.44	0.40
USER	-	-	0.87

> **Text**: “Both {home_team.name} and {away_team.name} held their first game in the competition.” **Condition**: “first_competition_game”

We find that the most probable reason for this template to be removed so much is that in league competitions, including it in the text becomes redundant if we previously said that the game was for the first fixture of the competition. In the future, this template could be removed for league matches, or perhaps removed completely.

We have noticed that the collaborators like to add context to the final paragraph regarding the classification after the match, which is why in Table 5.4 we see *n-grams* containing terms such as “position” or “classification”: while Prosebot only says the positions in the ordinal or cardinal form for each team, the authors include if a team is in first or last places, if a team has the same points as another, if a team has closed or extended the gap to other teams, or if a team has won the league. In other situations, the authors add information that Prosebot does not have access to, such as goal opportunities or rule breaks. Additionally, it seems that some users like to rewrite the goal sentences in order to provide more context, as we can see that some of the most removed *n-grams* mention the minute a goal took place.

frequency	bigram/trigram
11	quer equipa
6	posição pontos
6	minuto jogo
6	deste resultado
5	marcou golo
5	lugar classificação geral
4	segundo golo
4	relógio assinalava minuto
4	ocupa primeiro
4	fechou contagem
3	terceiro minuto jogo
3	primeiro lugar classificação
3	passagem minuto
3	ocupa primeiro lugar
3	marcou segundo golo
3	grande penalidade
3	geral encontrar
3	encontrar quarta
3	classificação geral pontos
3	apontou quinto temporada

Table 5.4: Most frequently removed bigrams/trigrams.

frequency	bigram/trigram
8	segunda parte
7	equipa casa
6	grande penalidade
5	ambas equipas
4	guarda redes
3	vasco gama
3	minuto jogo
3	jogo atraso
3	frente tabela
3	equipa gd
3	duas equipas
3	dentro baliza
3	cd cinfães
3	amarante fc
3	af lisboa
3	adc lodares

Table 5.5: Most frequently added bigrams/trigrams.

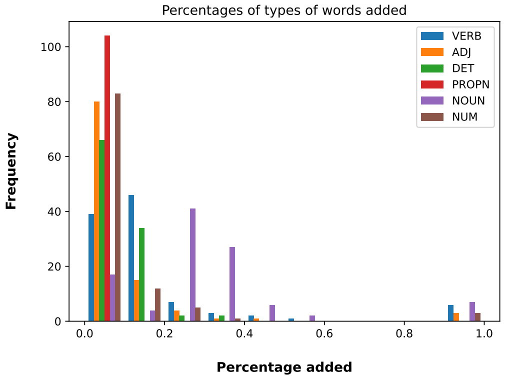


Figure 5.10: Distribution of added text per type of word.

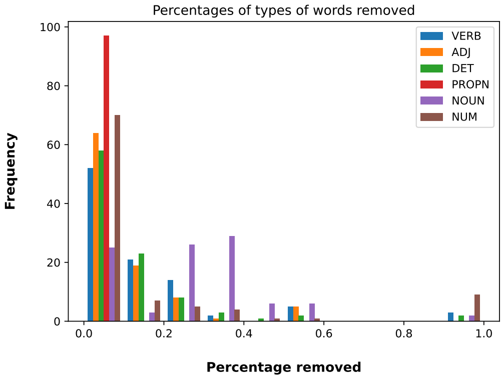


Figure 5.11: Distribution of removed text per type of word.

Finally, to understand what types of words the collaborators kept, added, or removed, we have applied part-of-speech tagging using spaCy[3], a natural language processing tool for Python. For these experiments, we have considered verbs, adjectives, determinants, numerals, nouns and proper nouns. First, we have studied the distribution of added and removed text per type of word, as seen in Figures 5.10 and 5.11. It is interesting to note that, on average, names make up 30% of the added text, verbs amount for other 16%, while proper nouns, determinants, adjectives and numerals are rarely added. There is a slightly more balanced distribution for the removed text, as on average nouns make up for 25%, verbs for 13%, determinants for 11% and numerals for 15%. We believe the latter value is probably related to the rewriting of the goal events by discarding the minute numbers, as previously explained in this section.

We have also studied the distribution of text kept by the collaborators, that is, for each type of word, the percentage of text usually kept by the authors. Figure 5.12 shows overall very high values for every type of word, with averages of at least 93%. Proper nouns achieve the highest average with 97%, indicating that the collaborators are more likely to keep the names of the teams, competitions and players. In the last "column" of the figure, proper have a lower frequency relatively to other types, and this is most likely due to the part-of-speech tagger failing to identify proper nouns in other texts.

[3] https://spacy.io

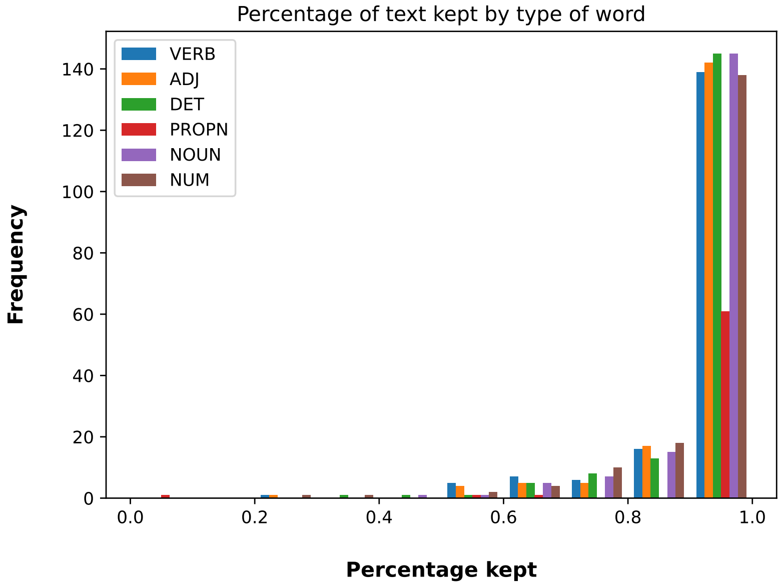


Figure 5.12: Distribution of types of words kept by the collaborators.

Chapter 6

Survey on the Impact of Automatic Tools on the Newsroom

Following the survey conducted within *zerozero.pt*'s newsroom, we decided to expand on it and conduct a survey with Portuguese journalists, regardless of area of work, with the only requirement being the possession of a journalist's professional license. The survey is similar to the first, however we excluded the first section more closely tied with the design of the Prosebot platform, replacing it with questions for general characterization of the participants, although it remained anonymous. We also added questions to the second section, regarding the impact of automatic tools on the newsroom. We have asked the Journalists' Union to divulge the survey among its members, and have also shared it with personal contacts. It was active from the 18th to the 29th of May.

For the general characterization section, we consulted a report on a previous survey conducted with Portuguese journalists on the matter of working conditions [10], which was developed in collaboration with the Journalists' Union and the Journalists' Professional License Committee. The full survey can be consulted in Section B.2.

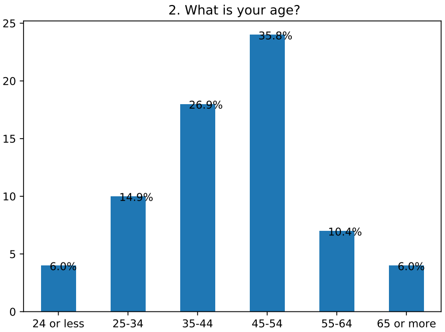


Figure 6.1: Age of the respondents.

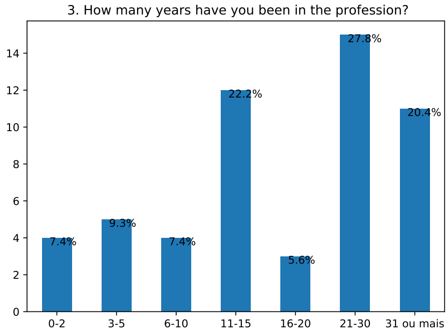


Figure 6.2: Years in the field.

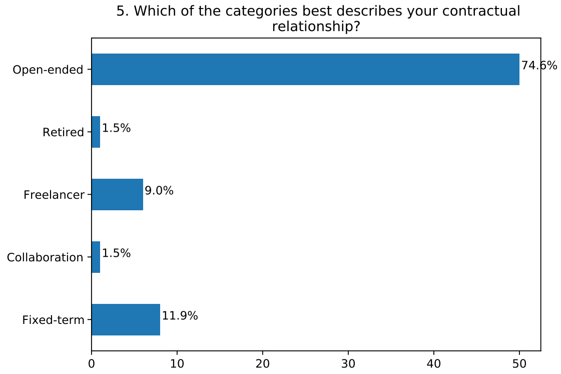


Figure 6.3: Contractual relationship.

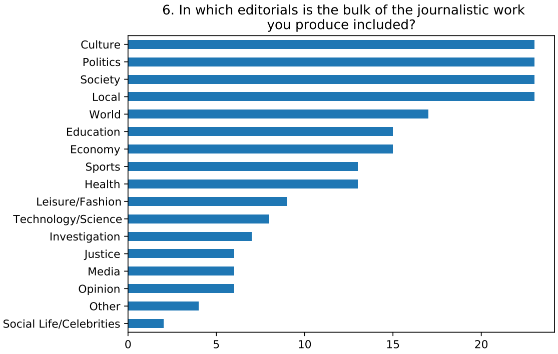


Figure 6.4: The journalists' main editorials.

6.1 General Characterization

The survey had 67 participants, with a slightly bigger distribution towards the male population, with 59.7% of the participants, where the remaining 40.3% were female. The modal age group is the one that includes participants between the ages of 45-54 as seen in Figure 6.1. With regards to the longevity of their professional career (Figure 6.2), we can highlight that a quarter of the respondents have been a journalist for more than 30 years, and if we include the respondents who have worked for more than 20 years, that number is increased to 52.4% of the participants.

Two questions were included to inquire about the contractual relationship of the participants. The large majority of the respondents (94%) are currently in activity, with the remaining 6% being retired or unemployed. As for the contractual bond, Figure 6.3 shows that most respondents (74.5%) are working under an open-ended contract ("sem termo"), while 11.9% have a fixed-term contract and 9% work as a freelancer.

The editorials that the respondents work in (not exclusively, i.e. multiple choices were allowed) are displayed in Figure 6.4. The sections with the biggest representations are Culture, Politics, Society, and Local, with all of these areas gathering 34.3% of the respondents. As for the mediums of communication that the journalists work in, an overview is provided by Figure 6.5. We can see that the Press gathers approximately 42% of the respondents as their main activity, while Online journalism through the use of websites is tied with the former as the most common secondary activities.

The majority of the respondents have not had practical contact with automatic tools, including almost 50% not being aware of the concept, as Figure 6.6 shows. In a follow-up question, we asked the respondents to share examples of their previous contact with these tools, and most of them mention agencies including Lusa, Forbes, Bloomberg and Reuters using automatic tools to generate news for the finances field. After further research, we found that Lusa's case has been reported in an article from ECO [19]. It mentions a program that fetches data from the opening and closure of the stock market, generates the text and fills the respective data, and finally sends it to a journalist. Therefore, this is a semi-automatic process, since there is always a final validation before the article is published. We were also able to find samples of articles published using

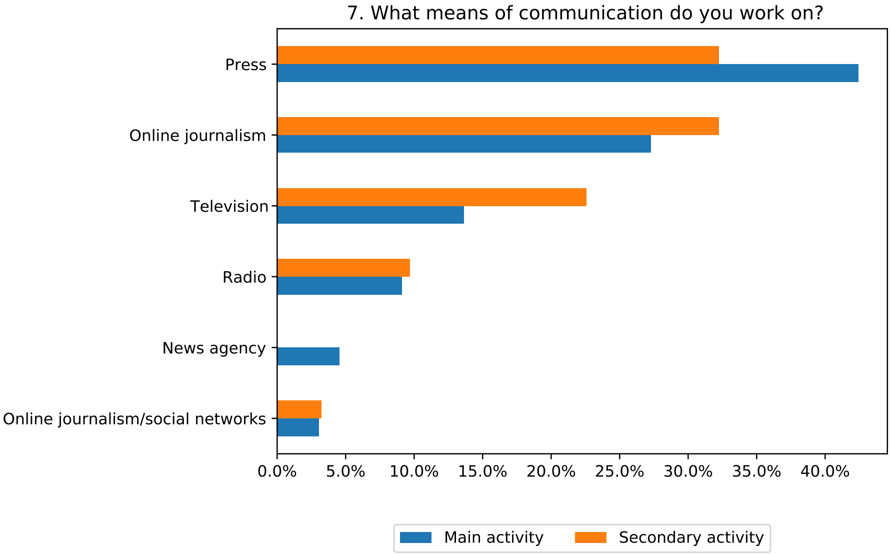


Figure 6.5: Journalists' mediums of communication.

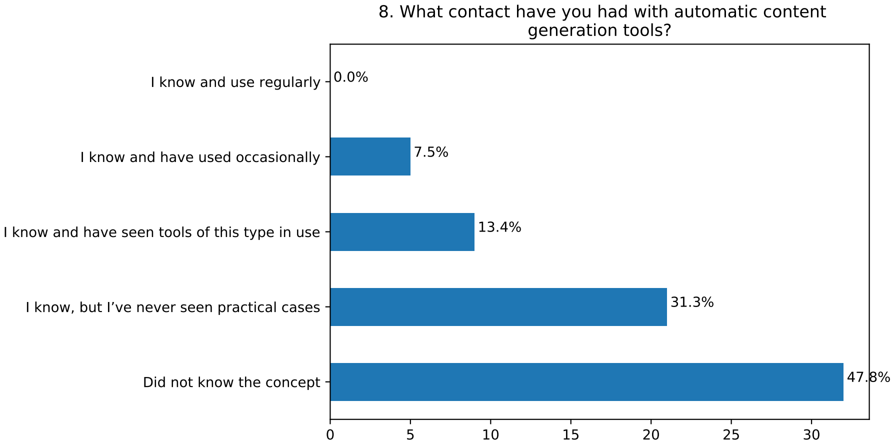
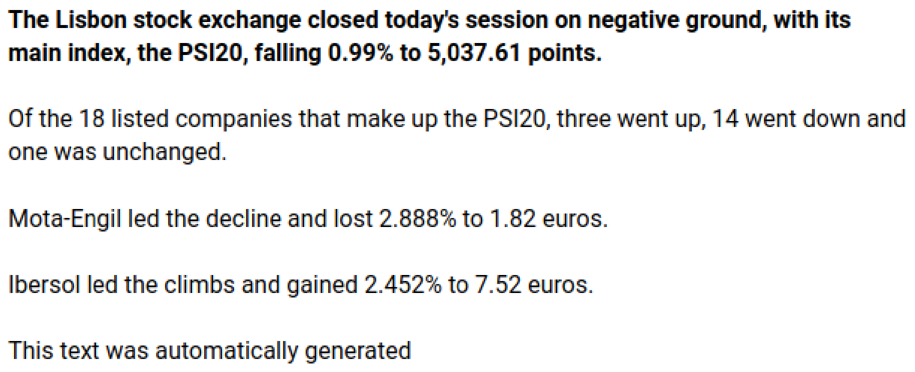


Figure 6.6: Previous contact with automatic tools.



The Lisbon stock exchange closed today's session on negative ground, with its main index, the PSI20, falling 0.99% to 5,037.61 points.

Of the 18 listed companies that make up the PSI20, three went up, 14 went down and one was unchanged.

Mota-Engil led the decline and lost 2.888% to 1.82 euros.

Ibersol led the climbs and gained 2.452% to 7.52 euros.

This text was automatically generated

Figure 6.7: Sample of automatically generated finances article from Lusa.

this method, which are all very similar and indicate that the article was automatically generated. Figure 6.7 shows one of these texts, translated from an article published by SAPO24 [17].

In the final question of this section, we inquired about the likelihood of automatic tools being implemented in the journalist's newsroom, and Figure 6.8 shows that 16.5% have a clear belief that such an event might happen. While a significant amount of respondents (23.9%) do not have a clear opinion on the matter, still the majority (59.7%) do not believe these tools will be used in their company.

6.2 Impact of Automatic Tools on the Future of Journalism

The first two questions of this section pertain to the matter of identifying the respective automatic tools when articles generated by them, partially or fully, are published. The overwhelming majority agrees they should be mentioned in both cases (respectively, 98,15% and 87.3%), which is a similar result to what was found in the survey to *zerozero.pt*'s journalists, and a practice that we follow in the Prosebot platform. If we recall the Lusa program mentioned in the previous section, it indicates that the articles were automatically generated, however they do not provide additional information regarding the tool.

Regarding the perceived importance of automatic tools in the multiple stages of news production, Figure 6.9 shows that from the initial draft to the final verification, there is a decay in the attached importance from the respondents. The biggest takeaway for us however, is that only approximately 20% of the respondents find automatic tools important in the elaboration of the initial draft. This number is much lower when compared to the more than 80% registered in the survey to *zerozero.pt*'s journalists, although admittedly the background of the respondents and their knowledge towards these tools may be a factor.

The next questions of this section are related to the strong and weak points of automatic tools when compared to journalists, as well as opportunities and threats which may arise from the use of these tools. We have adapted these questions from the conclusions drawn in a study from van Dalen [39]. A summary of the results can be found in Figure 6.10, with the labels for the questions

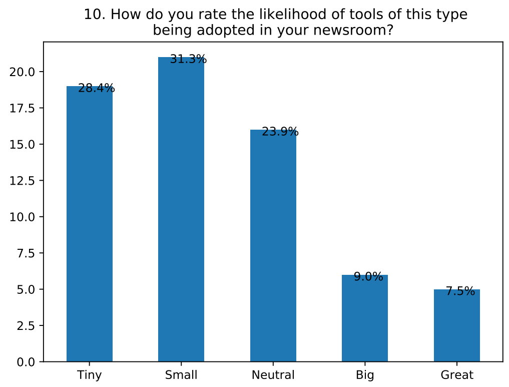


Figure 6.8: Likelihood of applying automatic tools at the journalists' newsroom.

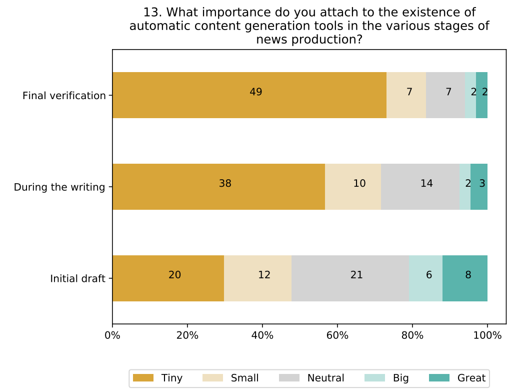


Figure 6.9: Importance attached to automatic tools in the various stages of news production.

presented in Table 6.1. Overall, there seems to be a negative view towards automatic tools. The respondents see themselves as more flexible and analytical [Q1, Q2], and do not believe these tools can provide a wider coverage than they do [Q3]. They realize that the production of content by automatic tools is cheaper [Q4] and faster [Q5], however most respondents do not believe these tools can help free their time to produce more detailed content [Q6] nor improve the content produced by the newsrooms [Q7]. There also seems to be a split in opinions on if automatic tools can help extend coverage to more specific topics, such as covering electoral results on the council level [Q8]. Additionally, as also highlighted in other sections of the survey, a majority of respondents believe there may be fewer journalists needed in the future [Q9].

With the introduction of automatic tools in the newsroom, ethical issues may arise, as agreed by 70% of the respondents [Q10]. Table 6.2 also gives us more insight into the opinions of the journalists, which mention the problems of transparency, authorship and accountability for the content, legal legitimacy, quality of sources, and possible programmer prejudices.

Comparing with the survey to *zerozero.pt*'s newsroom presented in Section 4.3.2, we find two main points of contrast. First, *zerozero.pt*'s journalists generally believe that automatic tools such as Prosebot can help extend coverage to smaller, more specific audiences, and the majority of them also do not believe they will be replaced in the future. We attribute these differences mainly to their background and the previous contact with Prosebot, since they realize their newsroom does not have the human resources to provide textual content for the many different competitions that take place each week, but they are also assured that Prosebot was not created to substitute their work, but instead help them, for instance by generating an initial draft that they can use when writing a news article.

Despite automatic tools being a good way to provide assistance with repetitive tasks, there are obviously aspects that they cannot replicate. Table 6.3 shows the most frequent *n-grams* found in the responses, which on one hand highlight journalism tasks including interviews and reports which require analytical skills and ability to understand, for instance, what sources to consult and how to interact with the interviewees. On another hand, the respondents mention the importance

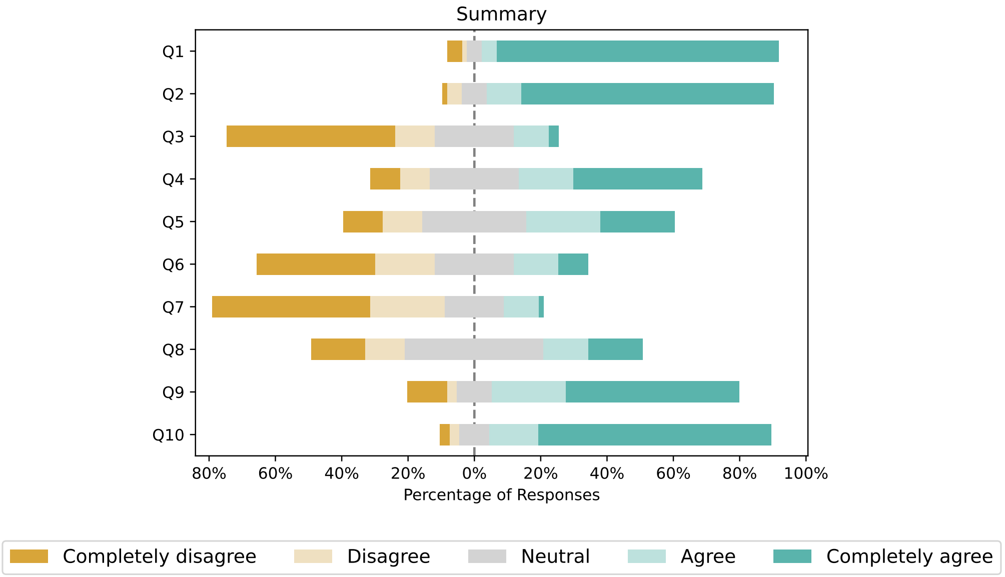


Figure 6.10: Results for Likert-scale questions.

Table 6.1: Survey question labels.

ID	**Question**
Q1	The flexibility of journalists, in contrast to automatic tools, allows them to cover content in a more personalized way, such as interviews, chronicles and reports.
Q2	Journalists' analytical skills allow them to provide a more in-depth coverage than automatic tools.
Q3	Automatic tools can help provide a wider coverage than journalists do.
Q4	Content produced exclusively by journalists has a higher financial cost than content produced by automated tools.
Q5	The production of content by automatic tools is faster.
Q6	Automating content frees journalists' time to investigate and produce more detailed news.
Q7	The introduction of automatic tools will result in the improvement of the content produced by the newsroom.
Q8	Automatic tools make it possible to extend coverage to more specific topics, generating, for example, content about football results for all levels of a competition or electoral information at the parish council level.
Q9	Increased use of automatic tools could lead to fewer journalists being needed.
Q10	The use of automatic tools raises ethical issues, such as copyright and transparency.

Table 6.2: Most frequent *n-grams* for the question of ethical problems.

Frequency	Bigram/Trigram
4	direitos autor
3	reflete preconceitos programador
2	interesse público
2	falsa objectividade
2	autoria informação
2	análise crítica

of understanding the context of what is being reported on, the human perception of events, the discourse that uses elements culturally shared with the audiences, as well as the creativity, subjectivity and personal view of a journalist, which are fundamental aspects to make a well-written article.

Having found examples of automatic tools being applied in areas including sports, weather, finances and politics (which we presented in Chapter 2), we wanted to see how useful the participants found these tools to be in these areas. As Figure 6.11 shows, the results are very similar for all of the options, with sports and weather having nearly 50% of the respondents giving a positive review. In a follow-up question, we asked for other examples where these tools could be successfully applied: approximately 43% did not provide an answer, and another 25% stated they do not find any other areas where the tools could be used. Other answers mention the application in fields that may generate lists of information or deal with a great amount of numerical data, giving examples such as reporting on pandemics, car traffic, releases of music or books, financial information (public purchases, account reports, corporate results, stocks exchange), as well as electoral results. Additionally, they also mention that automatic tools not only have to be used in the context of writing a report, but also in the collection and treatment of data, with examples including transcription and translation.

After presenting a collection of possible strong, weak points, threats and opportunities, we asked the journalists for other topics that could be included in this section. As main strong points, they mention the speed of automatic tools which leads to a higher quantity of information, while for weak points they highlight the lack of creativity and analysis to understand the nuance of events and what may be a truth or lie. As for threats, the respondents underline the reduction of journalism's social relevancy, the loss of diversity of content and approaches/points of view,

Table 6.3: Most frequent *n-grams* on the question of characteristics impossible to replicate.

Frequency	Bigram/Trigram
2	relação fontes
2	olhar jornalista
2	capacidade ouvir
2	análise crítica

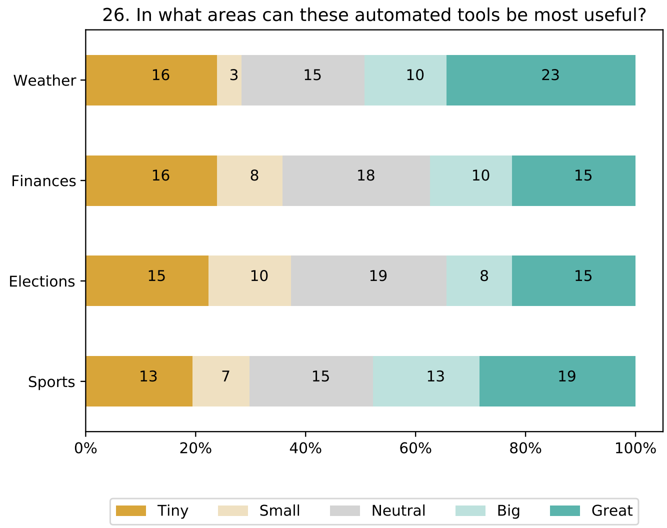


Figure 6.11: How useful are automatic tools in sports, finances, elections and weather.

lower quality of texts, the unaccountability for damages caused for third parties, and the increase in unemployment.

In the final question of the survey, we ask what changes could happen to the role of a journalist with the introduction of automatic tools in the newsroom. Many respondents share the fear of unemployment as a result of cost-cutting measures, while others think that these tools could be very helpful in handling tasks such as data collection and visualization, thus freeing time for journalists to reflect, ask questions, speak with sources, and pursue deeper types of work, such as investigative reports.

6.3 Further Analysis

In order to find possible patterns in the responses, we have crossed some general characterization questions with the main questions of the survey, in particular the questions of the likelihood of

Figure 6.12: Response distribution according to the answer to the question regarding previous contact with automatic tools.

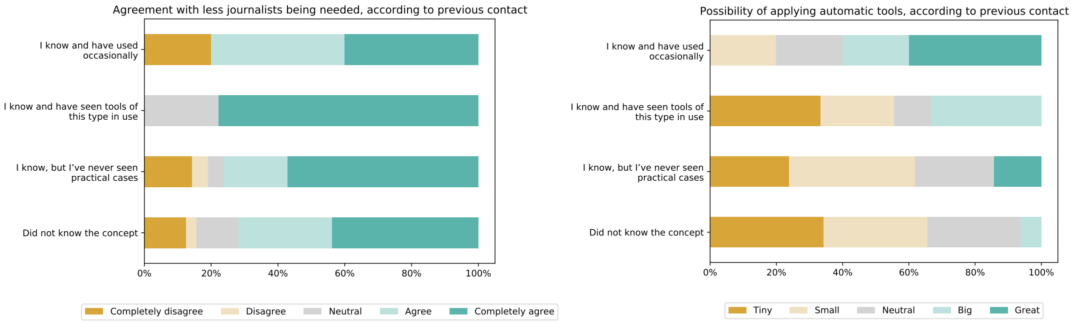
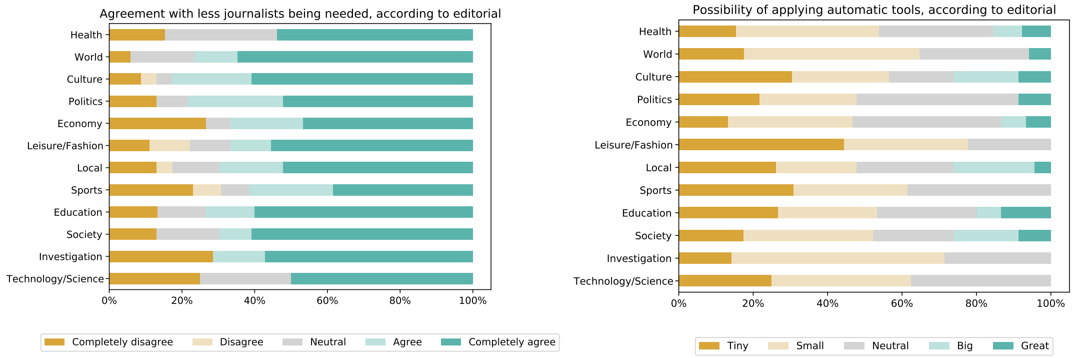


Figure 6.13: Response distribution according to the editorials of the journalists.



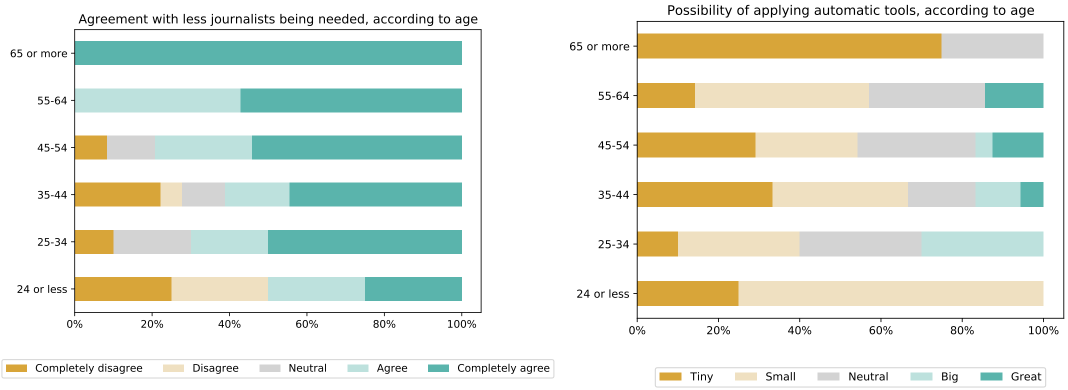
using automatic tools, and the possibility of fewer journalists being needed.

First, we used the question regarding the previous contact with automatic tools as a pivot. Figure 6.12 shows that this is not really a factor in the agreement with fewer journalists being needed, however we can see a trend on the possibility of applying automatic tools in the journalists' newsroom: the more a respondent is familiarized with the concept, the bigger the likelihood of such tools being used in his workplace.

Next, we studied the responses to the same questions, but using the journalists' editorials as a basis (Figure 6.13) – with this analysis, we have to keep in mind that one respondent may belong to multiple editorials. It seems that the editorials whose journalists are more receptive towards automatic tools are "local", "society" and "culture", although the percentages are still relatively low. We find it interesting that "sports" did not receive any positive answer, but one possible justification is that the respective newsrooms are not concerned with expanding coverage to more competitions. As for the other question we are studying, once again there are similar results across all editorials.

One more point of analysis is to study the answers according to the age of the participants. In Figure 6.14 we can see that, with the exception of the range 25-34 with the following, older respondents tend to agree more with the possibility of journalists being replaced. While there is

Figure 6.14: Response distribution according to the age of the journalists.



a split in the younger range, the older two ranges all share this sentiment. Despite this belief that journalists may be replaced in the future, it seems that, in the present, even the older ranges do not think there is a high possibility of such tools being used in their newsrooms.

The survey with journalists has helped us better understand the state of automatic tools in the Portuguese space, and how to proceed in the future as developers of these tools. Although most of the respondents were not acquainted with the tools, we have learned examples of applications for stock market reports by Portuguese news agencies. We have also observed an overall negative sentiment from the journalists, as they have not recognized advantages such as time-freedom and wider coverage, but show concerns towards the possibility of being replaced in the future, which marks a contrast with the better receptivity from *zerozero.pt*'s newsroom. Therefore, the journalists are not favorable to the unsupervised publication of automatically generated news articles, as they believe this would lower the quality of content produced. As developers, we understand this perspective, and believe that journalists are fundamental in the development and improvement of automatic tools. It is something we have strived to do with Prosebot: taking into account the feedback of the journalists helps us deliver a better product so that it can produce a competent draft that journalists can complete.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

We have presented the Prosebot platform, the first initiative in the Portuguese space that includes a community of readers in a process of post-editing of automatically generated sports match summaries. These summaries are published by *zerozero.pt*'s collaborators, after being given a draft generated by a data-to-text template-based system.

We started by surveying *zerozero.pt*'s journalists on their opinion on the platform and how some decisions (such as the name or signing of the texts) should be made, and also included some questions regarding the impact of automatic tools in the newsrooms. Overall, we received positive feedback: most likely due to their previous contact with Prosebot, the journalists agreed with the idea that the tool could be helpful in extending coverage to a wider range of competitions, and they also believe that the tool will be used as a support to their work, not as a competitor/replacement.

This survey was later extended to Portuguese journalists outside *zerozero.pt*'s newsroom (excluding the platform planning section), and we have encountered mixed opinions on automatic tools. There seems to be a negative opinion on these tools, in particular if they are used to publish news articles without the intervention of a journalist. The respondents believe that the adoption of automatic tools by media companies could lead to the dismissal of journalists via cost-cutting measures, which would in turn diminish the quality of the produced content. We understand the views of the journalists, which is why we believe it is fundamental that they are included in the development of these tools (as we have done in our work), so that we can better understand how to use these tools to their advantage.

Regarding Prosebot, we have made changes to the NLG system, taking into account feedback from the journalists and product managers, including more information to the text based on data and statistics from *zerozero.pt*'s database, fixed bugs with the algorithm and the templates, and also added more templates with the help of one journalist from the newsroom. Finally, we have launched the platform which had approximately one month of activity. The platform is still very recent and the preliminary results are very positive, as the surveyed participants have shown a very affirmative opinion on the platform, and the analysis of the user changes shows that a large amount of text is kept by the users, meaning that the system generates a useful draft. We should still consider the COVID-19 pandemic which has impacted the world of sports: some competitions did not take place, and for the matches that did happen, the public was not allowed to attend, and therefore this may have also affected the platform's activity.

7.2 Future Work

This work also opens many possibilities for future improvements. The analysis of the changes made to the text may be used to add new templates to the NLG system and thus increase its diversity. For instance, we have noticed that in matches where there are a lot of goals, the events paragraph feels a bit repetitive because there is at least a sentence for each goal. One interesting approach we have seen in the collaborator's summaries is goal aggregation (e.g. "The home team was already winning 2-0 after the first 10 minutes").

We also believe that the versatility of the NLG module can help expand the platform: as of now, the platform is available for the Portuguese football section of the website, however Prosebot's template system allows it to generate summaries for multiple sports, including football, futsal and hockey, as well as multiple languages such as English, Brazilian Portuguese and Spanish. Therefore, in the future, the experiment we conducted could be expanded to other sections of *zerozero.pt*, thus generating more data that can be used to improve the system.

Appendix A

Software Requirements Specification

A.1 Introduction

A.1.1 Purpose

This document specifies the requirements for the first release of a platform for the publication of sports articles, where a first version of the article is created by an NLG system, which is then edited by a collaborator before being published. There are two main purposes: to increase the number of matches covered with textual content and to improve the NLG system through the analysis of user feedback and editions made to the text.

A.1.2 Intended Audience and Intended Use

This document is aimed towards the developers and product managers of the platform and intends to provide an overview of the product in Section A.2, as well as a detailed list of requirements presented in Section A.3.

A.1.3 Project Scope

This platform aims to allow *zerozero.pt*'s readers to publish summaries for matches that are not already covered by *zerozero.pt*'s journalists, by starting off with a text generated by Prosebot, an NLG system that can automatically generate an article for a given match, and editing it as the user sees fit. After submission, the summary would be available on the website.

Our main goals are to publish more articles about matches in order to increase coverage, to bring Prosebot into production, and to understand how the community values the coverage of smaller competitions. In our point of view, this platform can be beneficial since it helps evaluate the text generated by Prosebot, i.e. by analyzing the changes made to the text, we can explore possible ways of improvement. Additionally, involving the community will possibly increase engagement, and even increase visibility for smaller teams.

A.1.4 Definitions

There are some concepts of our domain that we need to define in order to avoid confusion. As described above, **Prosebot** is a data-to-text template-based system that generates textual summaries for sports matches. We may refer to this system as the NLG module, or the Prosebot system. Since the product we are developing is based on this tool, we have decided to call it the **Prosebot platform**, so to distinguish these applications, when mentioned alone, Prosebot always refers to the NLG system.

Regarding the output of the Prosebot platform, it will be referred to as a **match summary**, since it aims to summarize the match's background, events, and aftermath. It is also a way to distinguish these texts from **news articles**, which are written by journalists – and that is not the case on this platform.

Finally, we refer to the users of the platform, i.e. the authors of the summaries, as **collaborators**.

A.1.5 Risks

The main risk we need to keep in mind is the collaborator's approach to writing a text: in this platform, we will give freedom to *zerozero.pt*'s collaborators to create summaries for matches, and there is a possibility that they introduce bias or make wrong value judgments on players or teams, which harm *zerozero.pt*'s image.

A.2 Overall Description

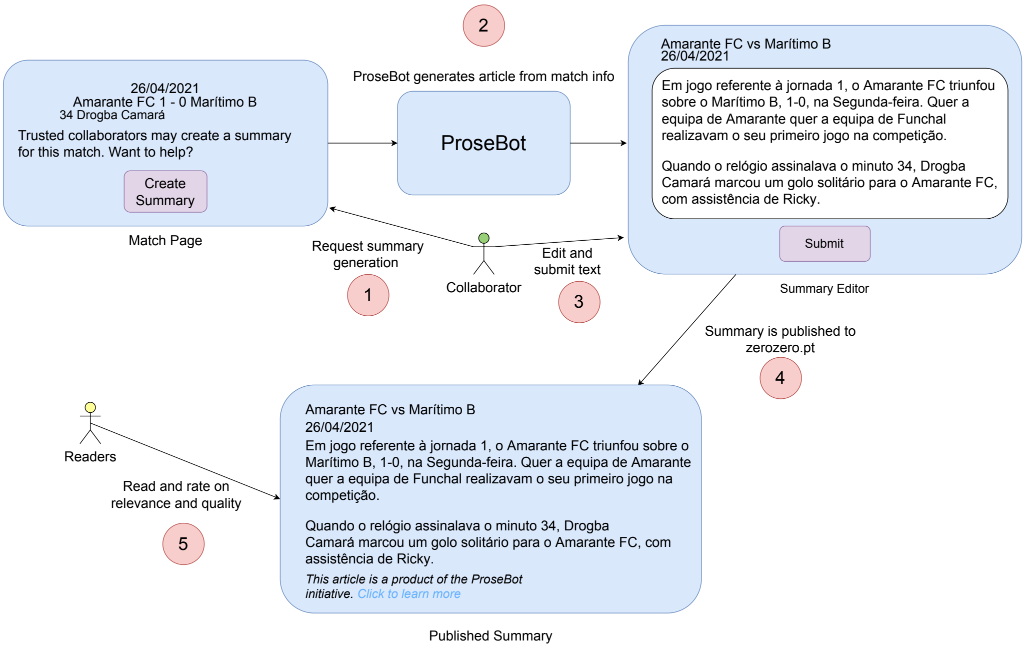
A.2.1 Product Perspective

Prosebot is an NLG system that can automatically generate news articles for a sports match, where the user only needs to input the match identifier from *zerozero.pt*'s website. It has been developed for several years, and one of the current goals is to use Prosebot as a tool to publish textual articles. This product intends to accomplish this goal by using Prosebot as a component of its system, that will provide a first draft for users to publish a match summary. Figure A.1 presents a rich picture of the platform, presenting the steps needed for a collaborator to create and publish a summary to *zerozero.pt*. There is also a component diagram for the summary creation process in Figure A.2, showing how each component of the system interfaces with one another.

A.2.2 Product Functions

The major functions the collaborator can perform are presented in the following list:

1. To contribute with a match summary;
2. To see a text editor pre-filled with Prosebot's output for a requested match;
3. To make changes and submit the summary;


2
ProseBot generates article from match info
26/04/2021
Amarante FC 1 - 0 Marítimo B
34 Drogba Camará
Create Summary
Match Page
ProseBot
Amarante FC vs Marítimo B
26/04/2021
Submit
Summary Editor
Request summary generation
1
Collaborator
Edit and submit text
3
Summary is published to zerozero.pt
4
Readers
Read and rate on relevance and quality
5
Amarante FC vs Marítimo B
26/04/2021
Published Summary

Trusted collaborators may create a summary for this match. Want to help?

Em jogo referente à jornada 1, o Amarante FC triunfou sobre o Marítimo B, 1-0, na Segunda-feira. Quer a equipa de Amarante quer a equipa de Funchal realizavam o seu primeiro jogo na competição.

Quando o relógio assinalava o minuto 34, Drogba Camará marcou um golo solitário para o Amarante FC, com assistência de Ricky.

Em jogo referente à jornada 1, o Amarante FC triunfou sobre o Marítimo B, 1-0, na Segunda-feira. Quer a equipa de Amarante quer a equipa de Funchal realizavam o seu primeiro jogo na competição.

Quando o relógio assinalava o minuto 34, Drogba Camará marcou um golo solitário para o Amarante FC, com assistência de Ricky.

This article is a product of the ProseBot initiative. Click to learn more

Figure A.1: Rich picture of the platform.

4. To view a submitted summary on the match page.

A.2.3 User Classes and Characteristics

This product will be used primarily by three groups of people: collaborators, readers and moderators.

zerozero.pt allows any of its readers to submit contributions to the website, such as completing information regarding a team or player, and those will gain the status of collaborator. Additionally, there are different subsets of collaborators: for this product, we intend to give access to *zerozero.pt*'s "edition collaborators" (users who can submit contributions to matches from certain competitions). To become an "edition collaborator", one needs to submit an application including their reasoning for being knowledgeable about the respective competition. After manual approval from *zerozero.pt*, the user may then submit match results, and now also match summaries.

We consider any of *zerozero.pt*'s users to be readers, and the moderators are a subset of *zerozero.pt*'s administrators which will monitor the generated summaries and act when necessary, for instance removing biased summaries.

A.2.4 Design and Implementation Constraints

zerozero.pt will be responsible for maintaining the platform. Additionally, the platform is embedded into *zerozero.pt*'s system, and therefore uses PHP across different components (editor page, summary page, Prosebot).

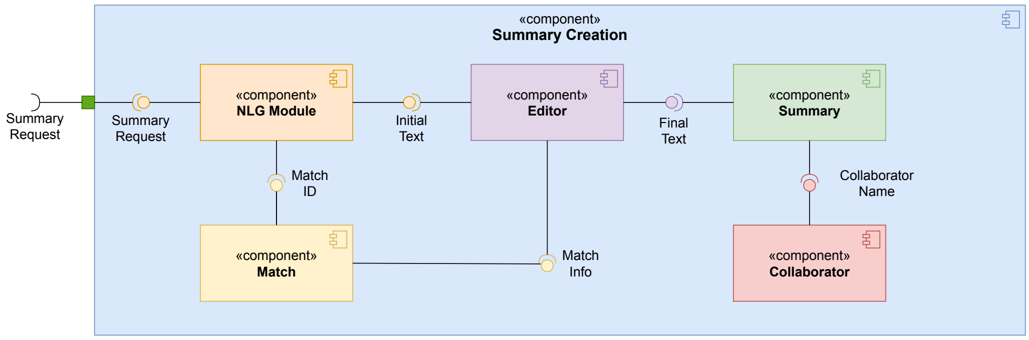


Figure A.2: Component diagram for the summary creation process.

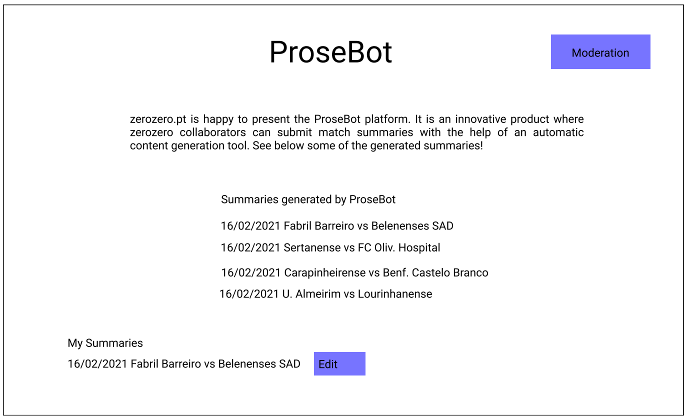
A.2.5 Assumptions and Dependencies

The product is heavily reliant on Prosebot, a current project at *zerozero.pt*, which will be responsible for generating the initial match summaries for collaborators to edit.

A.3 System Features and Requirements

A.3.1 User Interfaces

In this section are presented sample images for the user interfaces. They aim to highlight the main functions that the platform should provide. While these samples are mock-ups, the interfaces should be following *zerozero.pt*'s style guide.



ProseBot

zerozero.pt is happy to present the ProseBot platform. It is an innovative product where zerozero collaborators can submit match summaries with the help of an automatic content generation tool. See below some of the generated summaries!

Summaries generated by ProseBot

16/02/2021 Fabril Barreiro vs Belenenses SAD

16/02/2021 Sertanense vs FC Oliv. Hospital

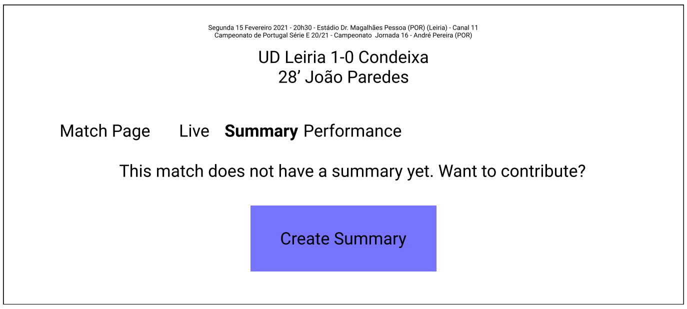
16/02/2021 Carapinheirense vs Benf. Castelo Branco

16/02/2021 U. Almeirim vs Lourinhanense

My Summaries

16/02/2021 Fabril Barreiro vs Belenenses SAD

Figure A.3: Hub sample image.

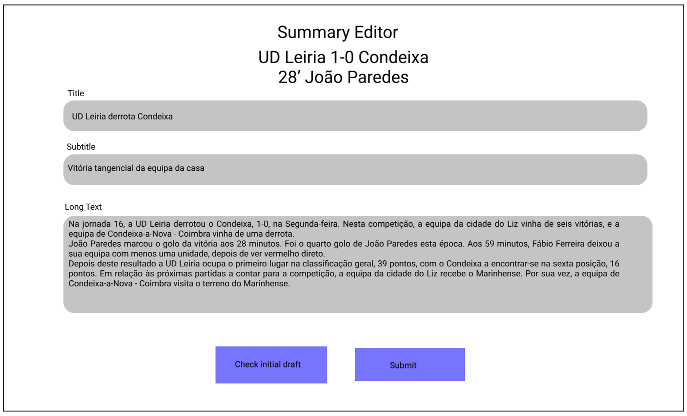


Segunda 15 Fevereiro 2021 - 20h30 - Estádio Dr. Magalhães Pessoa (POR) (Leiria) - Canal 11
Campeonato de Portugal Série E 20/21 - Campeonato Jornada 16 - André Pereira (POR)

UD Leiria 1-0 Condeixa
28’ João Paredes

This match does not have a summary yet. Want to contribute?

Figure A.4: Match page sample image.



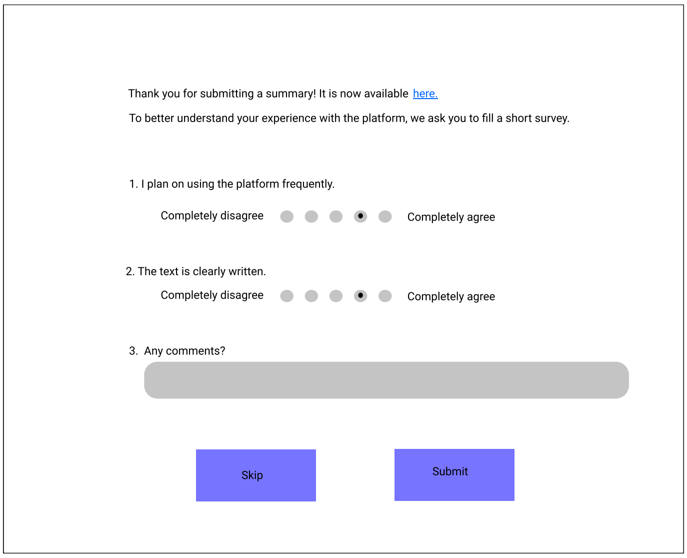
Summary Editor
UD Leiria 1-0 Condeixa
28’ João Paredes

Title: UD Leiria derrota Condeixa

Subtitle: Vitória tangencial da equipa da casa

Long Text:
Na jornada 16, a UD Leiria derrotou o Condeixa, 1-0, na Segunda-feira. Nesta competição, a equipa da cidade do Liz vinha de seis vitórias, e a equipa de Condeixa-a-Nova - Coimbra vinha de uma derrota.
João Paredes marcou o golo da vitória aos 28 minutos. Foi o quarto golo de João Paredes esta época. Aos 59 minutos, Fábio Ferreira deixou a sua equipa com menos uma unidade, depois de ver vermelho direto.
Depois deste resultado a UD Leiria ocupa o primeiro lugar na classificação geral, 39 pontos, com o Condeixa a encontrar-se na sexta posição, 16 pontos. Em relação às próximas partidas a contar para a competição, a equipa da cidade do Liz recebe o Marinhense. Por sua vez, a equipa de Condeixa-a-Nova - Coimbra visita o terreno do Marinhense.

Figure A.5: Editor sample image.



Thank you for submitting a summary! It is now available here.

To better understand your experience with the platform, we ask you to fill a short survey.

1. I plan on using the platform frequently.

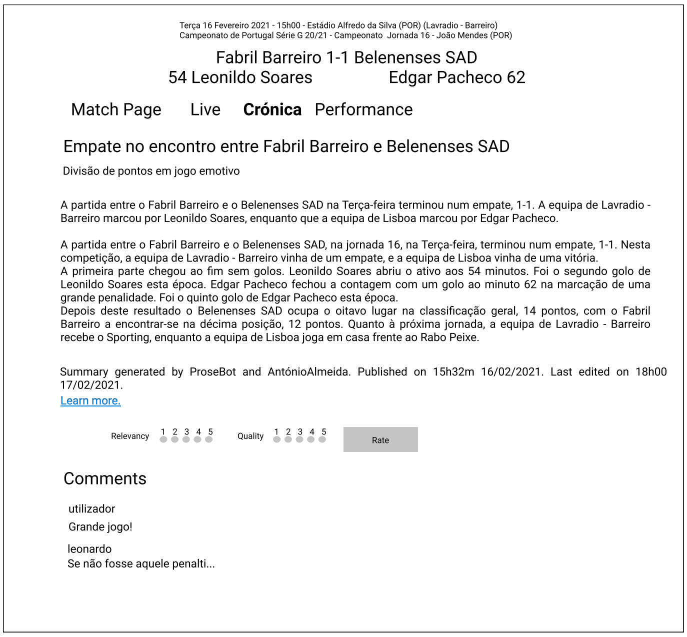
Completely disagree ○ ○ ○ ● ○ Completely agree

2. The text is clearly written.

Completely disagree ○ ○ ○ ● ○ Completely agree

3. Any comments?

Figure A.6: Post-submission survey sample image.



Terça 16 Fevereiro 2021 - 15h00 - Estádio Alfredo da Silva (POR) (Lavradio - Barreiro)
Campeonato de Portugal Série G 20/21 - Campeonato Jornada 16 - João Mendes (POR)

Fabril Barreiro 1-1 Belenenses SAD
54 Leonildo Soares Edgar Pacheco 62

Empate no encontro entre Fabril Barreiro e Belenenses SAD

Divisão de pontos em jogo emotivo

A partida entre o Fabril Barreiro e o Belenenses SAD na Terça-feira terminou num empate, 1-1. A equipa de Lavradio - Barreiro marcou por Leonildo Soares, enquanto que a equipa de Lisboa marcou por Edgar Pacheco.

A partida entre o Fabril Barreiro e o Belenenses SAD, na jornada 16, na Terça-feira, terminou num empate, 1-1. Nesta competição, a equipa de Lavradio - Barreiro vinha de um empate, e a equipa de Lisboa vinha de uma vitória.
A primeira parte chegou ao fim sem golos. Leonildo Soares abriu o ativo aos 54 minutos. Foi o segundo golo de Leonildo Soares esta época. Edgar Pacheco fechou a contagem com um golo ao minuto 62 na marcação de uma grande penalidade. Foi o quinto golo de Edgar Pacheco esta época.
Depois deste resultado o Belenenses SAD ocupa o oitavo lugar na classificação geral, 14 pontos, com o Fabril Barreiro a encontrar-se na décima posição, 12 pontos. Quanto à próxima jornada, a equipa de Lavradio - Barreiro recebe o Sporting, enquanto a equipa de Lisboa joga em casa frente ao Rabo Peixe.

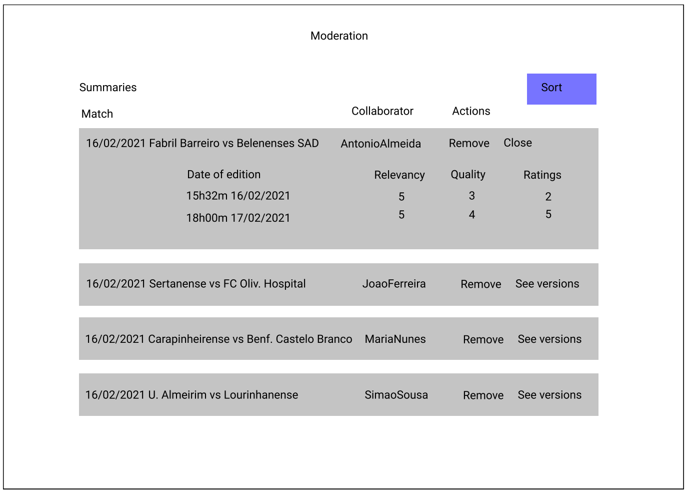
Summary generated by ProseBot and AntónioAlmeida. Published on 15h32m 16/02/2021. Last edited on 18h00 17/02/2021.
Learn more.

Comments

utilizador
Grande jogo!

leonardo
Se não fosse aquele penalti...

Figure A.7: Summary page sample image.



Moderation

Summaries

Match	Collaborator	Actions
16/02/2021 Fabril Barreiro vs Belenenses SAD	AntonioAlmeida	Remove Close
16/02/2021 Sertanense vs FC Oliv. Hospital	JoaoFerreira	Remove See versions
16/02/2021 Carapinheirense vs Benf. Castelo Branco	MariaNunes	Remove See versions
16/02/2021 U. Almeirim vs Lourinhanense	SimaoSousa	Remove See versions

Date of edition	Relevancy	Quality	Ratings
15h32m 16/02/2021	5	3	2
18h00m 17/02/2021	5	4	5

Figure A.8: Moderator dashboard sample image.

A.3.2 Software Interfaces

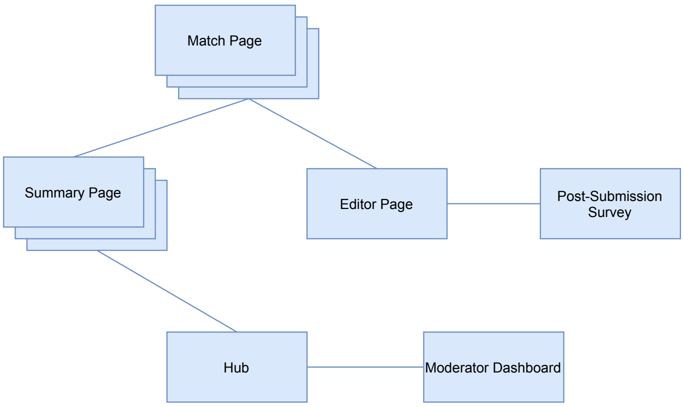


Figure A.9: Sitemap.

The platform should interface with the Prosebot NLG system, by requesting the generation of a match summary for the specified ID and fetching the response, which will be used to pre-fill the summary editor. Additionally, the platform uses *zerozero.pt*'s database to store information regarding the summaries created by the collaborators (e.g. match, initial and final text, date, survey results and ratings). Figure A.9 displays a sitemap showing the relationships between the different pages of the platform.

A.3.3 System Features

A.3.3.1 Hub

Description and Priority The platform hub will contain a presentation of the platform so that collaborators, readers, or external public can understand the context, goals, and how to use the system. It is of medium priority. Figure A.3 displays an example of what content this page should have.

Functional Requirements

REQ-1 Include presentation of the platform

REQ-2 Showcase latest published summaries

REQ-3 If the collaborator is logged in, display his submitted summaries

A.3.3.2 Match page

Description and Priority The match page (Figure A.4) is another fundamental component in the system since in many cases, it's the way users access all of the information regarding a match. Therefore, this page should both include methods for collaborators to generate a summary, as well as to access the summary if it exists. Thus, the priority for these requirements is high.

Functional Requirements

REQ-1 Include button to request the generation of a summary

REQ-2 Include an interface to access the summary, if it exists

REQ-3 If a summary already exists, the previous button should redirect to that summary

A.3.3.3 Editor

Description and Priority The summary editor (Figure A.5) is accessed after requesting the generation of a match summary, starts pre-filled with Prosebot's text and allows users to make changes before submitting. Once again, the priority for this component is high.

Functional Requirements

REQ-1 Should be pre-filled with text generated by Prosebot

REQ-2 Should allow users to freely edit the text

REQ-3 Should allow users to access the initial draft at any time

REQ-4 Should let users include links to players and teams from *zerozero.pt*

A.3.3.4 Post-submission survey

Description and Priority The post-submission survey (Figure A.6) should appear after a summary is submitted and will be used to collect feedback from the platform's users on the quality and relevancy of the platform. This is fundamental for future improvements to the system, therefore its priority is high.

Functional Requirements

REQ-1 Questions should be optional;

REQ-2 Likert-scale should be easily understandable;

REQ-3 Should always appear after a summary is submitted;

REQ-4 Users may skip the survey.

A.3.3.5 Summary page

Description and Priority After the summary is submitted, it will be automatically published to *zerozero.pt*. Figure A.7 shows that this page is similar to other news pages in *zerozero.pt*, with the exception of the disclaimer and rating sections. The priority is, of course, high.

Functional Requirements

REQ-1 Users can rate on relevancy and quality using a 5-point Likert scale;

REQ-2 Users may comment on the summary;

REQ-3 The summary should show a disclaimer mentioning the author and the involvement of an NLG system;

REQ-4 If the summary is being accessed by its author, he should be able to make changes by again accessing the editor;

REQ-5 Summary should be accessible from the match page.

A.3.3.6 Moderation dashboard

Description and Priority The moderation dashboard (Figure A.8) will allow *zerozero.pt*'s moderators to track the summaries published to the platform and perform actions such as summary removal. Since it is important to maintain the quality of the texts, the priority is high.

Functional Requirements

REQ-1 List the submitted summaries;

REQ-2 Display information regarding each summary (match, date, text, ratings, versions);

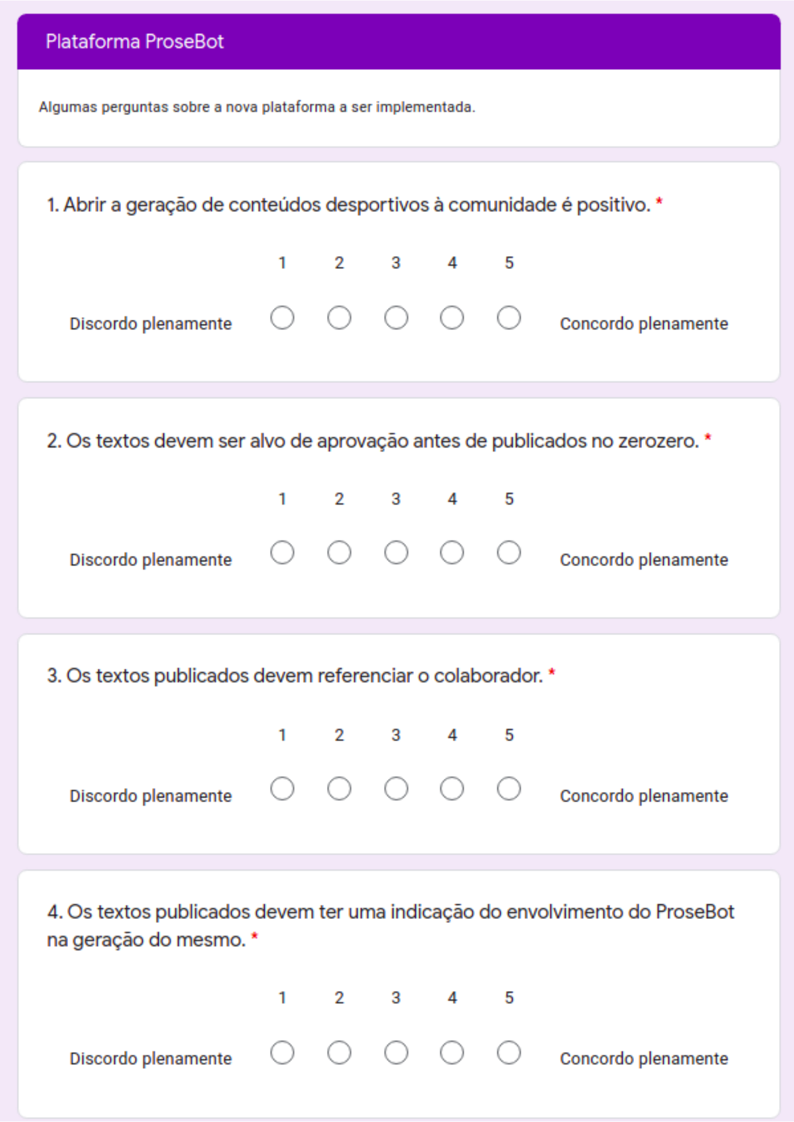
REQ-3 Allow the removal of summaries.

Appendix B

Surveys

B.1 Introduction of Automatic Tools in *zerozero.pt*'s Newsroom

A first survey conducted with journalists from *zerozero.pt*. The first section is aimed at collecting the journalist's opinion on the Prosebot platform, including decisions such as what should be the name of the texts generated with this platform. The second section intends to understand the journalist's perception of the usage of automatic tools in the newsroom.



Plataforma ProseBot

Algumas perguntas sobre a nova plataforma a ser implementada.

1. Abrir a geração de conteúdos desportivos à comunidade é positivo. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

2. Os textos devem ser alvo de aprovação antes de publicados no zerozero. *

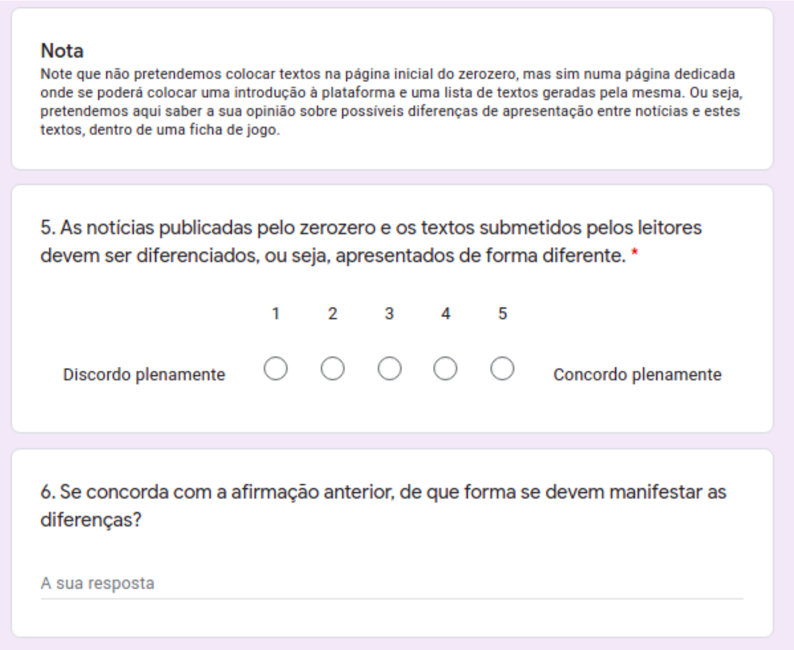
	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

3. Os textos publicados devem referenciar o colaborador. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

4. Os textos publicados devem ter uma indicação do envolvimento do ProseBot na geração do mesmo. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente



Nota

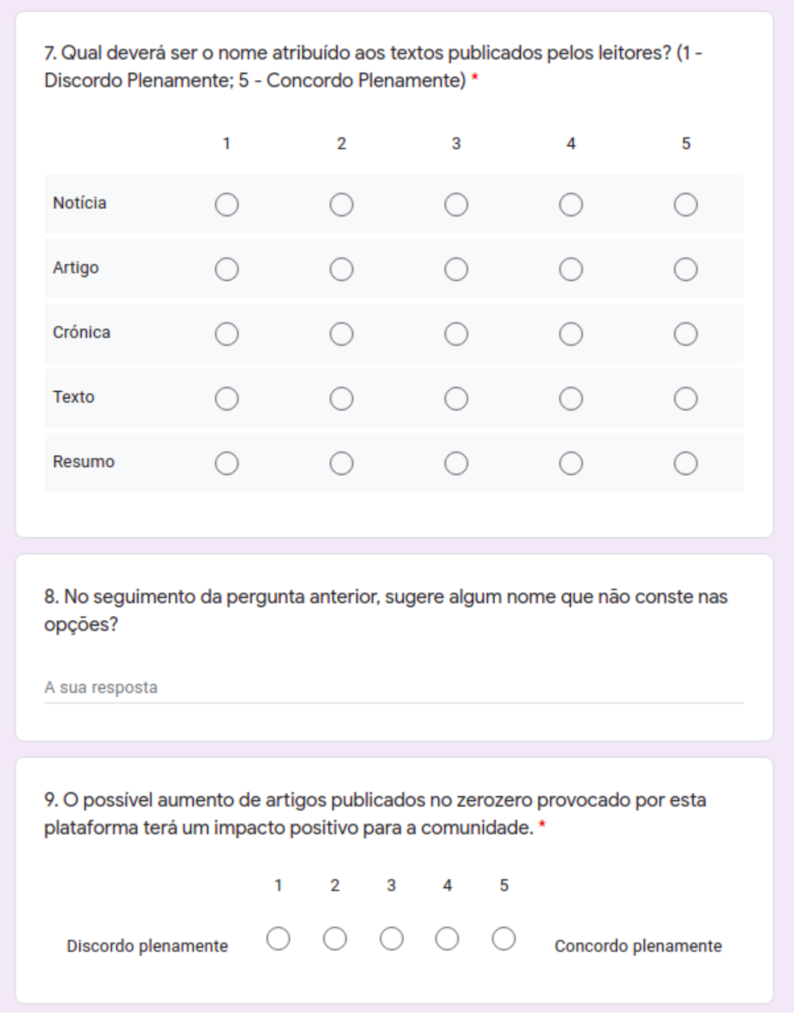
Note que não pretendemos colocar textos na página inicial do zerozero, mas sim numa página dedicada onde se poderá colocar uma introdução à plataforma e uma lista de textos geradas pela mesma. Ou seja, pretendemos aqui saber a sua opinião sobre possíveis diferenças de apresentação entre notícias e estes textos, dentro de uma ficha de jogo.

5. As notícias publicadas pelo zerozero e os textos submetidos pelos leitores devem ser diferenciados, ou seja, apresentados de forma diferente. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

6. Se concorda com a afirmação anterior, de que forma se devem manifestar as diferenças?

A sua resposta



7. Qual deverá ser o nome atribuído aos textos publicados pelos leitores? (1 - Discordo Plenamente; 5 - Concordo Plenamente) *

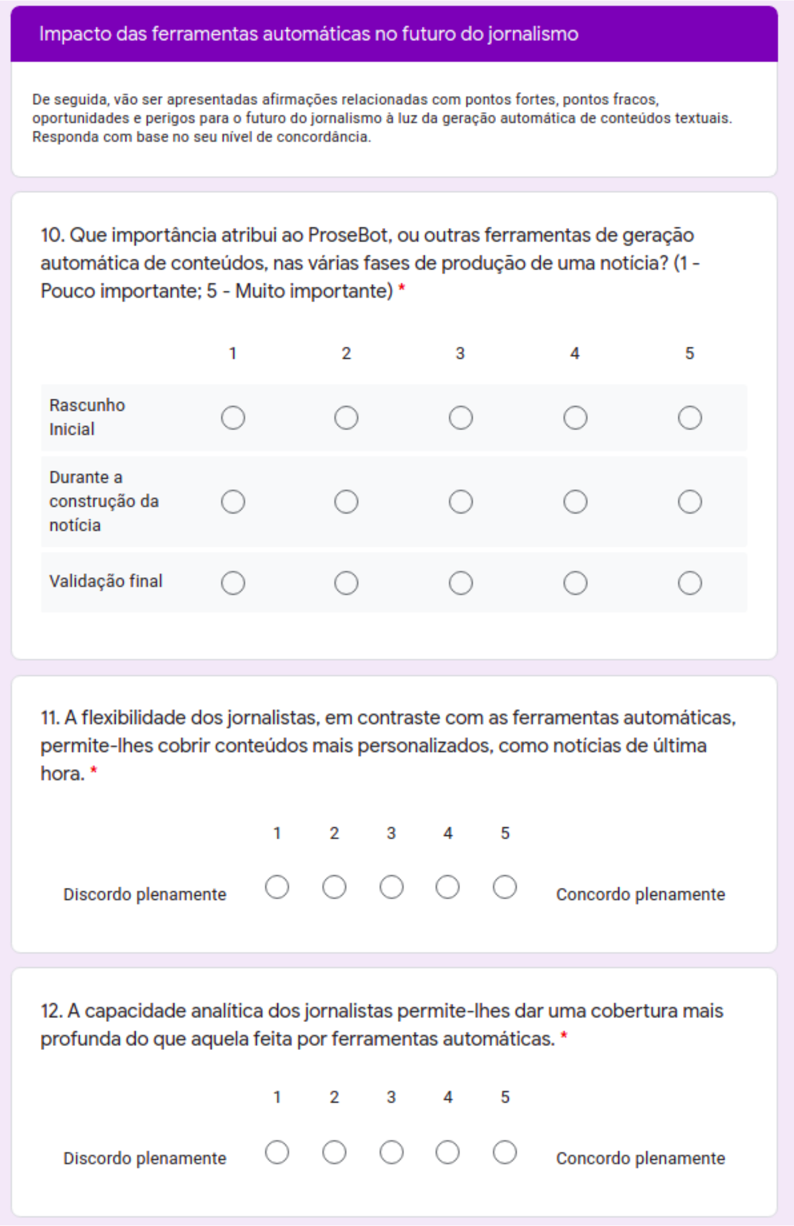
	1	2	3	4	5
Notícia	○	○	○	○	○
Artigo	○	○	○	○	○
Crónica	○	○	○	○	○
Texto	○	○	○	○	○
Resumo	○	○	○	○	○

8. No seguimento da pergunta anterior, sugere algum nome que não conste nas opções?

A sua resposta

9. O possível aumento de artigos publicados no zerozero provocado por esta plataforma terá um impacto positivo para a comunidade. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente



Impacto das ferramentas automáticas no futuro do jornalismo

De seguida, vão ser apresentadas afirmações relacionadas com pontos fortes, pontos fracos, oportunidades e perigos para o futuro do jornalismo à luz da geração automática de conteúdos textuais. Responda com base no seu nível de concordância.

10. Que importância atribui ao ProseBot, ou outras ferramentas de geração automática de conteúdos, nas várias fases de produção de uma notícia? (1 - Pouco importante; 5 - Muito importante) *

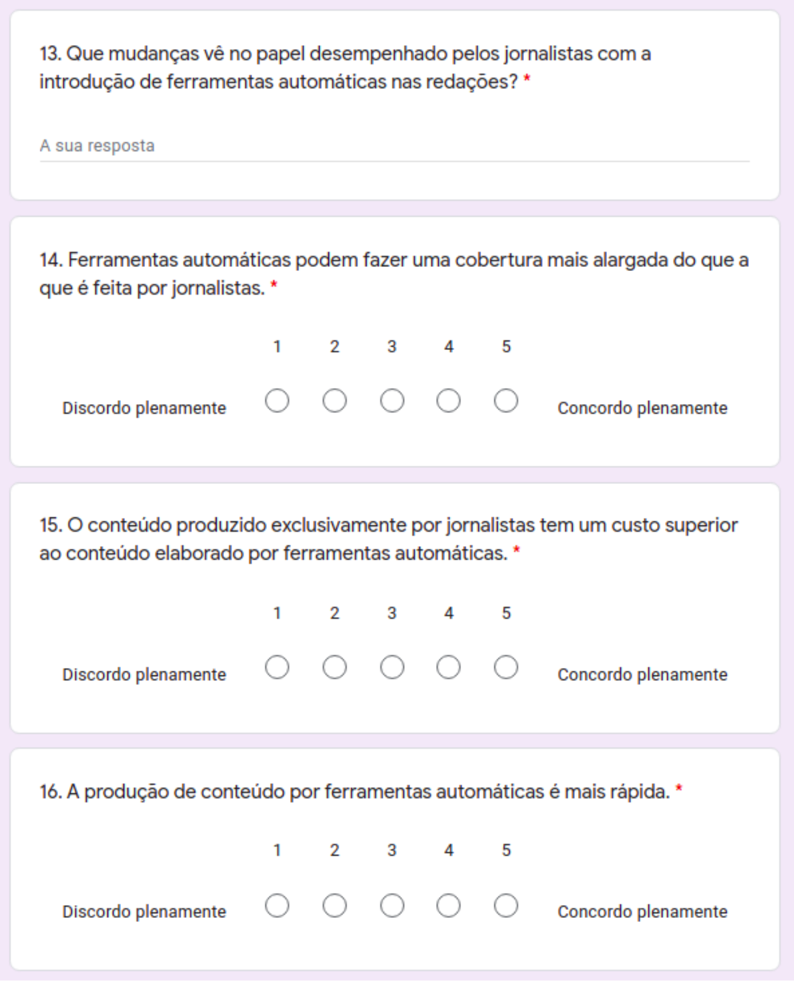
	1	2	3	4	5
Rascunho Inicial	○	○	○	○	○
Durante a construção da notícia	○	○	○	○	○
Validação final	○	○	○	○	○

11. A flexibilidade dos jornalistas, em contraste com as ferramentas automáticas, permite-lhes cobrir conteúdos mais personalizados, como notícias de última hora. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

12. A capacidade analítica dos jornalistas permite-lhes dar uma cobertura mais profunda do que aquela feita por ferramentas automáticas. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente



13. Que mudanças vê no papel desempenhado pelos jornalistas com a introdução de ferramentas automáticas nas redações? *

A sua resposta

14. Ferramentas automáticas podem fazer uma cobertura mais alargada do que a que é feita por jornalistas. *

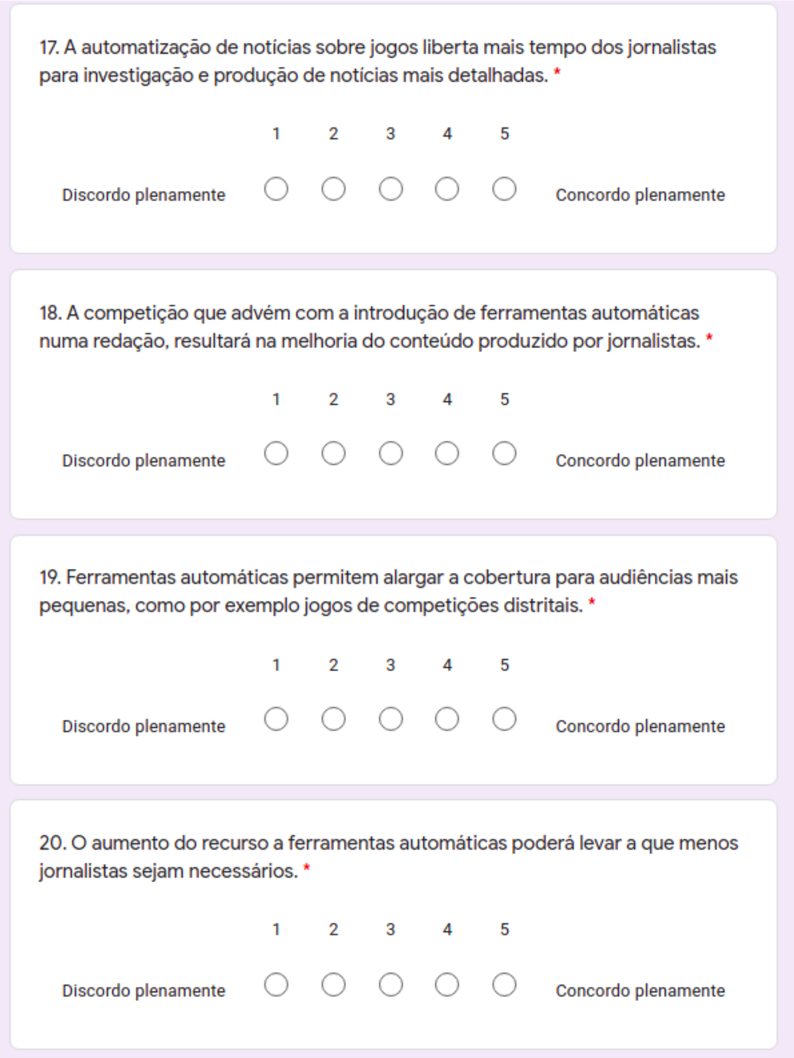
	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

15. O conteúdo produzido exclusivamente por jornalistas tem um custo superior ao conteúdo elaborado por ferramentas automáticas. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

16. A produção de conteúdo por ferramentas automáticas é mais rápida. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente



17. A automatização de notícias sobre jogos liberta mais tempo dos jornalistas para investigação e produção de notícias mais detalhadas. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

18. A competição que advém com a introdução de ferramentas automáticas numa redação, resultará na melhoria do conteúdo produzido por jornalistas. *

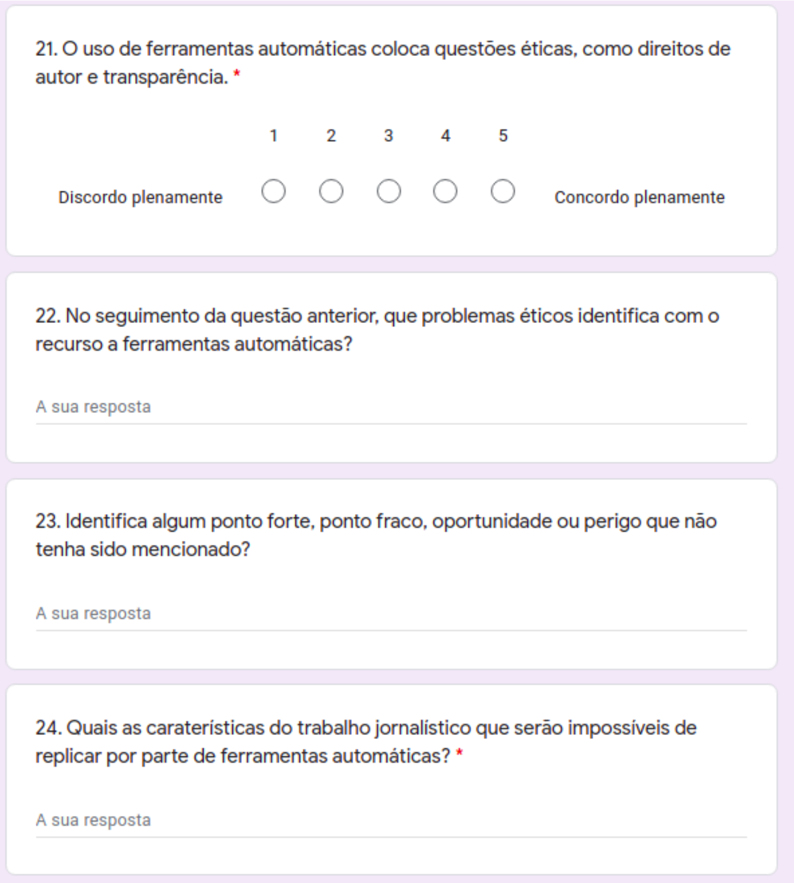
	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

19. Ferramentas automáticas permitem alargar a cobertura para audiências mais pequenas, como por exemplo jogos de competições distritais. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

20. O aumento do recurso a ferramentas automáticas poderá levar a que menos jornalistas sejam necessários. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente



21. O uso de ferramentas automáticas coloca questões éticas, como direitos de autor e transparência. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

22. No seguimento da questão anterior, que problemas éticos identifica com o recurso a ferramentas automáticas?

A sua resposta

23. Identifica algum ponto forte, ponto fraco, oportunidade ou perigo que não tenha sido mencionado?

A sua resposta

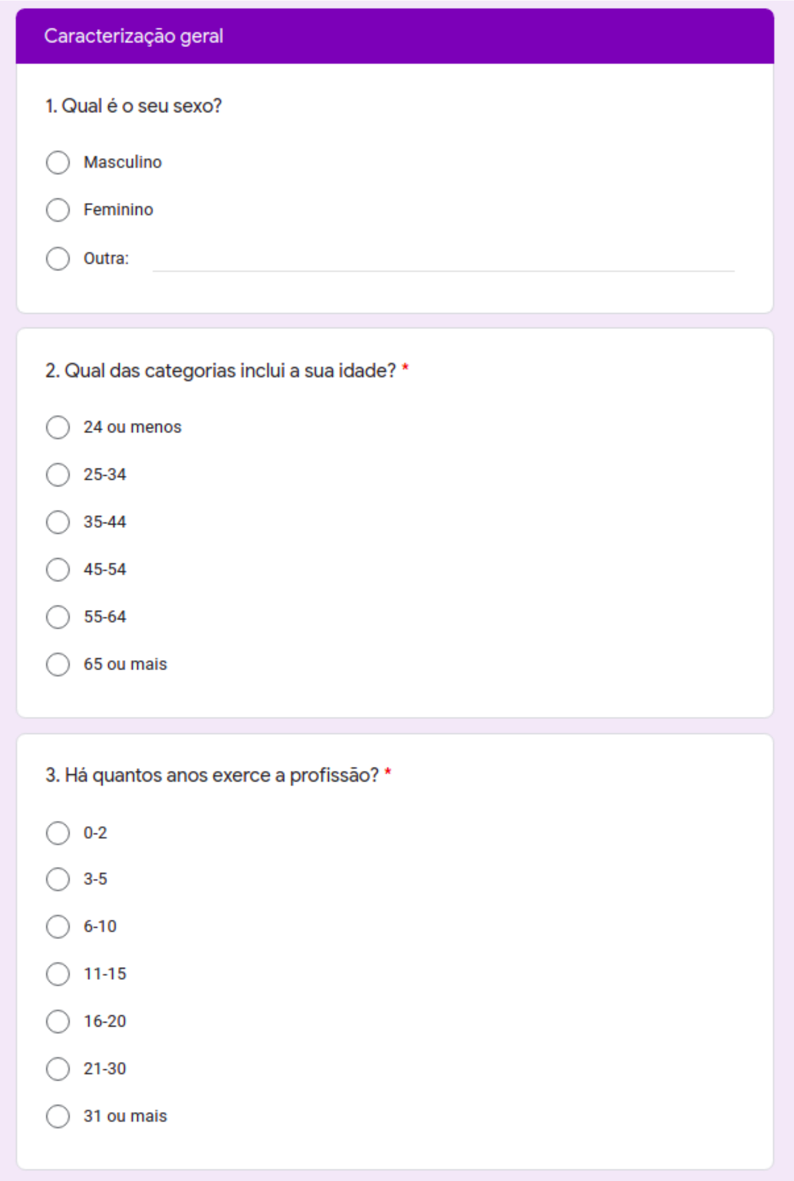
24. Quais as caraterísticas do trabalho jornalístico que serão impossíveis de replicar por parte de ferramentas automáticas? *

A sua resposta

Figure B.1: Survey on the impact of automatic tools *zerozero.pt*'s newsroom and the journalists' opinion on the Prosebot platform.

B.2 Impact of Automatic Tools on the Future of Journalism

A second survey on the impact of automatic tools was conducted with Portuguese journalists. The first section includes general characterization questions such as age and fields of expertise, while the second section asks questions on the perception of journalists on the usage of automatic tools, similar to the first survey.



Caracterização geral

1. Qual é o seu sexo?

○ Masculino

○ Feminino

○ Outra: ____________________

2. Qual das categorias inclui a sua idade? *

○ 24 ou menos

○ 25-34

○ 35-44

○ 45-54

○ 55-64

○ 65 ou mais

3. Há quantos anos exerce a profissão? *

○ 0-2

○ 3-5

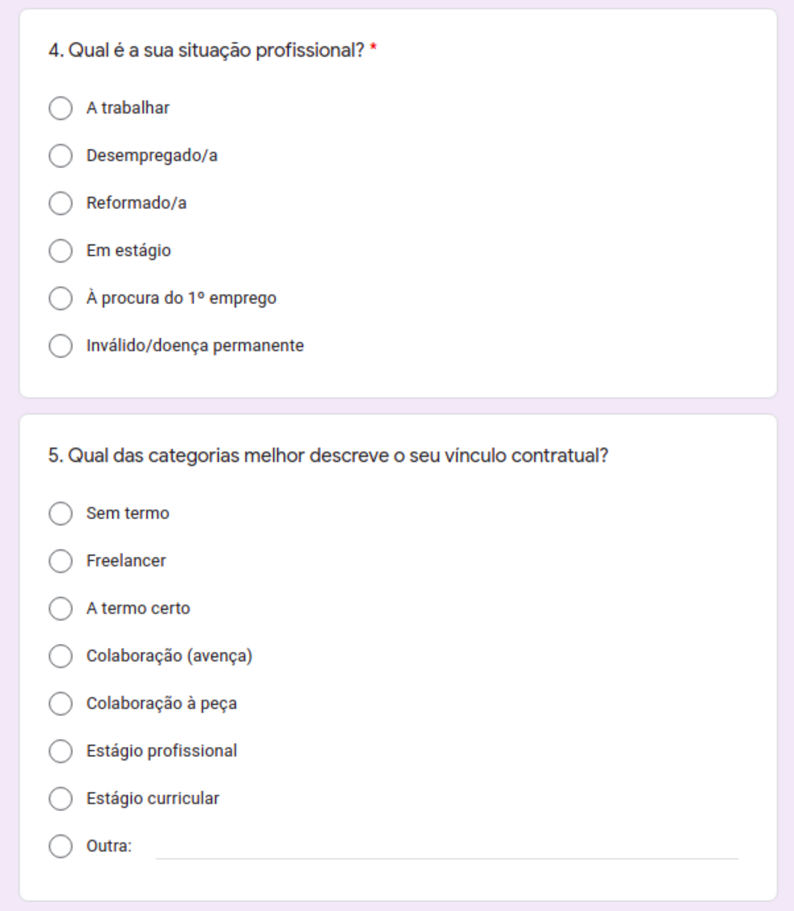
○ 6-10

○ 11-15

○ 16-20

○ 21-30

○ 31 ou mais



4. Qual é a sua situação profissional? *

○ A trabalhar

○ Desempregado/a

○ Reformado/a

○ Em estágio

○ À procura do 1º emprego

○ Inválido/doença permanente

5. Qual das categorias melhor descreve o seu vínculo contratual?

○ Sem termo

○ Freelancer

○ A termo certo

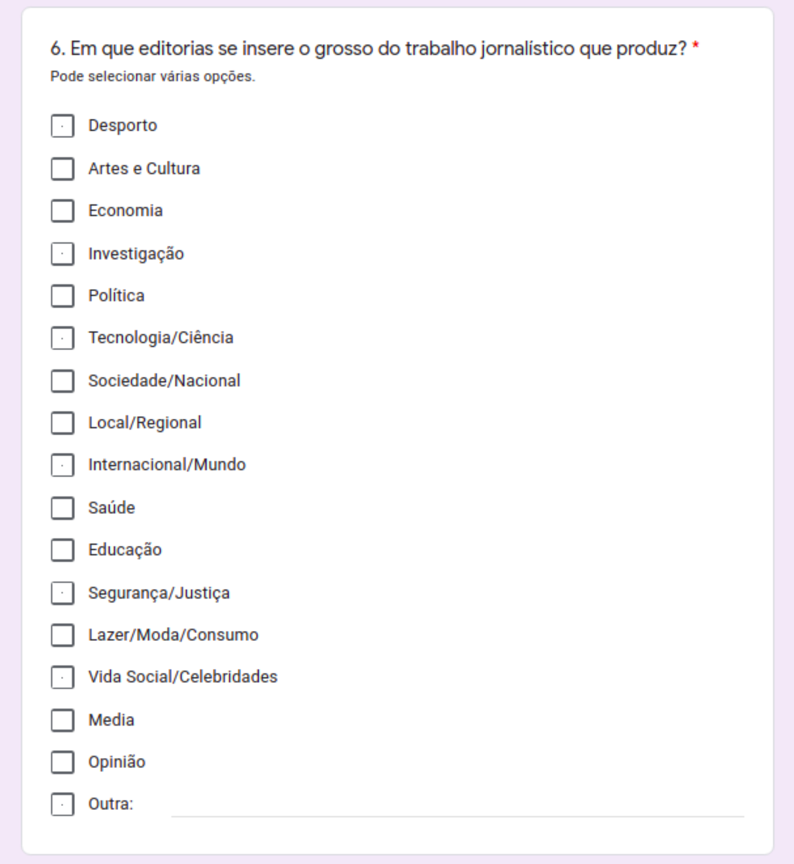
○ Colaboração (avença)

○ Colaboração à peça

○ Estágio profissional

○ Estágio curricular

○ Outra: ____________________



6. Em que editorias se insere o grosso do trabalho jornalístico que produz? *

Pode selecionar várias opções.

□ Desporto

□ Artes e Cultura

□ Economia

□ Investigação

□ Política

□ Tecnologia/Ciência

□ Sociedade/Nacional

□ Local/Regional

□ Internacional/Mundo

□ Saúde

□ Educação

□ Segurança/Justiça

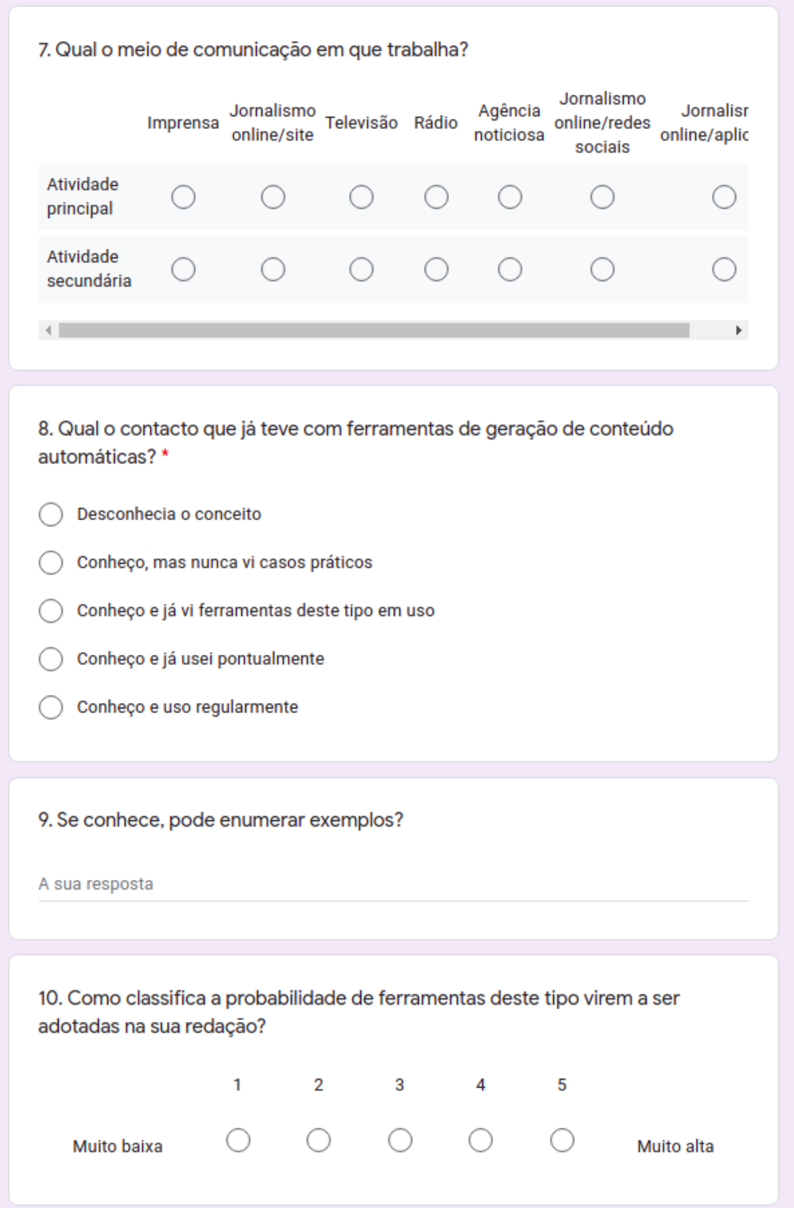
□ Lazer/Moda/Consumo

□ Vida Social/Celebridades

□ Media

□ Opinião

□ Outra: ____________________



7. Qual o meio de comunicação em que trabalha?

	Imprensa	Jornalismo online/site	Televisão	Rádio	Agência noticiosa	Jornalismo online/redes sociais	Jornalisr online/aplic
Atividade principal	○	○	○	○	○	○	○
Atividade secundária	○	○	○	○	○	○	○

8. Qual o contacto que já teve com ferramentas de geração de conteúdo automáticas? *

○ Desconhecia o conceito

○ Conheço, mas nunca vi casos práticos

○ Conheço e já vi ferramentas deste tipo em uso

○ Conheço e já usei pontualmente

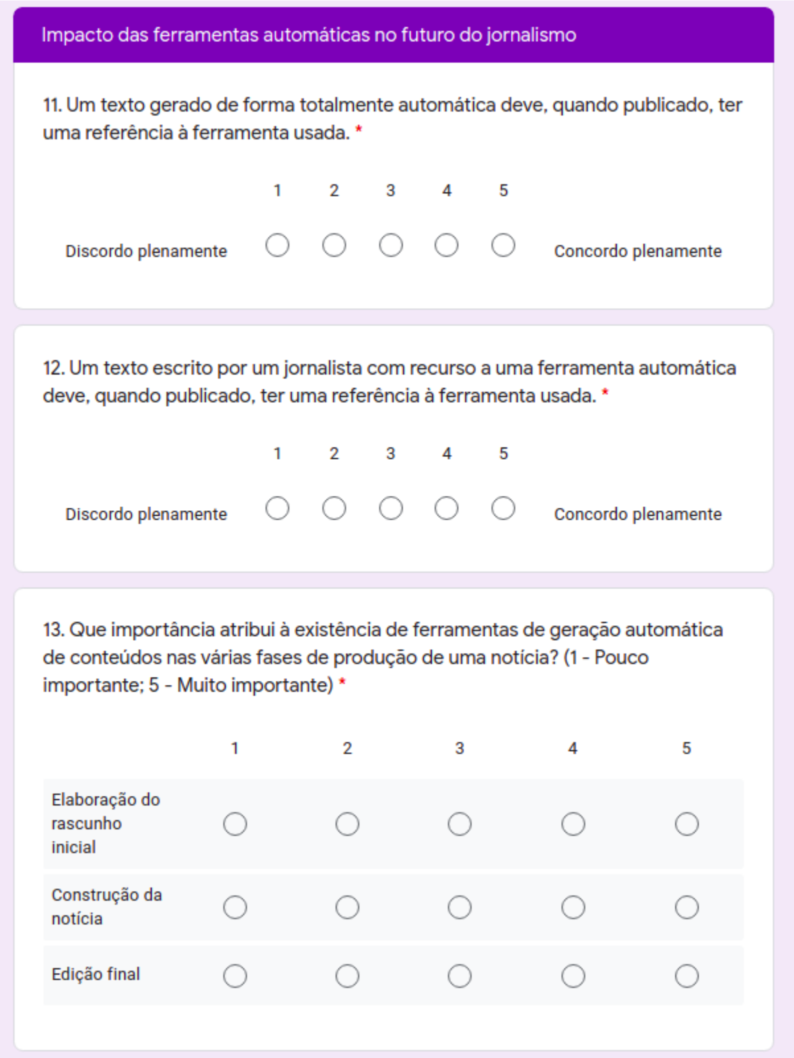
○ Conheço e uso regularmente

9. Se conhece, pode enumerar exemplos?

A sua resposta

10. Como classifica a probabilidade de ferramentas deste tipo virem a ser adotadas na sua redação?

	1	2	3	4	5	
Muito baixa	○	○	○	○	○	Muito alta



Impacto das ferramentas automáticas no futuro do jornalismo

11. Um texto gerado de forma totalmente automática deve, quando publicado, ter uma referência à ferramenta usada. *

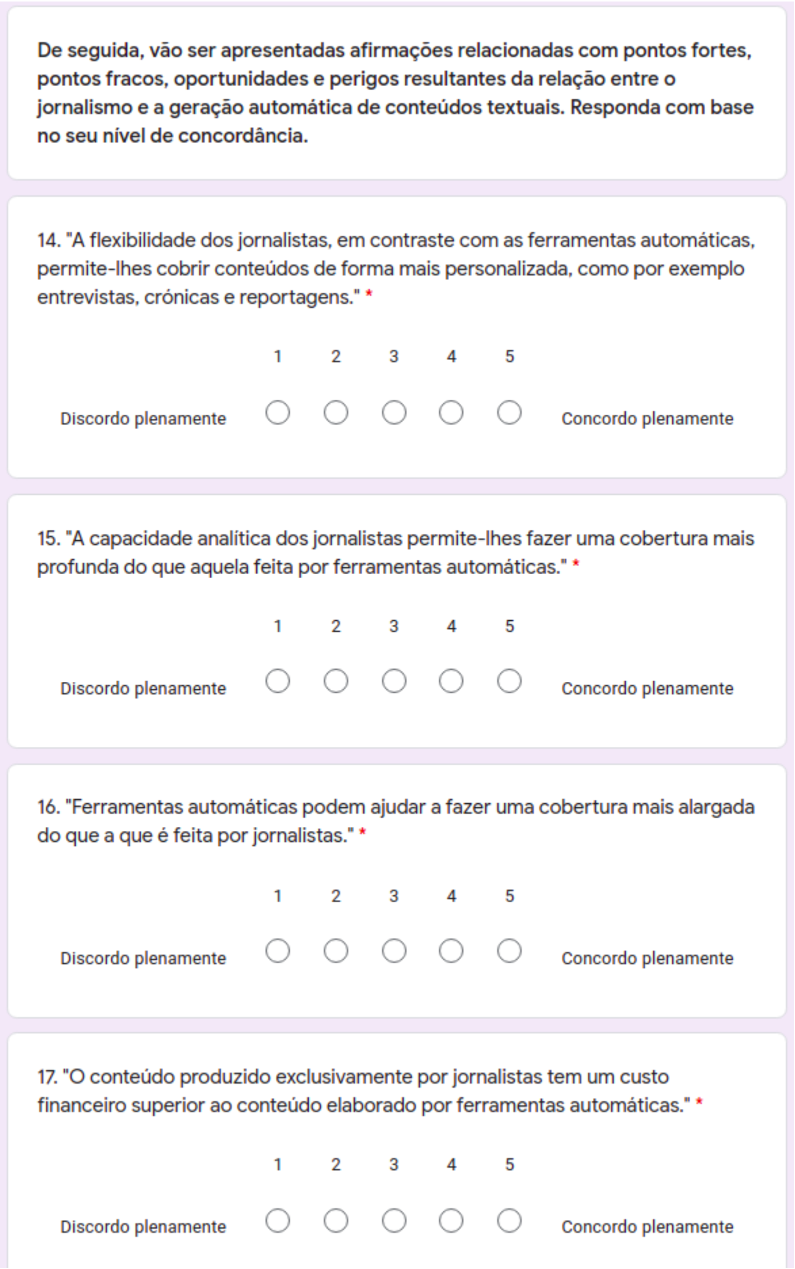
	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

12. Um texto escrito por um jornalista com recurso a uma ferramenta automática deve, quando publicado, ter uma referência à ferramenta usada. *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

13. Que importância atribui à existência de ferramentas de geração automática de conteúdos nas várias fases de produção de uma notícia? (1 - Pouco importante; 5 - Muito importante) *

	1	2	3	4	5
Elaboração do rascunho inicial	○	○	○	○	○
Construção da notícia	○	○	○	○	○
Edição final	○	○	○	○	○



De seguida, vão ser apresentadas afirmações relacionadas com pontos fortes, pontos fracos, oportunidades e perigos resultantes da relação entre o jornalismo e a geração automática de conteúdos textuais. Responda com base no seu nível de concordância.

14. "A flexibilidade dos jornalistas, em contraste com as ferramentas automáticas, permite-lhes cobrir conteúdos de forma mais personalizada, como por exemplo entrevistas, crónicas e reportagens." *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

15. "A capacidade analítica dos jornalistas permite-lhes fazer uma cobertura mais profunda do que aquela feita por ferramentas automáticas." *

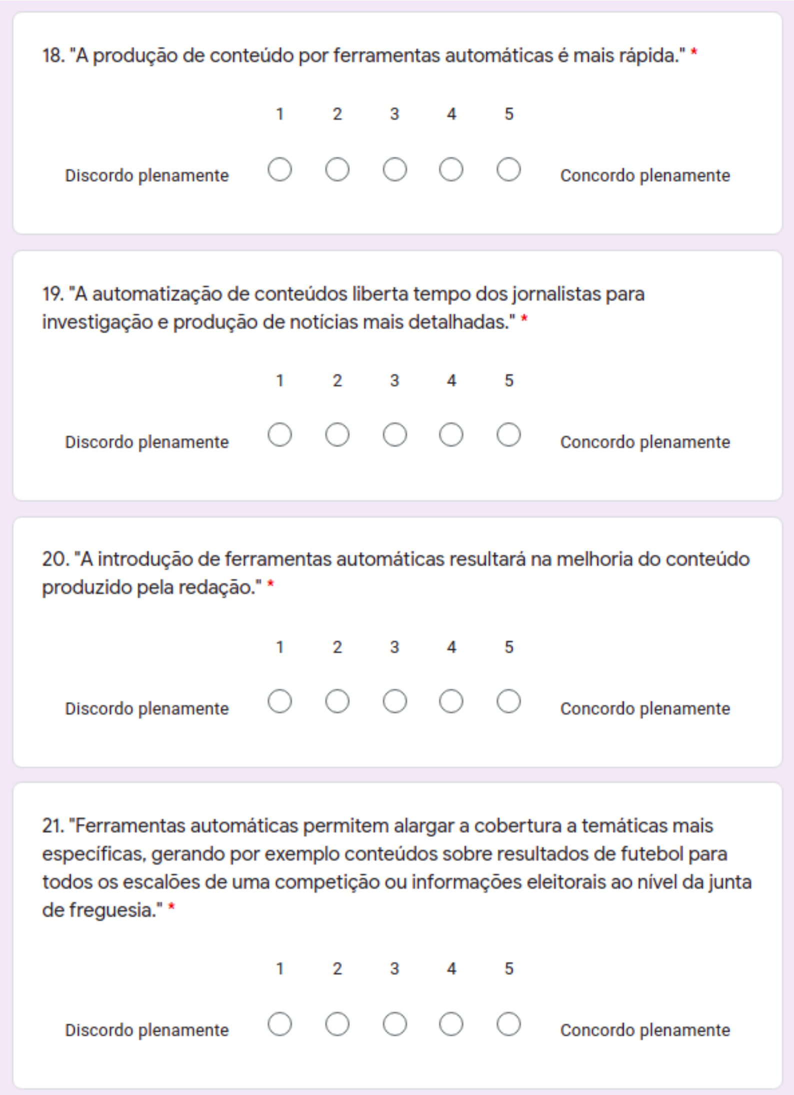
	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

16. "Ferramentas automáticas podem ajudar a fazer uma cobertura mais alargada do que a que é feita por jornalistas." *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

17. "O conteúdo produzido exclusivamente por jornalistas tem um custo financeiro superior ao conteúdo elaborado por ferramentas automáticas." *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente



18. "A produção de conteúdo por ferramentas automáticas é mais rápida." *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

19. "A automatização de conteúdos liberta tempo dos jornalistas para investigação e produção de notícias mais detalhadas." *

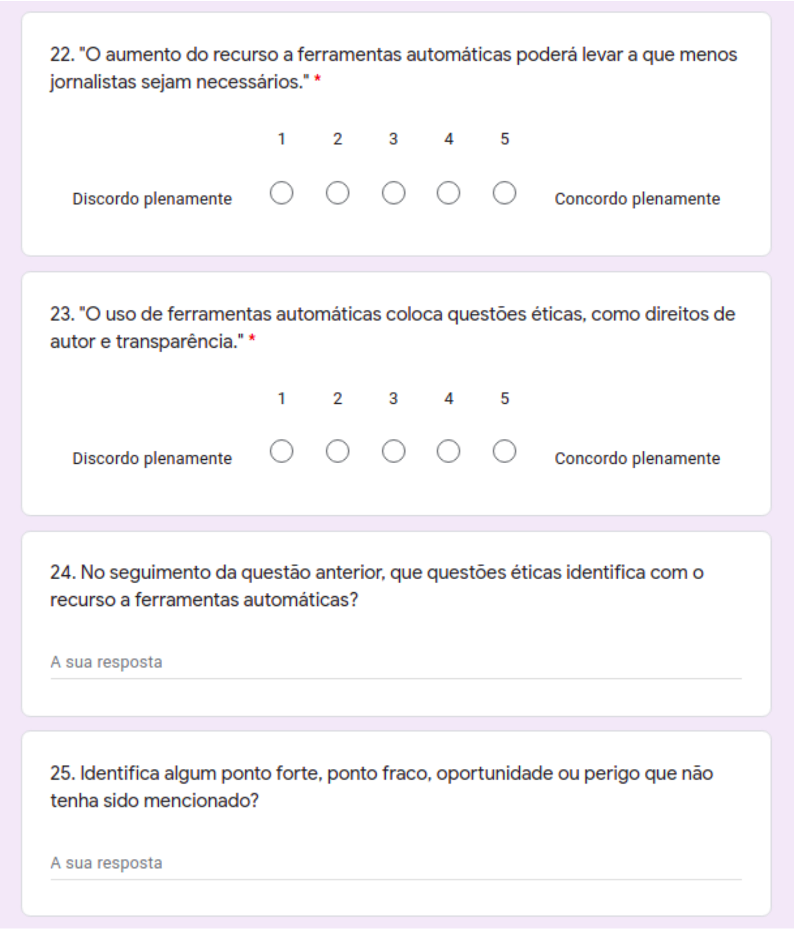
	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

20. "A introdução de ferramentas automáticas resultará na melhoria do conteúdo produzido pela redação." *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

21. "Ferramentas automáticas permitem alargar a cobertura a temáticas mais específicas, gerando por exemplo conteúdos sobre resultados de futebol para todos os escalões de uma competição ou informações eleitorais ao nível da junta de freguesia." *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente



22. "O aumento do recurso a ferramentas automáticas poderá levar a que menos jornalistas sejam necessários." *

	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

23. "O uso de ferramentas automáticas coloca questões éticas, como direitos de autor e transparência." *

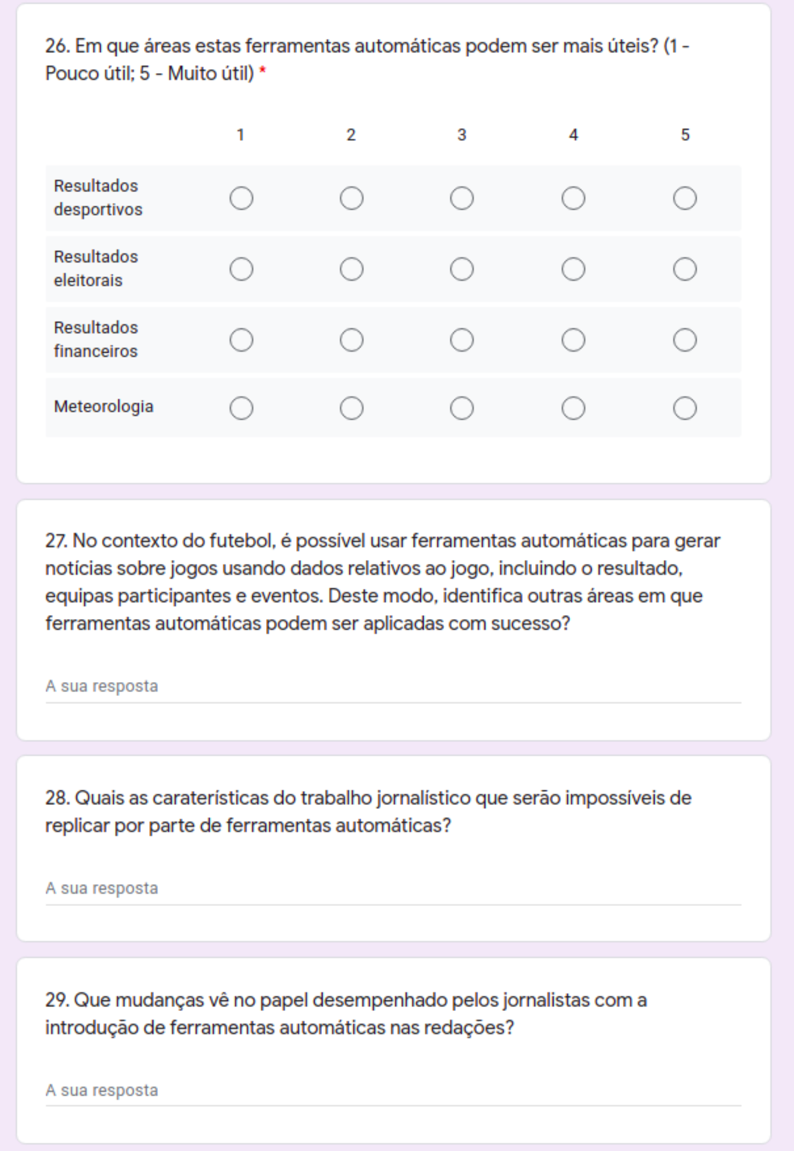
	1	2	3	4	5	
Discordo plenamente	○	○	○	○	○	Concordo plenamente

24. No seguimento da questão anterior, que questões éticas identifica com o recurso a ferramentas automáticas?

A sua resposta

25. Identifica algum ponto forte, ponto fraco, oportunidade ou perigo que não tenha sido mencionado?

A sua resposta



26. Em que áreas estas ferramentas automáticas podem ser mais úteis? (1 - Pouco útil; 5 - Muito útil) *

	1	2	3	4	5
Resultados desportivos	○	○	○	○	○
Resultados eleitorais	○	○	○	○	○
Resultados financeiros	○	○	○	○	○
Meteorologia	○	○	○	○	○

27. No contexto do futebol, é possível usar ferramentas automáticas para gerar notícias sobre jogos usando dados relativos ao jogo, incluindo o resultado, equipas participantes e eventos. Deste modo, identifica outras áreas em que ferramentas automáticas podem ser aplicadas com sucesso?

A sua resposta

28. Quais as caraterísticas do trabalho jornalístico que serão impossíveis de replicar por parte de ferramentas automáticas?

A sua resposta

29. Que mudanças vê no papel desempenhado pelos jornalistas com a introdução de ferramentas automáticas nas redações?

A sua resposta

Figure B.2: General survey on the impact of automatic tools on the future of journalism.

References

[1] João Pinto Barbosa Machado Aires. Automatic generation of sports news. Master's thesis, Faculty of Engineering of the University of Porto, 2016.

[2] Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. STORIUM: A dataset and evaluation platform for machine-in-the-loop story generation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6470–6484. Association for Computational Linguistics, 2020. URL https://www.aclweb.org/anthology/2020.emnlp-main.525/.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

[4] Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare R. Voss, editors, *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics, 2005. URL https://www.aclweb.org/anthology/W05-0909/.

[5] Nadine Braun, Martijn Goudbeek, and Emiel Krahmer. The multilingual affective soccer corpus (MASC): compiling a biased parallel corpus on soccer reportage in english, german and dutch. In Amy Isard, Verena Rieser, and Dimitra Gkatzia, editors, *INLG 2016 - Proceedings of the Ninth International Natural Language Generation Conference, September 5-8, 2016, Edinburgh, UK*, pages 74–78. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/w16-6612. URL https://doi.org/10.18653/v1/w16-6612.

[6] John Brooke. *"SUS-A quick and dirty usability scale." Usability evaluation in industry*. CRC Press, June 1996. URL https://www.crcpress.com/product/isbn/9780748404605. ISBN: 9780748404605.

[7] David L. Chen and Raymond J. Mooney. Learning to sportscast: a test of grounded language acquisition. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 128–135. ACM, 2008. doi: 10.1145/1390156.1390173. URL https://doi.org/10.1145/1390156.1390173.

[8] Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. Creative writing with a machine in the loop: Case studies on slogans and stories. In Shlomo Berkovsky, Yoshinori Hijikata, Jun Rekimoto, Margaret M. Burnett, Mark Billinghurst, and Aaron Quigley, editors, *Proceedings of the 23rd International Conference on Intelligent User Interfaces, IUI 2018, Tokyo, Japan, March 07-11, 2018*, pages 329–340. ACM, 2018. doi: 10.1145/3172944.3172983. URL https://doi.org/10.1145/3172944.3172983.

[9] Luís Miguel Cardoso Lopes Correia. Evaluation metrics for text and creation of writing tool for sports journalism. Master's thesis, Faculty of Engineering of the University of Porto, 2020.

[10] Mário Crespo, Joana Azevedo, João Carlos Sousa, Gustavo Cardoso, and Miguel Paisana. Jornalistas e condições laborais: retrato de uma profissão em transformação. In *Relatórios OBERCOM fevereiro 2017*. OBERCOM, 2017.

[11] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26 (3):297–302, 1945. doi: https://doi.org/10.2307/1932409. URL https://esajournals.onlinelibrary.wiley.com/doi/abs/10.2307/1932409.

[12] Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, 61:65–170, 2018. doi: 10.1613/jair.5477. URL https://doi.org/10.1613/jair.5477.

[13] Alexander Koller, Kristina Striegnitz, Andrew Gargett, Donna Byron, Justine Cassell, Robert Dale, Johanna D. Moore, and Jon Oberlander. Report on the second NLG challenge on generating instructions in virtual environments (GIVE-2). In John D. Kelleher, Brian Mac Namee, Ielka van der Sluis, Anja Belz, Albert Gatt, and Alexander Koller, editors, *INLG 2010 - Proceedings of the Sixth International Natural Language Generation Conference, July 7-9, 2010, Trim, Co. Meath, Ireland*. The Association for Computer Linguistics, 2010. URL https://www.aclweb.org/anthology/W10-4233/.

[14] Jessica Kunert. Automation in sports reporting: Strategies of data providers, software providers, and media outlets. *Media and Communication*, 8(3):5–15, 2020. ISSN 2183-2439. doi: 10.17645/mac.v8i3.2996. URL https://www.cogitatiopress.com/mediaandcommunication/article/view/2996.

[15] Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. Data-driven news generation for automated journalism. In José Maria Alonso, Alberto Bugarín, and Ehud Reiter, editors, *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, pages 188–197. Association for Computational Linguistics, 2017. doi: 10.18653/v1/w17-3528. URL https://doi.org/10.18653/v1/w17-3528.

[16] Chin-Yew Lin and Eduard H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In Marti A. Hearst and Mari Ostendorf, editors, *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics, 2003. URL https://www.aclweb.org/anthology/N03-1020/.

[17] Lusa. Bolsa de lisboa fechou a sessão de hoje em terreno negativo. SAPO24. https://24.sapo.pt/noticias/bolsa-de-lisboa-fechou-a-sessao-de-hoje-em_5de69a45ef78a379890e25f9, 2019. Accessed: 2021-30-05.

[18] Conor Molumby and Joe Whitwell. General election 2019: Semi-automation makes it a night of 689 stories. BBC. https://bbcnewslabs.co.uk/news/2019/salco-ge/, 2019. Accessed: 2021-01-02.

[19] Flávio Nunes. Robôs também já escrevem notícias em portugal. estão a estagiar na lusa. ECO. https://eco.sapo.pt/especiais/robos-tambem-ja-escrevem-noticias-em-portugal-estao-a-estagiar-na-lusa/, 2019. Accessed: 2021-30-05.

[20] Will Oremus. The first news report on the l.a. earthquake was written by a robot. Slate Magazine. https://slate.com/technology/2014/03/quakebot-los-angeles-times-robot-journalist-writes-article-on-la-earthquake.html, Mar 2014. Accessed: 2021-01-15.

[21] 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 of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040/.

[22] Karl Pearson. Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187):253–318, 1896.

[23] Vassilis Plachouras, Charese Smiley, Hiroko Bretz, Ola Taylor, Jochen L. Leidner, Dezhao Song, and Frank Schilder. Interacting with financial data using natural language. In Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel, editors, *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 1121–1124. ACM, 2016. doi: 10.1145/2911451.2911457. URL https://doi.org/10.1145/2911451.2911457.

[24] Alec Radford. Better language models and their implications. OpenAI. https://openai.com/blog/better-language-models/, Sep 2020. Accessed: 2021-02-09.

[25] Ehud Reiter. An architecture for data-to-text systems. In Stephan Busemann, editor, *Proceedings of the Eleventh European Workshop on Natural Language Generation, ENLG 2007, Schloss Dagstuhl, Germany, June 17-20, 2007*, 2007. URL https://www.aclweb.org/anthology/W07-2315/.

[26] Ehud Reiter. Human editing of NLG texts. Ehud Reiter's Blog. https://ehudreiter.com/2020/06/08/human-editing-of-nlg-texts/, 2020. Accessed: 2021-01-15.

[27] Ehud Reiter and Anja Belz. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Comput. Linguistics*, 35(4):529–558, 2009. doi: 10.1162/coli.2009.35.4.35405. URL https://doi.org/10.1162/coli.2009.35.4.35405.

[28] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press, 2000. doi: 10.1017/CBO9780511519857.

[29] Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. Choosing words in computer-generated weather forecasts. *Artif. Intell.*, 167(1-2):137–169, 2005. doi: 10.1016/j.artint.2005.06.006. URL https://doi.org/10.1016/j.artint.2005.06.006.

[30] Vasco Ferreira Ribeiro. Jornalista-robot: produção automática de conteúdos de texto como apoio ao jornalismo desportivo. Master's thesis, Faculty of Engineering of the University of Porto, 2019.

[31] Melissa Roemmele and Andrew S. Gordon. Creative help: A story writing assistant. In Henrik Schoenau-Fog, Luis Emilio Bruni, Sandy Louchart, and Sarune Baceviciute, editors, *Interactive Storytelling - 8th International Conference on Interactive Digital Storytelling, ICIDS 2015, Copenhagen, Denmark, November 30 - December 4, 2015, Proceedings*, volume 9445 of *Lecture Notes in Computer Science*, pages 81–92. Springer, 2015. doi: 10.1007/978-3-319-27036-4_8. URL https://doi.org/10.1007/978-3-319-27036-4_8.

[32] Ben Samuel, Michael Mateas, and Noah Wardrip-Fruin. The design of writing buddy: A mixed-initiative approach towards computational story collaboration. In Frank Nack and Andrew S. Gordon, editors, *Interactive Storytelling - 9th International Conference on Interactive Digital Storytelling, ICIDS 2016, Los Angeles, CA, USA, November 15-18, 2016, Proceedings*, volume 10045 of *Lecture Notes in Computer Science*, pages 388–396, 2016. doi: 10.1007/978-3-319-48279-8_34. URL https://doi.org/10.1007/978-3-319-48279-8_34.

[33] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1162. URL https://doi.org/10.18653/v1/p16-1162.

[34] João Ricardo Pintas Soares. Statistical language models applied to news generation. Master's thesis, Faculty of Engineering of the University of Porto, 2017.

[35] Somayajulu Sripada, Ehud Reiter, and Lezan Hawizy. Evaluation of an NLG system using post-edit data: Lessons learnt. In Graham Wilcock, Kristiina Jokinen, Chris Mellish, and Ehud Reiter, editors, *Proceedings of the Tenth European Workshop on Natural Language Generation, ENLG 2005, Aberdeen, UK, August 8-10, 2005*. ACL, 2005. URL https://www.aclweb.org/anthology/W05-1615/.

[36] Reid Swanson and Andrew S. Gordon. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Trans. Interact. Intell. Syst.*, 2(3):16:1–16:35, 2012. doi: 10.1145/2362394.2362398. URL https://doi.org/10.1145/2362394.2362398.

[37] Mariët Theune, Esther Klabbers, Jan-Roelof de Pijper, Emiel Krahmer, and Jan Odijk. From data to speech: a general approach. *Nat. Lang. Eng.*, 7(1):47–86, 2001. URL http://journals.cambridge.org/action/displayAbstract?aid=73673.

[38] Craig Thomson and Ehud Reiter. A gold standard methodology for evaluating accuracy in data-to-text systems. In Brian Davis, Yvette Graham, John Kelleher, and Yaji Sripada, editors, *Proceedings of the 13th International Conference on Natural Language Generation,*

INLG 2020, Dublin, Ireland, December 15-18, 2020, pages 158–168. Association for Computational Linguistics, 2020. URL https://www.aclweb.org/anthology/2020.inlg-1.22/.

[39] Arjen van Dalen. The algorithms behind the headlines. *Journalism Practice*, 6(5-6):648–658, 2012. doi: 10.1080/17512786.2012.667268. URL https://doi.org/10.1080/17512786.2012.667268.

[40] Kees van Deemter, Mariët Theune, and Emiel Krahmer. Real versus template-based natural language generation: A false opposition? *Comput. Linguistics*, 31(1):15–24, 2005. doi: 10.1162/0891201053630291. URL https://doi.org/10.1162/0891201053630291.

[41] H.A.J. van der Kaa and E.J. Krahmer. Journalist versus news consumer: The perceived credibility of machine written news. In *Proceedings of the Computation+Journalism conference*, 2014. Computation + Journalism Symposium 2014 ; Conference date: 24-10-2014 Through 25-10-2014.

[42] Chris van der Lee, Emiel Krahmer, and Sander Wubben. PASS: A dutch data-to-text system for soccer, targeted towards specific audiences. In José Maria Alonso, Alberto Bugarín, and Ehud Reiter, editors, *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, pages 95–104. Association for Computational Linguistics, 2017. doi: 10.18653/v1/w17-3513. URL https://doi.org/10.18653/v1/w17-3513.

[43] Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. Best practices for the human evaluation of automatically generated text. In Kees van Deemter, Chenghua Lin, and Hiroya Takamura, editors, *Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019, Tokyo, Japan, October 29 - November 1, 2019*, pages 355–368. Association for Computational Linguistics, 2019. doi: 10.18653/v1/W19-8643. URL https://aclweb.org/anthology/papers/W/W19/W19-8643/.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[45] Mary Lynn Young and Alfred Hermida. From Mr. and Mrs. Outlier To Central Tendencies. *Digital Journalism*, 3(3):381–397, 2015. doi: 10.1080/21670811.2014.976409. URL https://doi.org/10.1080/21670811.2014.976409.

[46] R. Michael Young. Using grice's maxim of quantity to select the content of plan descriptions. *Artif. Intell.*, 115(2):215–256, 1999. doi: 10.1016/S0004-3702(99)00082-X. URL https://doi.org/10.1016/S0004-3702(99)00082-X.