

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Improving Sports Search Engine Retrieval through Multiple Signal Integration

Guilherme Pereira Garrido



Mestrado em Engenharia Informática e Computação

Supervisor: Sérgio Nunes

July 17, 2024

Improving Sports Search Engine Retrieval through Multiple Signal Integration

Guilherme Pereira Garrido

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Maria Cristina de Carvalho Alves Ribeiro, PhD

External Examiner: Prof. Ricardo Nuno Taborda Campos, PhD

Supervisor: Prof. Sérgio Sobral Nunes, PhD

July 17, 2024

Abstract

Information Retrieval (IR) focuses on finding relevant information within large digital repositories. In order to satisfy their information needs, users rely on IR systems. This work concerns the search system of zerozero, a Portuguese sports-related platform, which provides an opportunity to improve the quality of search results.

We started by analyzing the search systems of some of the most used sports-related websites in the world, focusing on their search user interfaces and how they handle popularity in search results. Our findings revealed limitations in the filtering functionality of these systems. In contrast, zerozero stands out for its complete entity coverage, control features, and detailed results information. Regarding popularity handling, we defined a set of queries that allowed us to observe whether the order of results changed due to the impact of sporting events and observed that one website presented justified changes.

We then conducted a brief evaluation to assess the impact of Apache Solr, a framework recently integrated into the zerozero search system. By analyzing query logs from periods before and after the Solr integration, we found no significant changes in the search results. However, there was room for improvement in the search results for specific queries. To address these issues, we proposed a solution involving changes to the ranking algorithm.

Next, we explored the learning to rank module of Solr and its integration into the zerozero search system to improve the ranking process. In collaboration with the zerozero team, we defined features that could be used to assess the relevance of results, focusing on three main concepts: popularity, sporting merit of entities, and geographical location. For popularity, we used click-through data extracted from the search logs. For sporting merit, we accounted for the achievements and status of sports entities, while for geographical relevance, we considered the match between the origin of the query and specific entity attributes. Some of these features required a transformation step before being included in the ranking process. For the training process, we adopted a pairwise approach and used the number of clicks as the relevance factor. As training data, we again used the query logs provided by the zerozero team.

In the initial phase, we developed a system for each concept and used A/B testing to deploy these systems simultaneously in a common environment. The results demonstrated the potential of this approach, although they fell short of the original system. In the second iteration, we addressed the weaknesses of the systems and introduced an additional one encompassing features from all three concepts. The adjustments were effective, with all systems performing better than the original on the metrics we considered, namely ACP, MRR, wMRR, and Success. Finally, we conducted statistical tests to determine the significance of the results obtained. The tests demonstrated that the systems were able to return relevant results in the top positions and that the systems presented a statistically significant difference in the ordering of results compared to the original system.

Keywords: Information Retrieval, Domain-specific search, Learning to Rank

Resumo

Recuperação de Informação (RI) visa encontrar informação relevante em grandes repositórios digitais. Para satisfazer as suas necessidades de informação, os utilizadores recorrem a sistemas de RI. Este trabalho incide sobre o sistema de pesquisa da plataforma desportiva portuguesa zerozero, que apresenta uma oportunidade para melhorar a qualidade dos resultados de pesquisa.

Começámos por analisar os sistemas de pesquisa de algumas das plataformas desportivas mais utilizadas no mundo, incidindo nas suas interfaces de pesquisa e na forma como lidam com a popularidade nos resultados da pesquisa. As nossas observações revelaram limitações na funcionalidade de filtragem destes sistemas. Em contrapartida, a plataforma zerozero destaca-se pela cobertura completa de entidades, funcionalidades de controlo e resultados com informações detalhadas. Relativamente ao tratamento da popularidade, definimos um conjunto de interrogações que nos permitiu observar se a ordem dos resultados se alterava devido ao impacto dos eventos desportivos e observámos que uma plataforma apresentava alterações justificadas.

Em seguida, analisámos o impacto do Apache Solr, uma *framework* recentemente integrada no sistema de pesquisa da plataforma zerozero, utilizando, para isso, registos de interações dos utilizadores de períodos antes e depois da integração do Solr. Apesar de não verificarmos alterações significativas nos resultados da pesquisa, havia espaço para melhorar os resultados para certas consultas. Assim, propusemos uma solução que envolvia alterações ao algoritmo de ranking.

Posteriormente, explorámos o módulo de *learning to rank* do Solr e a sua integração no sistema de pesquisa da rede zerozero por forma a melhorar o processo de ranking. Em colaboração com a equipa do zerozero, definimos os sinais que poderiam ser utilizados para avaliar a relevância dos resultados, centrando-nos em três conceitos principais: popularidade, mérito desportivo das entidades e localização geográfica. Para a popularidade, utilizámos dados de cliques extraídos dos registos de interações. Para o mérito desportivo, tivemos em conta as proezas e o estatuto das entidades desportivas, enquanto que para a relevância geográfica, considerámos a correspondência entre a origem da consulta e certos atributos da entidade. Algumas destas características exigiram um processo de transformação antes de serem incluídas no processo de ranking. Para o processo de treino, adotámos uma abordagem par a par e utilizámos o número de cliques como fator de relevância. Como dados de treino, utilizámos novamente os registos de interações da plataforma.

Inicialmente, desenvolvemos um sistema para cada conceito e utilizámos testes A/B para os implementar simultaneamente num ambiente comum. Os resultados demonstraram o potencial desta abordagem, embora tenham ficado aquém do sistema original. Na segunda iteração, colmatámos os pontos fracos dos sistemas e introduzimos um sistema adicional com sinais dos três conceitos. Os ajustes foram eficazes, com todos os sistemas a alcançar um desempenho melhor do que o original nas métricas consideradas, nomeadamente ACP, MRR, wMRR e Success. Por fim, efetuámos testes estatísticos para determinar a significância dos resultados obtidos. Estes demonstraram que os sistemas produziram resultados relevantes nas posições de topo e que apresentaram uma diferença estatisticamente significativa na ordenação dos resultados face ao sistema original.

Palavras-chave: Recuperação de Informação, Pesquisa em domínio específico, Learning to Rank

Acknowledgements

I would like to thank my supervisor, Sérgio Nunes, who has always accompanied me and guided me in the right path and was always ready to clarify any doubts, allowing the work to flow at the desired pace.

I would also like to thank the zerozero team for the conditions provided and for their support and monitoring of this work, as well as for all the meetings that shaped the course of the project. A special thanks to Nuno Cardoso for his willingness to follow up and promote the practical aspect of this work.

Finally, I would also like to thank my family for all their support at this early stage of my life and Jéssica who has always pushed me and encouraged me to be true to myself.

Guilherme Garrido

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Goals	2
1.4	Contributions	2
1.5	Document structure	3
2	Search in Sports Environment	4
2.1	Overview	4
2.2	Search User Interfaces in Sports Platforms	5
2.2.1	Input Features	6
2.2.2	Control Features	7
2.2.3	Informational Features	8
2.3	Popularity Handling in Sports Platforms	9
2.4	zerozero	12
2.5	Summary	15
3	Evaluation in Information Retrieval	17
3.1	Overview	17
3.2	Relevance of a Document	17
3.2.1	Click-through Data	18
3.2.2	Reading Time	18
3.2.3	Query Reformulation	19
3.2.4	User History	20
3.2.5	Geographical Information	20
3.3	Evaluation Metrics	21
3.4	Comparison of Search Systems	22
3.4.1	A/B Test	22
3.4.2	Interleaving	23
3.5	Summary	24
4	Improving Relevance in Sports Search	26
4.1	Overview	26
4.2	Current zerozero Search System Architecture	26
4.3	Solr Framework Impact on zerozero	28
4.3.1	Dataset	28
4.3.2	Data Preparation	29
4.3.3	Search Characterization	30

4.3.4	Evaluation Process	33
4.3.5	Results	33
4.4	Problem	34
4.5	Proposed Solution	36
5	Learning to Rank in Sports Search	38
5.1	Learning to Rank	38
5.1.1	Learning to Rank in Solr	39
5.1.2	Learning to Rank Approaches	40
5.2	Applying Learning to Rank in zerozero	41
5.2.1	Test Setup	41
5.2.2	Feature Definition	41
5.2.3	Feature Engineering	44
5.2.4	Model Training	45
5.3	Solution Deployment	48
5.4	Summary	49
6	Solution Evaluation	51
6.1	First Version Systems Evaluation	52
6.2	System Improvements	54
6.2.1	Model Analysis	56
6.2.2	Evaluation	56
6.3	Summary	61
7	Conclusions	63
7.1	Main Contributions	64
7.2	Limitations	65
7.3	Future Work	65
	References	66
A	Search User Interface Features Comparison	70
B	Learning to Rank Models Used in the First Iteration	73
C	Learning to Rank Models Used in the Second Iteration	76

List of Figures

2.1	Result suggestions from sports.yahoo.com.	6
2.2	Search Engine Results Page on espn.com.	7
2.3	Number of retrieved results in espncricinfo.com.	9
2.4	Result information in flashscore.com.	11
2.5	Search bar results suggestions in zerozero.pt.	13
2.6	More results button in the search bar of zerozero.pt.	13
2.7	Search Engine Results Page on zerozero.pt.	14
2.8	Player result in zerozero.pt.	14
2.9	Team result in zerozero.pt.	14
3.1	Click patterns for queries and reformulations [8].	19
3.2	Re-ranking solution based on user history [17].	20
3.3	High level structure of an A/B test [16].	23
3.4	Example of interleaving for the query [napa valley] [27].	24
3.5	Detailed execution of Balanced and Team-Draft interleaving algorithms [28].	25
4.1	System architecture for retrieval and indexing processes and main execution flow steps.	27
4.2	Daily search distribution (pre-Solr Query Set).	31
4.3	Daily search distribution (Solr Query Set).	31
4.4	Search Engine Results Page for the query [aveiro] in zerozero.pt.	35
4.5	Proposed solution conceptual view.	37
5.1	Search pipeline of a system that uses Learning to Rank [26].	39
5.2	Training pipeline of a system that uses Learning to Rank [26].	40
5.3	Distribution of daily accesses to Artur Jorge’s player page on the zerozero network.	46
6.1	Evaluation metrics from first iteration systems.	54
6.2	Evaluation metrics from second iteration systems.	59

List of Tables

2.1	List of selected sports websites that have search functionality.	5
2.2	Overview of Search User Interface features and behavior in the selected websites.	10
2.3	Result information fields in the selected websites.	12
4.1	Description of main fields from a zerozero query log.	29
4.2	Click ranking distribution comparison.	30
4.3	Click per domain distribution comparison.	31
4.4	Top 20 most frequent queries (pre-Solr Query Set).	32
4.5	Top 20 most frequent queries (Solr Query Set).	32
4.6	Evaluation metrics results.	34
5.1	Description of the features used to assess relevance.	42
5.2	Example of geographic location features for <i>player-I</i> 's document and query text [ronaldo].	47
6.1	Evaluation metrics from first iteration systems.	52
6.2	Click position distribution from first iteration systems.	53
6.3	Correct answer position distribution from first iteration systems.	53
6.4	Clicks per entity distribution in the first iteration systems.	55
6.5	Evaluation metrics from second iteration systems.	57
6.6	Cumulative click position distribution from second iteration systems.	57
6.7	Cumulative correct answer position distribution from second iteration systems.	58
6.8	Average click position per entity in the second iteration systems.	58
6.9	Mann-Whitney U test <i>p-values</i>	60
A.1	Search User Interface Input features in sports-related websites.	70
A.2	Search User Interface Control features in sports-related websites.	71
A.3	Search User Interface Informational features in sports-related websites.	72

Listings

4.1	Example of query log entry.	29
5.1	Example record from the processing of clicks.	44
5.2	Example records from the training entries used as input for model training.	46
5.3	Example records from the training entries for <i>player-1</i>	47
5.4	Example relevance-features map pair for <i>player-1</i>	47
B.1	Model Pop features and weights.	73
B.2	Model Mer features and weights.	74
B.3	Model Geo features and weights.	74
C.1	Model Pop version 2 features and weights.	76
C.2	Model Mer version 2 features and weights.	76
C.3	Model Geo version 2 features and weights.	77
C.4	Model All features and weights.	78

Abbreviations

IR	Information Retrieval
SERP	Search Engine Results Page
UCL	UEFA Champions League
UEL	UEFA Europa League
AP	Average Precision
MAP	Mean Average Precision
MRR	Mean Reciprocal Rank
RR	Reciprocal Rank
wMRR	weighted Mean Reciprocal Rank
ACP	Average Click Position
ML	Machine Learning
LTR	Learning to Rank
PSQS	Pre-Solr Query Set
SQS	Solr Query Set

Chapter 1

Introduction

Information Retrieval (IR) dates back to the 1940s, marking the beginning of a field that has been growing in importance alongside the rise of digital information. Manning et al. [21] define IR as the task of "*finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).*" An information need is the topic about which the user wants to know more [21]. It can be seen as a gap that leads one to initiate a search for specific information. It is important to note that information need differs from query, which is what the user types in an attempt to satisfy their information need. Document relevance is another crucial related concept. A document is considered relevant if the user perceives it contains value for their information need. In order to fulfill their information need, the user utilizes a search engine, which is an IR tool designed to take a query and return a list of documents deemed relevant [21]. Ranking algorithms are a critical component of search engines, as they determine the order in which the documents are presented to the user based on their predicted relevance to the query.

1.1 Context

The present work aims to improve the search system of zerozero¹, a Portuguese sports platform initially centered around football since its inception in 2003. Although the platform has recently expanded to include six additional sports modalities, such as handball and basketball, and has extended its reach to ten other countries, such as Brazil and England, most of its activity remains rooted in the football section of the Portuguese domain. Managing an extensive database of almost two million records presented significant performance challenges, which were addressed through a recent technological shift with the incorporation of the Apache Solr framework. This transition improved the overall search experience, but there is still a margin for improving the relevance of the retrieved results.

¹<https://www.zerozero.pt>

The company acknowledges that the results for some queries are not yet satisfactory and is looking at alternatives to improve the ranking algorithm, mainly through the integration of different signals, such as the popularity and success of sports entities and geographical relevance. Combining these elements in the right proportions is also a key factor in improving the quality of results.

1.2 Motivation

The original system still faces difficulties retrieving relevant documents in the highest positions for some queries. We expect that by addressing this issue, the system will be able to provide results that better align with user expectations, thereby improving the overall user experience on the platform. This work explored IR techniques to help improve the relevance of results from an already established system comprising almost two million documents. The change in technology with the integration of the Solr framework has also presented opportunities for the evolution of the search system that have yet to be explored in the context of zerozero, such as the integration of learning to rank techniques, which have shown notable success [18].

1.3 Goals

The main objective of our work was to integrate several signals into the ranking algorithm that could help assess the relevance of the results to improve the retrieval quality. While some of these signals were attributes of the documents, others had to be computed, such as the data associated with the clicks of the platform users, which was extracted from the query logs provided by the zerozero team. Instead of manually incorporating these signals into the ranking algorithm, as was done in the original zerozero system, we decided to explore techniques for automatically constructing the ordering function and evaluate its effectiveness in handling the variety of signals to produce relevant results at the top of the ranking.

1.4 Contributions

The work developed significantly improves the search system of the zerozero network by applying IR techniques to improve the quality of results. In addition, we have implemented an A/B testing architecture that allows for continuous optimization by directly comparing different system versions. These improvements demonstrate immediate benefits to the zerozero platform while also providing a foundation for future enhancements.

Furthermore, this study contributes to the field of IR systems within sports platforms. We conducted a comprehensive analysis of search interfaces across different sports platforms, highlighting the strengths of the zerozero interface. Additionally, we explored techniques for obtaining implicit feedback on relevance and applied them to the ranking function by exploiting query logs.

1.5 Document structure

Beyond this introduction, the document is structured as follows: in Chapter 2, we analyze and compare the search systems of the most visited sports-related websites worldwide and zerozero regarding search interface and popularity handling. Chapter 3 presents several concepts related to the evaluation in IR, namely some methods for assessing the relevance of a document, evaluation metrics for information retrieval systems, and techniques for comparing IR systems. Chapter 4 starts with a description of the zerozero search system, in particular, the architecture of the system and the impact of introducing the Solr framework. It then proceeds to describe the problem faced and the proposed solution. In Chapter 5, we introduce the concept of learning to rank and describe the implementation of the learning to rank module of Solr in the zerozero system. We go through the processes of feature definition, feature engineering, model training, and deployment through A/B testing architecture. Chapter 6 describes the evaluation of the solution and the changes made to improve the systems. Finally, in Chapter 7, we present the conclusions of this work, highlighting the main contributions, limitations, and possible future extensions.

Chapter 2

Search in Sports Environment

2.1 Overview

Information needs in sports can vary widely from a football fan seeking statistics on a particular player's career or being updated with the most recent news about their favorite team to a manager or a scout getting to know more about promising players from training-level teams from another country. Zhai et al. [40] state that sports information resources are becoming more abundant and that IR has increasingly become indispensable for sports information management. The most common entities in the sports domain are teams and players, but others exist, like managers, competitions, stadiums, referees, matches, board directors, and agents. The relevance of sports results can be influenced by several signals, including the popularity¹ and the merit (achievements) of sports entities. In order to fulfill their information needs, users rely on a IR system, which requires them to interact with its Search User Interface (SUI) to submit their query. There are numerous established principles for the design of SUI [37].

In order to analyze search systems in sports environments, we explored some of the most widely used sports-related platforms globally. To identify them, we gathered information about "the most visited sports websites in the world" from [semrush.com](https://www.semrush.com/website/top/global/sports/)² (most visited websites in September 2023), [similarweb.com](https://www.similarweb.com/top-websites/sports/)³ (also from September 2023), and [statista.com](https://www.statista.com/statistics/1417997/most-visited-sports-websites-world-wide/)⁴ (August 2023), and on "the best sports websites" from [feedly.com](https://feedly.com/i/top/football-blogs)⁵ webpage.

To choose the ones to be analyzed, we gave weights based on the ranking position: 2 points for a website in the top 10, 1 point for top 10-20 and 0.5 points for mentions on an article from [bleacherreport.com](https://bleacherreport.com/articles/1004578-top-football-websites-for-fans-and-writers)⁶ (2012), a sports specialized platform, providing historical relevance. We chose 14 websites with a score of 2.5 or more and 4 with a score of 2 that are mentioned by two

¹Popular items are likely to be known by users and unlikely to be perceived as novel [22].

²<https://www.semrush.com/website/top/global/sports/>

³<https://www.similarweb.com/top-websites/sports/>

⁴<https://www.statista.com/statistics/1417997/most-visited-sports-websites-world-wide/>

⁵<https://feedly.com/i/top/football-blogs>

⁶<https://bleacherreport.com/articles/1004578-top-football-websites-for-fans-and-writers>

Table 2.1: List of selected sports websites that have search functionality.

Website	Searchable entities	Sports covered
espn.com	All	Multiple sports
goal.com	Matches	Football
as.com	News	Multiple sports
cricbuzz.com	All	Cricket
espnricinfo.com	All	Cricket
mundodeportivo.com	News	Multiple sports
si.com	News	Multiple sports
mlb.com	All	American football
flashscore.com	All	Multiple sports
caughtoffside.com	All	Football
livescore.com	Competitions	Multiple sports
nfl.com	All	Baseball
lequipe.fr	All	Multiple sports
sportskeeda.com	All	Multiple sports
sports.yahoo.com	All	Multiple sports

different sources. From these 18 sports-related websites, we removed three that did not have search functionality, leading to the list shown in Table 2.1. We decided to analyze the ones where all entities are searchable, plus livescore.com as competitions are entities we want to consider, leading to 11 relevant websites. From these, one exclusively addresses football (caughtoffside.com), two cricket (cricbuzz.com and espnricinfo.com), one is related only to American football (nfl.com), and one to baseball (mlb.com). The rest gathers information from multiple sports.

Being the most visited sports-related websites in the world does not necessarily mean they have the best IR systems in the field of sports. In fact, some of them do not even have a search functionality, which shows that these websites can be widely used due to other features that attract users, such as the publication of news, personal opinions, or live match scores.

In the following sections, we will analyze the selected websites to ascertain the SUI of sports websites and how they handle popularity in the ranking of results.

2.2 Search User Interfaces in Sports Platforms

In this section, we will explore the search user interface (SUI) of 11 of the most used sports-related websites in the world. The findings will be presented based on the framework proposed by Wilson [37], which categorizes SUI elements into four main groups: Input, Control, Informational, and Personalisable features. In our work, we will consider only the first three. We then provide a

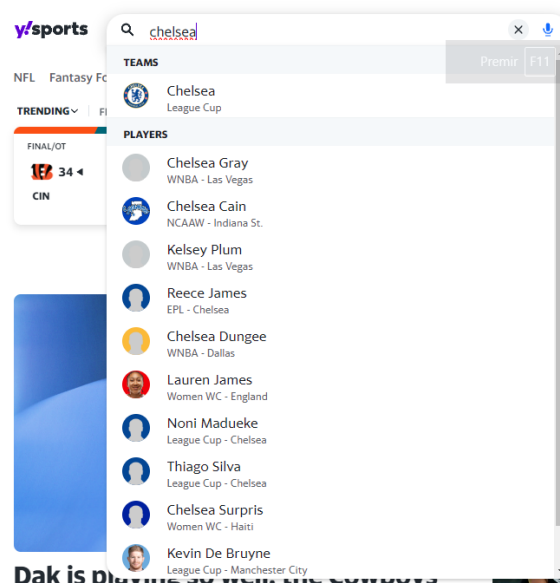


Figure 2.1: Result suggestions from sports.yahoo.com.

global comparison in tables 2.2 and 2.3, while in Appendix A, we gathered the details across the selected websites.

2.2.1 Input Features

The input mechanisms of an IR system are intrinsically associated with the search bar, which is a central feature. The way it behaves helps to shape the search, for example, by auto-completing the query [37]. All the analyzed websites provide a search box even though only yahoo.sports.com makes it fully visible. Eight websites provide the magnifying glass icon that, when clicked, enables the search box. The remaining two websites hide this feature as an option in the menu (nfl.com and lequipe.fr).

Suggesting results or text while the users are typing their queries is a way to speed up the search task if relevant results are presented (Figure 2.1). When using the search bar, five analyzed websites (45,5%) show results suggestions while the user is typing. These websites present a limited number of suggestions, with a maximum of 20 suggested results shown. Some websites organize the results by grouping them by entities, which helps the user search more efficiently. Two other websites show text suggestions on the search bar to help the user complete the query, although they do not show any results at this stage. On the other four websites, nothing happens when the user is typing, even though on espn.com, the results update on the search page when the query is typed.

Next, we analyzed the behavior of these systems when the user submits the query, i.e., when the ENTER key is pressed or when clicking on the search button to submit the search. Most of the websites (72,7%) redirect the user to a Search Engine Results Page (SERP) with relevant results (Figure 2.2). On the other hand, cricbuzz.com redirects the user to the page of the first result shown in the suggestions list. A few of them (flashscore.com and livescore.com) do not allow the

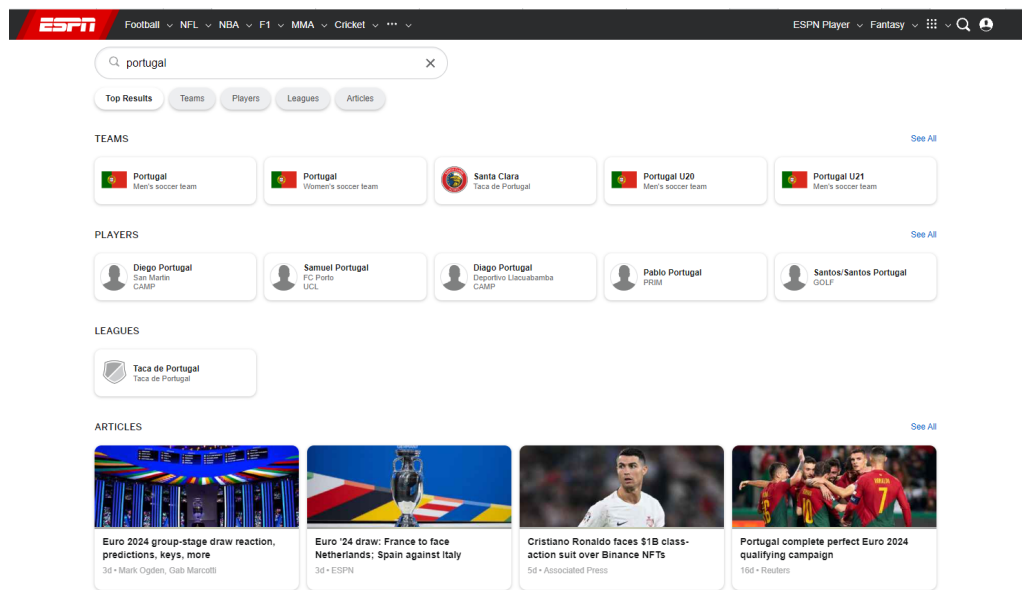


Figure 2.2: Search Engine Results Page on espn.com.

user to submit the query and limit their search functionality to showing a list of clickable results that is immediately presented on query typing.

The last input feature we analyzed was the visual presence of the query text. It can also be considered an Informational feature, as it provides information about the search. Eight of the websites keep the query text in the search bar. The two platforms that do not allow query submission (flashscore.com and livescore.com) keep the query visible. Finally, when the query is submitted, sportskeeda.com hides the search bar and does not provide the query text anywhere else. The details about Input features in the analyzed websites are gathered in Table A.1.

2.2.2 Control Features

Control features are mechanisms that help users shape the search results. Research has shown that a good support provided by Control features can significantly improve the search task [37]. We analyzed websites regarding search filters, ordering (or sorting) of results, and the behavior when terms are misspelled.

Search filters are tools that allow users to restrict search results in some way [37]. Ten websites provide filters to help focus relevant results, but they only allow filtering by entity (the most common are teams, players, news, and competitions) or by sport. This is visible in figures 2.2 and 2.4 respectively. Within an entity or a sport, there are no ways to restrict results or customize the search. The exception is nfl.com, which allows users to search for players and filter them by activity (active or retired).

In terms of ordering the retrieved results, this feature is not common between the analyzed websites. Only three of the eleven (27,3%) allow the user to order the results: they provide the

option to order by date or relevance. This feature is helpful in ordering news as the websites that have it retrieve news as a result of a user query.

To understand how these websites handle spelling errors, we submitted the query [dalot] on the ones that contain football data. We observed that the results were relative to the Portuguese football player Diogo Dalot. We then submitted the query with a spelling error [dalott] and observed that no results were found and no corrections or suggestions were available. On the websites that do not have football data, we did the same task but with the query [indya], a typo for [india], and we observed the same: no results found and no corrections/suggestions provided. Table A.2 gathers the details about the Control features.

2.2.3 Informational Features

Informational features provide information about the results of the SERP, namely in terms of organization and display [37]. We analyzed whether or not pagination and information about the volume of retrieved results are present, as well as the information provided for each result.

Pagination is a common feature when there is a large set of results to be presented. There is a division regarding this functionality from the websites that provide the complete list of the retrieved results (and not just the suggestions, like flashscore.com and livescore.com do). 63,6% (7) provide pagination. Three of them show the five most relevant results for each entity, but within each entity, they show all the results on a single page. Another three websites provide pagination, showing ten results per page, while sports.yahoo.com shows only seven. The remaining two websites do not provide pagination, using infinite scrolling to show the complete list of retrieved results on a single page.

In terms of providing the number of retrieved results, 63,6% of the analyzed websites (7) show this information. espncriinfo.com provides this information for each entity (Figure 2.3), giving a better understanding of the composition of the retrieved results. sportskeeda.com also shows the number of retrieved results for each entity, but only when the user switches to its specific tab. Table A.3 presents the details about these Informational features in the analyzed sports platforms.

The way the results are presented helps users identify whether their information needs are present in the retrieved results. These results usually appear in the SERP or as suggestions when the query is typed, and they can contain text or non-text data, such as images. For instance, for a football player, the result can include information about the player's name, photo, and the name of the team they play for (Figure 2.4). The objective is to allow the user to decide its relevance according to the information presented. In sports, result information usually differs from entity to entity as each has different fields with different relevance. For example, while a club result can contain a label for its country, a national team already has this information as the team's name. Likewise, a player's result may contain the name of the team they represent, but this information does not make sense when presented in a team result.

From our analysis of the selected websites, we found different designs and different kinds of information presented in the results. All of them show the result's name (for players and teams) or the news article's title. The majority (81,8%) also provide an image along with the text content

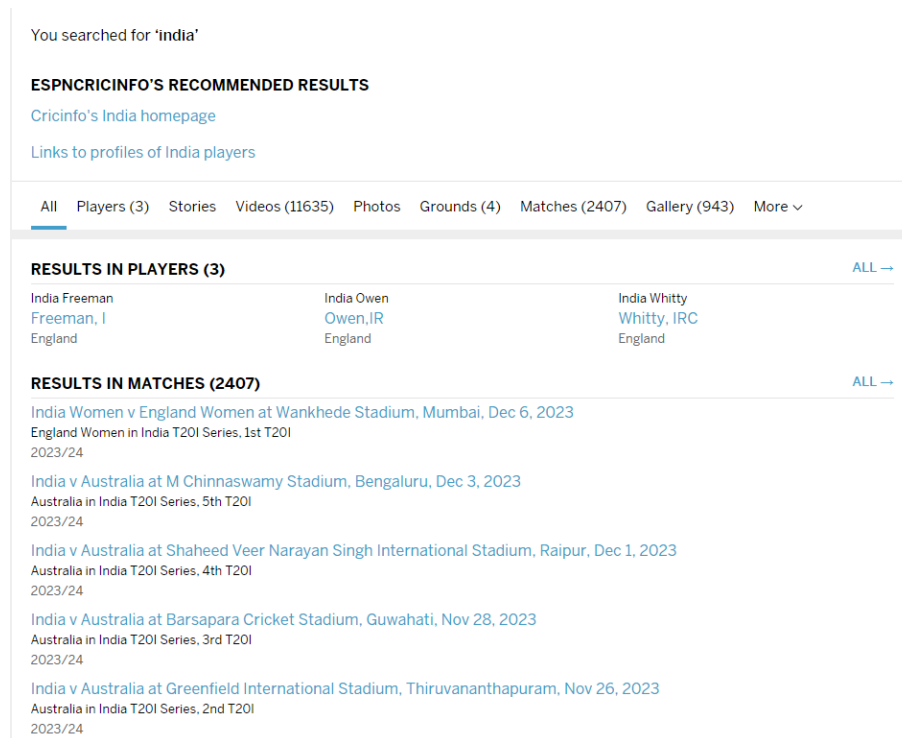


Figure 2.3: Number of retrieved results in espncricinfo.com.

to help users visually perceive each result. The other most common fields for players are the team in which they play (in 6 of the websites), the nationality (in 3), and the position (in 2). For team results, the attributes shown are the league and the country in which the team plays. Competitions are entities recognized only on some websites, but the results do not contain more than their logo and name. When managers are identified entities, the results contain identical information to the players, as shown in Figure 2.4. These websites do not recognize the rest of the entities mentioned before. In Table 2.3, we gathered the number of websites that show specific fields in the information of a result.

2.3 Popularity Handling in Sports Platforms

We observed the ranking of results for specific queries on the chosen websites in order to analyze how sports platforms deal with the popularity of entities in their IR system. This analysis took place from the 5th of February until the 6th of March 2024, a period in which major football matches that should attract more fans took place, including 12 UEFA Champions League (UCL) round of 16 games and 16 UEFA Europa League (UEL) playoff games. We, therefore, based our analysis on football. For this analysis, we excluded not only the websites related to cricket, American football, and baseball but also livescore.com, which only allows users to search for competitions, and therefore, the order of results would hardly change, leading to a total of 6 websites.

Table 2.2: Overview of Search User Interface features and behavior in the selected websites.

Features/Behavior		Number of websites
Search bar/icon	Visible*	9 (81,8%)
	Hidden	2 (18,2%)
On query typing	Results suggestions*	5 (45,5%)
	Text suggestions	2 (18,2%)
	Nothing	4 (36,4%)
On query submission	Page with results*	8 (72,7%)
	First result page	1 (9,1%)
	Not allowed	2 (18,2%)
Query text	Visible*	10 (90,9%)
	Not visible	1 (9,1%)
Filters	By entity	5 (45,5%)
	Other	5 (45,5%)
	No filters	1 (9,1%)
	First by entity, then specific filters*	0 (0,0%)
	By date and relevance	3 (27,3%)
Ordering	Not allowed	8 (72,7%)
	By relevance and alphabetical order*	0 (0,0%)
	Yes*	7 (63,6%)
Pagination	No	2 (18,2%)
	Infinite scroll	2 (18,2%)
Volume of results	Yes*	7 (63,6%)
	No	4 (36,4%)

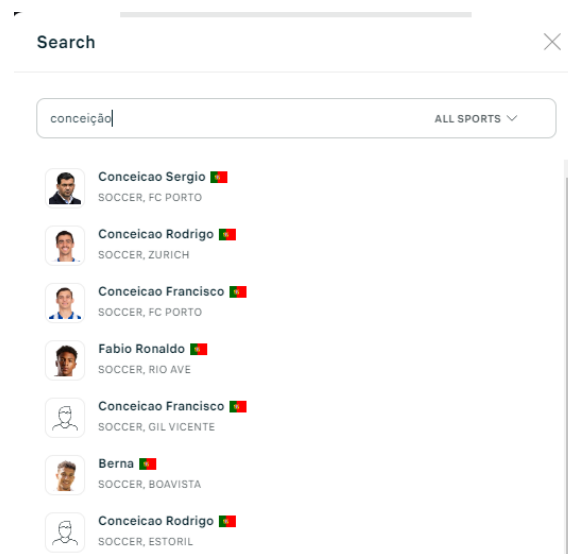


Figure 2.4: Result information in flashscore.com.

In order to see if there were any changes in the order of the results, we defined a set of queries that could facilitate such an analysis. We, therefore, used the queries [port], [real], [spain] and [manc]. With the first one, results such as the Portuguese teams FC Porto and Sporting CP, the Portuguese national team, and the NBA team Portland Trail Blazers can be retrieved. Through the [real] query, we can reach several Spanish clubs such as Real Madrid, Real Betis, Real Sociedad, and Real Valladolid. By querying [spain], we wanted to understand the ranking of Spanish clubs, and, finally, with the query [manc], we wanted to understand which of the Manchester clubs (United or City) was presented at the top of the ranking. Out of the six platforms we analyzed, 4 showed no difference in the ordering of results except for slight variations, such as on the yahoo.sports.com website where, in the 'spain' query, Alavés was the fifth result on February 5th and on March 6th it was no longer in the top 7. The rest remained the same, including the Real Madrid team result, which was always behind FC Barcelona, even during the weeks when they were the only team playing in the UCL. This indicates that there is a fixed order and that recent popularity is not considered in the ranking process.

The two websites that showed significant variations in the order of the results were espn.com and flashscore.com. In the first, it varied considerably, but we could only justify it with the popularity factor in the [port] query: in espn.com, we saw FC Porto at the fifth and fourth positions at the start of February and March, respectively. On the 22nd of February, the day after FC Porto's win in the UCL, the team result was ranked second for that query. For the remaining queries, there were some changes, but we were unable to justify them with the occurrence of sporting events. Therefore, it is not feasible to draw conclusions about the impact of popularity on the ordering of results on the espn.com platform. In flashscore.com, the query [port] had different variations. During the analysis period, the first two results were always FC Porto and Sporting CP, with FC Porto coming first on UCL match days and Sporting CP on UEL match days. During the rest of the days, there were some changes that were not explained. Regarding the [real] query, on the

Table 2.3: Result information fields in the selected websites.

Fields	Number of websites containing the field
Name*	11 (100%)
Image*	9 (81,8%)
Club*	6 (54,5%)
Nationality*	3 (27,3%)
League	2 (18,2%)
Sport*	2 (18,2%)
Position*	2 (18,2%)
Status*	1 (9,1%)

first day of analysis, the results were, in this order: Real Madrid, Real Betis and Real Sociedad. Real Betis had the game closest to that date. After Real Sociedad's UCL match on the 14th of February, this team's result was shown in the second position. On the 21st of the same month, the day before Real Betis' match in the UEFA Europe Conference League, the order was reversed. The same thing happened again on the day before Real Sociedad's match in the second leg of the UCL round of 16 (February 4th), putting this team's result back in second place. A similar thing happened with the query [manc]. The first two results correspond to the teams Manchester United and Manchester City, which on the 5th of February were in that sequence on the flashscore.com website. After Manchester City's UCL match on February 15, the order was reversed, and on the 6th of March, the date of the team's next match in the same competition, Manchester City was the top result. For the query [spain], no changes were observed during this period, which was to be expected as the system of this platform only retrieves results for Spanish national teams and not for Spanish clubs. We also analyzed the query [sak], which returned tennis player Maria Sakkari as the first result and Arsenal footballer Bukayo Saka as the second. This order stayed the same during the period in question, even on days close to Arsenal football matches on the UCL. Therefore, we can conclude that the changes were not global.

2.4 zerozero

zerozero is a Portuguese website created in 2003 that started by being concerned more about football. Later, the network expanded into international domains and other sports using its database with almost 2 million records, which is known as the world's largest football database.

We focused our analysis on the Portuguese domain, zerozero.pt, but the rest of the network has a similar interface. The search bar is visible at the top of the page. Introducing a query in the search bar shows the top-10 results as suggestions (Figure 2.5), where each result links to the respective specific profile page. It also shows an option to see "more results" (Figure 2.6) that

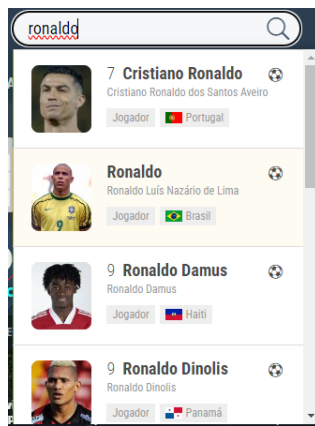


Figure 2.5: Search bar results suggestions in zerozero.pt.

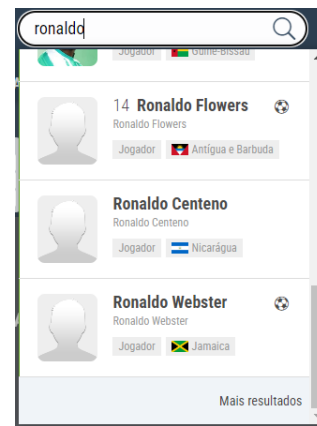


Figure 2.6: More results button in the search bar of zerozero.pt.

redirects the user to the SERP, which gathers results from all entities. The same thing happens when the user submits the query. On that page (Figure 2.7), the query text is kept, the results are paginated (20 results on each page), the number of retrieved results is shown, as well as some options for ordering (most popular and alphabetical order) and filters to help the user customize its search. When the entity is not specified, there is a filter by entity. When the entity is defined, multiple other filters that vary from entity to entity are shown. For a player, it is possible to narrow down the results by sport, nationality, in-game position, state of career (active or retired), whether they are international players, gender, age interval, current club, past clubs, and the country they play in.

Table 2.2 shows an overview of the SUI features in the selected websites. The asterisk (*) highlights the feature used by the zerozero system. In conclusion, we can see that zerozero follows the majority of platforms in terms of Input and Informational features, while for Control features, it stands out for its more complete filtering and ordering of results than any other website analyzed.

The entities defined by zerozero are players, teams, competitions, managers, stadiums, referees, directors, agents, and editions⁷. The results for entities that are people (players, managers, referees, directors, and agents) (Figure 2.8) highlight the person's photo, abbreviation (the name by which the person is known in the world of sports), and a color representing their current activity or status (green for active, grey for retired or deceased and yellow for legend). A result also contains the number of the shirt (only for players), full name, nationality(ies), date of birth, current team, number of international caps and trophies, an icon representing the sport, age, and position (only for players). A star icon is also available if that person is considered a legend of the respective sport besides still being active. For clubs and international teams results (Figure 2.9), the photo and the abbreviation are also the most highlighted information. The whole team name, country, city, date of foundation, number of teams (other sports and training levels teams for that club), trophies, and an icon representing the sport are also presented. Stadiums show name, photo,

⁷Edition is an instance of a competition, typically identified by the year or a sequential numbering system (e.g., *Premier League 2023/2024*).

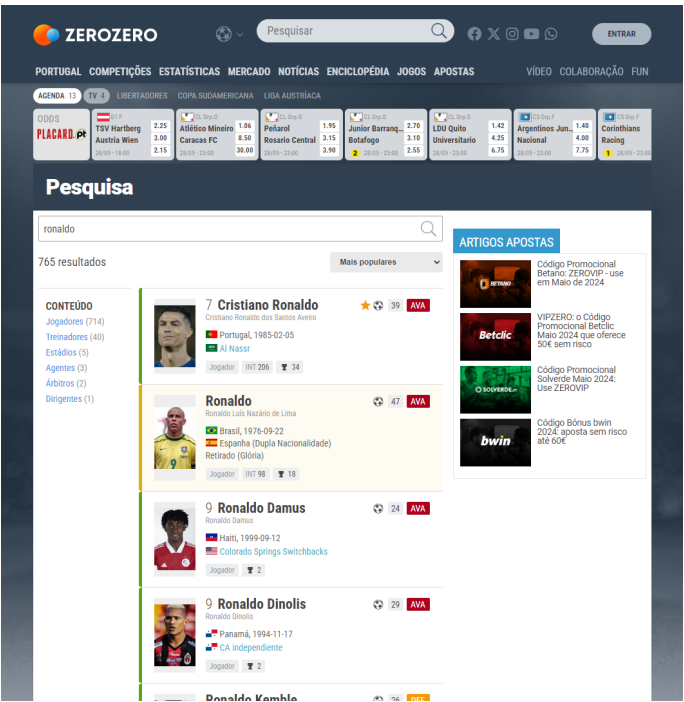


Figure 2.7: Search Engine Results Page on zerozero.pt.

country and city, teams that play in them, and the maximum capacity, while competition results provide information about the name, country, number of editions, games and teams, associated sport, and competition level. Table 2.3 shows the fields that are presented in the search results of the other sports platforms. In this case, the asterisk marks whether they are present in zerozero's results, and we can see that only the league field misses out.

In the past, when the user submitted a query in the search bar, the platform redirected the user to the profile page of the top retrieved result by the system but also provided the option to search the complete list of results for that query. The search bar also showed information about the volume of results with the number of results per entity type at the bottom. By clicking on this information, the user was directly taken to a page with the retrieved results for that query within the specific entity. In the current system configuration, the user has to submit the query and select the entity on the filter, while a click after a scroll down did the job in the past. Tables A.1, A.2 and A.3 gather the functionalities present in each analyzed search engine, including zerozero's old and new systems.

Another thing worth mentioning is the fact that only four results are shown at a time in the

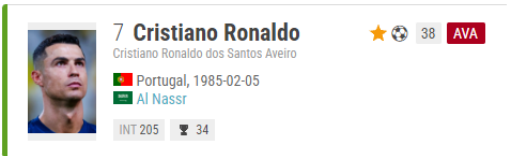


Figure 2.8: Player result in zerozero.pt.



Figure 2.9: Team result in zerozero.pt.

search bar when the query is being typed, meaning that only 4 of the ten results presented are shown without scrolling, as is visible in Figure 2.5. This complicates the visualization of results outside the top 4 as it requires user effort to see it.

Regarding handling entities' popularity, the ranking system of zerozero assigns some importance to user interactions by including fields related to feedback when searching for entities. To achieve this, documents are valued more highly the greater the number of visits to each entity's page using the internal search feature, i.e., each time an entity's page is accessed as a result of a search, the value of this field in that entity's document is increased. This ranking mechanism also includes a geographical factor since the value of the aforementioned field only corresponds to the access to an entity's page in a given domain. Thus, if an entity has many accesses in the Portuguese domain, it will only be valued in the sorting of results from the Portuguese domain and not from other domains.

2.5 Summary

We conducted a comprehensive analysis of the search engine of a set of the most used sports-related websites in the world. The focus was on the search user interface and the handling of entities' popularity. In terms of the interface, we analyzed some of the most relevant features. For the Input features, the trend was for the query text and the search bar (or icon) to be visible, for suggestions of results to be displayed while the query is being typed, and for the SERP to be displayed when the query is submitted. For the Control features, most of the selected websites provide some sort of filtering, but ordering results is not allowed. Finally, for the Informational features, the trend was to provide pagination and information on the number of results retrieved. One of the most surprising findings was that only one of the analyzed sports platforms, cricbuzz.com (and the previous zerozero system configuration), directly leads to the first result when the query is submitted. Additionally, the filtering functionality of the websites under analysis is notably limited, allowing, at most, the filtering of the results by entity. Regarding the ranking of results, only a few of the search engines showed changes for some queries that we carefully defined for the analysis period. Nevertheless, we observed slight but justified changes in the order of results for three queries on one of the websites.

Subsequently, we analyzed the search system of the Portuguese sports network zerozero, which is the focus of this work. We can conclude that zerozero is significantly more complete, particularly in terms of entity coverage, control features, and results information. This enables users to refine their search and find the desired information more simply. The results in zerozero contain not only 7 of the eight fields present on the other analyzed websites (as illustrated in Table 2.3) but also additional fields such as age, date of birth, number of international matches, and trophies. The data presented in Table 2.2 allows us to conclude that zerozero is either following the majority or more advanced than the other websites analyzed regarding the search interface features explored.

With the change in the query submission behavior, zerozero adopted a strategy that aligned with the majority of the websites we analyzed, contributing to an enhanced search experience

by establishing a more predictable and consistent user interface. On the other hand, removing clickable information about the volume of results represented a negative change, as it now requires more effort for the user to find the same set of results. Furthermore, it can be anticipated that results outside the top-4 are likely to be less visualized, which may influence the results of a click-through data analysis (which will be discussed in the following chapter). Finally, the mechanism zerozero uses to handle popularity consists of valuing the documents of the entities with the most searched pages.

Chapter 3

Evaluation in Information Retrieval

3.1 Overview

Evaluation in information retrieval (IR) is a critical phase for assessing system effectiveness. It allows for comparative analysis of different systems, identifies strengths and weaknesses, and provides feedback for iterative improvement. As stated by Manning et al. [21], delivering relevant results is the most important factor in user happiness. In their study, they outline three key components involved in a comprehensive evaluation:

- a document collection;
- a complete set of information needs in the form of queries;
- a set of relevance judgments, typically a binary assessment of either relevant or non-relevant for each query-document pair.

A relevant document is one that contains valuable information for the user information need [21]. Given a set of queries, it is necessary to ascertain the relevance of the documents to them. In the next section, we will explore different ways of assessing the relevance of documents.

The selection of metrics is also of critical importance in the evaluation process, as it significantly impacts the conclusions drawn about the effectiveness of a system. This will be addressed further below. Finally, we will cover the main techniques for simultaneously deploying system variants in IR. This approach has the advantage of enabling direct comparisons between systems within the same environment.

3.2 Relevance of a Document

There are many ways of assessing relevance, implicit or explicit, but, in the end, a binary assessment for each query-document pair is needed to evaluate a system [21]. Explicit relevance feedback refers to the direct indication by users about the perceived relevance of a document for

a given query. At the same time, implicit relevance is related to the use of indirect sources of evidence [21], i.e., it refers to indicators of document relevance that are inferred from user behavior on a system (for example, the amount of time the user spends in a particular document). Even though explicit relevance feedback is easy to use, it becomes difficult to generalize given the definition of relevant document, and users rarely contribute with this kind of feedback [10]. On the other hand, implicit feedback *"has the advantage that it can be collected at much lower cost, in much larger quantities, and without burden on the user of the retrieval system"*, even though it is more difficult to interpret and could be noisy [11]. Next, we will analyze some ways of obtaining implicit relevance feedback.

3.2.1 Click-through Data

Utilizing click data presents a potential method for enhancing result relevance with low-cost [32]. Click data can be used to acquire implicit relevance feedback from search results: if a user clicks on a result, we assume they found it relevant. Joachims et al. [11] suggest that click-through data on the Web can be *"reliably extracted from observed user behavior"*.

In their study, Joachims et al. [12] mention that interpreting clicks as a relevance assessment has some issues, namely trust bias and quality bias. The first one refers to the fact that users trust the search engine and click on results that are better ranked. Their study showed that users tend to click on the first result even if the second result's abstract is more relevant, concluding that the ranking influences users' clicking behavior. Click-through data is also biased by the quality of the ranking of a system. That is, as the relevance of the retrieved results decreases, users' clicks are also less relevant on average [12]. There are other limitations to the technique:

- Users might stop clicking on results once their information needs have been met, so they might ignore other relevant results that will be addressed as non-relevant.
- Snippet bias, i.e., the click on a result is often influenced by the snippet the user examined, which makes the click information judge the snippet rather than the document itself.
- Users might get distracted from their current task and click on a result that is not relevant to their information need simply because they find that result interesting.

3.2.2 Reading Time

The time spent on a document page can be related to its relevance because users spend more time reading interesting information than not-interesting [24]. In their study about implicit feedback on the reading of news articles, Morita and Shanoda [24] also considered three factors that may affect the time spent on a page: the size or length of an article, the readability and the number of articles left to read (if the user has a large number of articles to read, the time spent for each article will get shorter). In the end, Morita and Shanoda [24] concluded that there is a strong tendency to spend a long time reading articles of interest and not to do so in the ones that could be rated as

	Searcher Actions on Results			
Initial Query	Click		Skip	
Reformulation	Click	Skip	Click	Skip

Figure 3.1: Click patterns for queries and reformulations [8].

not interesting, meaning that there is a correlation between the time spent on a document and its relevance.

On the other hand, Kelly and Belkin [15] defend that the objective of the user search, referred to as the "task" leading to information-seeking behavior, significantly impacts the time spent on documents. For example, news articles are displayed longer than shopping items or job searching, and therefore, the "task" is a limitation to the relation between reading time and relevance.

3.2.3 Query Reformulation

Users frequently refine their queries to express their information needs better, often undergoing multiple reformulations before finding satisfactory results [12]. On the other hand, analyzing user click-through information can be challenging due to factors such as snippet bias and trust bias, and, therefore, examining query reformulations becomes a relevant strategy in understanding user satisfaction [6].

Examining the queries separately may lead to misinterpretations. For instance, users might click on a result during their initial query but then perform a subsequent query as they are unsatisfied with the accessed content. If we analyze these queries independently, we might infer that the user was satisfied with the first result when, in fact, they could have performed a query reformulation because they considered the result not relevant [6]. If a user modifies their query and finds the subsequent results more relevant, it suggests that the initial results were not entirely satisfactory. On the other hand, when there is no further search activity for at least 30 seconds, the user should be satisfied with the clicked result. In their study, Hassan et al. [6] concluded that search success is an incremental process in most multi-query tasks.

Studies cited by Joachims et al. [12] indicate that when a user refrains from clicking any result but opts to reformulate the query, there is strong evidence that the top results from the initial query were not relevant. Moreover, they verified that 69% of the top results of a later query were more relevant than the top result in an earlier query. Another scenario proposed by Joachims et al. [12] involves a sequence of queries without any clicks after the query "oed" followed by a reformulation to "oxford english dictionary" with a click. This might suggest that the clicked result is relevant to the original query "oed", even if the document does not explicitly contain this string. Huang and Efthimiadis [8] synthesized the four possible click patterns of a reformulation process through a simple model (Figure 3.1). A "SkipClick" pattern, where a click follows a skip, suggests that the user found the reformulation useful, while a "ClickSkip" indicates the opposite.

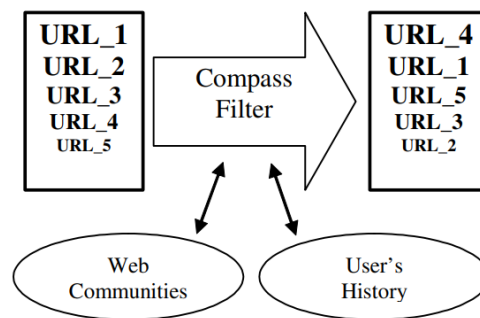


Figure 3.2: Re-ranking solution based on user history [17].

3.2.4 User History

User's past interactions with the Web is a type of implicit feedback that can be used to infer a goal or intent [35]. By capturing the user's preferences from previously issued queries or previously visited Web pages [35], there is an opportunity for the search to be tailored to the searcher, leading to an expected increase in the relevance of retrieved documents. For example, suppose a user's history shows that only FC Porto players' pages were visited. In that case, the user will likely find FC Porto's Portuguese right-back¹ more relevant than Benfica's midfielder² on a [João Mário] query.

In their studies, Teevan et al. [35] concluded that personalizing the search by using information from the searcher obtained "*statistically significant improvements over the default Web ranking*" that treats all users the same way. Kritikopoulos and Sideri [17] proposed a re-ranking solution (the "Compass Filter" illustrated in Figure 3.2) that takes the initial results and considers the user's history and groups of similar documents ("Web communities") to provide a personalized set of retrieved results. On the other hand, this raises privacy concerns and should be used carefully not to limit recently introduced documents that have no past interactions for being new.

3.2.5 Geographical Information

Documents often contain geographic data that can be used to assess their relevance given a query. Geographical relevance ranking aims to provide geographically relevant results to the users. It includes three steps [23]:

- the processing of documents to extract geographic information and associate them with places or coordinates;
- the understanding of the geographic intentions behind user queries by collecting and mapping data such as IP address and user's history;

¹<https://www.zerozero.pt/jogador/joao-mario/421283> (accessed December 2023)

²<https://www.zerozero.pt/jogador/joao-mario/56560> (accessed December 2023)

- the retrieval and ranking of documents according to user queries refined through a fusion of textual and geographic similarity metrics using ranking algorithms.

In the context of sport, there are many geographical factors that can determine the relevance of results. For instance, the nationality of a player, manager or other sports entity can indicate relevance, as users in a particular country may be more interested in sports figures who share their nationality. Similarly, the team a player is associated with, or the team itself, can also be a signal of relevance, as fans may seek information about teams from their country. In addition, the popularity of certain sports varies from region to region. For example, American football is more popular in the United States, while cricket is the most popular sport in India [29].

Sanderson and Kohler [30] analyzed the proximity between users' location and the places they sought information about, and the results revealed a trend: individuals tend to search for information related to their local surroundings. Moreover, there is a higher frequency of searches for information about cities compared to countries. When examining search topics, they identified correlations between various phenomena and the distances to relevant places. For example, searches for restaurants were generally closer to the user's location than for hotels.

One of the problems that can complicate geographical information retrieval is the disambiguation of names [13]. For example, Melbourne can be the capital of Australia or a city in the Florida state and Turkey can be a country or a bird. One possible solution is to use geographic coordinates instead of place names to disambiguate [23]. Also, user privacy concerns are an issue to consider in this implicit feedback type.

3.3 Evaluation Metrics

Metrics in information retrieval can be employed to measure the effectiveness of search engines and other retrieval systems. The metrics of precision and recall are frequently used for this purpose. However, these metrics are designed for unordered sets of documents. They assess performance based on the presence of relevant results without considering the order in which these results appear.

On the other hand, the *Average Precision (AP)* considers the position of relevant results. The *AP* of a query is defined as the average of the precision value obtained for the set of the top- k documents each time a relevant document is retrieved [21]. For example, for a query returning relevant results in positions 1, 3, and 7, the *AP* would be:

$$\frac{1/1 + 1/3 + 1/7}{3} = 0.492$$

The *Mean Average Precision (MAP)* of a system is the mean of the average precision for the queries of that system.

The *Mean Reciprocal Rank (MRR)* [39] of a system is the average of the queries' *Reciprocal Rank (RR)*. The *RR* of a given query is the inverse of the position of the first correct answer, i.e., if the highest ranked correct answer is r , then the query's *RR* will be equal to $\frac{1}{r}$. This metric favors

the correct answer being ranked before the incorrect ones, and it is most effective when there is a single correct answer. In the case of a single correct answer, the RR is equivalent to the AP, and consequently *MRR* is equal to the *MAP*.

Meanwhile, a system's *weighted Mean Reciprocal Rank* (*wMRR*) [36] is an expansion of the traditional *MRR* that considers click-through frequency. The *wRR* for each query is calculated by multiplying the number of clicks for each rank by the inverse of the rank, summing these values, and dividing by the total number of clicks. The system *wMRR* is the average of the queries *wRR*. To illustrate, consider a query with 13 clicks on the first result and three on the third. The *wRR* would be calculated as follows:

$$\frac{13 * 1 + 3 * 1/3}{16} = 0.875$$

Furthermore, the *Success at N* (*S@N*) [41] indicator measures the presence of correct answers in the top results. This metric corresponds to the proportion of queries with a correct answer in the top-N retrieved results. If the *S@2* of a system is 0.90, it can be inferred that by displaying the top-2 results, 90% of queries will have a correct answer available.

As seen before, clicks can be used to infer relevance, and this is already employed in *wMRR*. Another measure that uses clicks is the *Average Click Position* (*ACP*). The *ACP* of a system can be defined as the average position of all the received clicks. For example, consider a system that received eight clicks. Of these, five were on the first result, three on the second, and two on the third. The *ACP* is calculated as follows:

$$\frac{5 * 1 + 3 * 2 + 2 * 3}{8} = 2.125$$

3.4 Comparison of Search Systems

Following the implementation of an IR system, a critical step is assessing its performance in a production environment. If the system is used by many users, as is often the case with web search engines, variant systems can be deployed simultaneously [21]. This approach allows for a dynamic exploration of system modifications and enhancements. Once deployed, these systems can be compared using relevance measures to determine their impact on overall performance.

3.4.1 A/B Test

A/B testing is a technique commonly used in information retrieval to directly compare different variants of a system. It consists of randomly redirecting users to each one of the systems. Usually, a small proportion of users are directed to the variant system (1 to 10%), while the rest keeps using the original one [21]. The most straightforward setup is to evaluate a system with a modification in a single parameter. *Control* (version A) is the default or original version, and *Treatment* (version B) is the changed system [16]. However, this experiment can also have more variants being tested simultaneously. Kohavi and Longbotham [16] suggest starting with a phase in which version B is

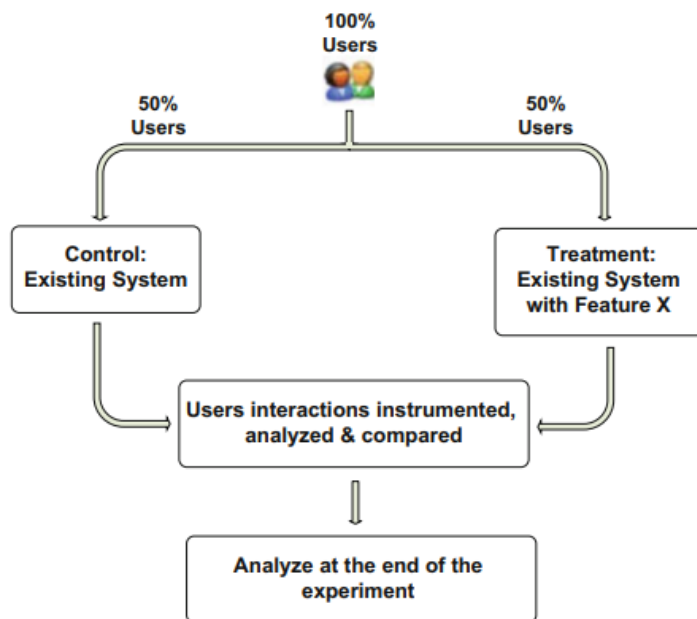


Figure 3.3: High level structure of an A/B test [16].

used at small percentages for error checking and then using an equal distribution between systems to maximize statistical power (Figure 3.3).

In information retrieval, this test can be used to evaluate changes in, for example, the ranking algorithm or user interfaces. In the case of an experiment in the ranking algorithm, the differences between systems can be measured through click data metrics, such as the average click position. A/B tests are considered easy to deploy, using randomization algorithms that direct users to one of the variants. They are also easy to understand and to make conclusions about the effect of the variation of a parameter by observing changes in user behavior.

3.4.2 Interleaving

Interleaving is an evaluation approach that consists of combining results from two retrieval algorithms in a single list shown to the user [27] (see Figure 3.4). Some of the most known interleaving algorithms proposed are described below.

Balanced Interleaving [10] combines two lists, A and B , into a new list I . The top results in I include the top results from A and the top results from B , ensuring that the difference between the counts of top results from A and B is at most 1. The algorithm works by tossing a coin, where the outcome is $t = (A)$ or $t = (B)$, and uses pointers pA and pB to track the highest-ranked documents not yet in I . If $pA < pB$, or $pA = pB$ and $t = (A)$, then the result associated with pA is added to the list, otherwise the one from pB is added. The pointers are adjusted after each addition. To score the variants, the system with the highest number of clicked results for a given query "wins" that ranking instance (also termed "impression"); otherwise, it is considered a tie. Balanced interleaving determines that the retrieval function with more impression wins is superior.

Ranking A	Ranking B	Presented Ranking
1. Napa Valley – The authority for lodging ... www.napavalley.com	1. Napa County, California – Wikipedia en.wikipedia.org/wiki/Napa_Valley	1. Napa Valley – The authority for lodging ... www.napavalley.com
2. Napa Valley Wineries – Plan your wine ... www.napavalley.com	2. Napa Valley – The authority for lodging ... om	2. Napa County, California - Wikipedia en.wikipedia.org/wiki/Napa_Valley
3. Napa Valley College www.napavalley.edu/	3. Napa Valley Wineries – Plan your wine ... www.napavalley.com	3. Napa Valley Wineries – Plan your wine ... www.napavalley.com/wineries
4. Been There Tips N www.ivebeenthere.co	4. Napa Valley College www.napavalley.edu/homex.aspx	4. Napa Valley College www.napavalley.edu/homex.aspx
5. Napa Valley Wineries www.napavinters.com	5. Napa: The Story of an American Eden ... www.amazon.co.uk/books?isbn=...	5. Napa: The Story of an American Eden ... www.amazon.co.uk/books?isbn=...
6. Napa County, Califor en.wikipedia.org/wiki	6. Napa Valley – The authority for lodging ... om	6. Napa Valley – The authority for lodging ... om
	7. Napa Valley – The authority for lodging ... om	7. Napa Valley – The authority for lodging ... om
	8. Napa Valley – The authority for lodging ... om	8. Napa Valley – The authority for lodging ... om
	9. Napa Valley – The authority for lodging ... om	9. Napa Valley – The authority for lodging ... om
	10. Napa Valley – The authority for lodging ... om	10. Napa Valley – The authority for lodging ... om

Figure 3.4: Example of interleaving for the query [napa valley] [27].

Team-Draft Interleaving [28] follows a similar approach to how sports teams are chosen in friendly matches. In each iteration, flip an unbiased coin, t . If $t = (A)$, add the next available result from A (i.e., the next highest result in A not already in I) to I , followed by the next available result from B . If $t = (B)$, follow the same procedure with B before A . This process continues, with a coin toss and two results added in each iteration. The output includes the interleaved list I and a record of which list contributed each result in I . To score the variants, each document is considered part of a team, and the ranker (team) with the most clicks on its documents wins the impression. A possible refinement involves ignoring clicks on results found in identical prefixes of both A and B , which enhances the algorithm's sensitivity [3]. A comparison with the Balanced approach of the working process of both algorithms is provided in Figure 3.5 for better understanding.

Probabilistic Interleaving [7] is similar to Team-Draft but has two main differences. First, it takes just one result per iteration, which could lead to the entire list I being made up of results from A alone in an extreme scenario, but, on average, A and B contribute an equal number of results. Second, instead of directly selecting the next available result from a list, in Probabilistic Interleaving, results are chosen based on a probability that decreases with the result's position in the ranking. For instance, when selecting the first document in I from A , a_1 is most likely chosen, followed by a_2 , a_3 , and so on, with decreasing likelihood. To assign scores to the ranker variants, a more sophisticated approach is used: given an interleaved ranking I , all possible combinations that could have produced I are computed. Then, for each possible combination, the algorithm uses the same logic as in Team Draft interleaving to determine which retrieval function wins the impression. Finally, it assigns each impression a probabilistic outcome based on the likelihood of each impression winning under all possible consistent combinations of coin tosses [27].

3.5 Summary

The evaluation stage is a fundamental aspect of implementing an IR system. Analyzing results using a set of metrics makes it possible to assess and compare the performance of these systems, particularly in terms of the quality of the results returned.

Rank	Input Ranking		Interleaved Rankings					
	A	B	<i>Balanced</i>		<i>Team-Draft</i>			
			A first	B first	AAA	BAA	ABA	...
1	a	b	a	b	a ^A	b ^B	a ^A	
2	b	e	b	a	b ^B	a ^A	b ^B	
3	c	a	e	e	c ^A	c ^A	e ^B	
4	d	f	c	c	e ^B	e ^B	c ^A	
5	g	g	d	f	d ^A	d ^A	d ^A	
6	h	h	f	d	f ^B	f ^B	f ^B	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Figure 3.5: Detailed execution of Balanced and Team-Draft interleaving algorithms [28].

We investigated several methods for obtaining implicit relevance feedback from user's interaction with the system. These techniques may have limitations, such as being noisy, but prove instrumental in understanding document relevance and enhancing the overall quality of search engine results, as explicit feedback is hard to acquire and generalize. Combining multiple methods may offer advantages, reducing the limitations inherent in individual approaches.

We then analyzed a set of metrics that can be employed to assess retrieval quality in an ordered set of results. The chapter concluded with techniques for comparing information retrieval systems. The deployment of variant systems serves as a direct method for evaluating changes in retrieval algorithms. A/B testing and interleaving algorithms are often combined with click-through metrics to compare variants and assess the impact of changes on the search engine.

Chapter 4

Improving Relevance in Sports Search

4.1 Overview

The search system of zerozero is the focus of this work. The zerozero network has expanded to incorporate more sports modalities and domains, having seven different modalities and 11 domains. Their database has been growing and reached almost 2 million searchable documents by the end of 2023, which poses challenges in the performance and relevance of the retrieved results. In 2023, the Apache Solr¹ framework was gradually incorporated into the zerozero search system, starting with the specific search by entity and later extended to the generic search page and the search bar. The use of this framework facilitated a faster and more structured search experience, as well as the exploration of more advanced ranking algorithms.

In Section 2.4, we described zerozero’s system in terms of the Search User Interface input, control, and informational features, compared it to other sports-related website search engines, and concluded that the zerozero platform offers plenty more options to the user to customize the search. Furthermore, an analysis of the ranking of search results was conducted, which led to the conclusion that there is at least one other platform where the results are affected by the popularity of entities. zerozero also employs a mechanism for this through clicks on entities’ pages.

4.2 Current zerozero Search System Architecture

The current system’s architecture is presented in Figure 4.1. It contains three main modules: a PHP application, containing a retrieval and an indexing service, a Solr server, and a MySQL database. These modules run on separate machines, with the database and the Solr server being allocated to only one each. Upon a user query, the retrieval service treats the query string and builds the URL, which is then forwarded to the Solr server. Then, the query is run in the Solr module, with the matching documents ordered based on zerozero ranking function. Subsequently, a *SolrObject* comprising the set of matching documents is returned to the retrieval service. Finally, the service handles this data object and presents the results to the user.

¹<https://solr.apache.org/>

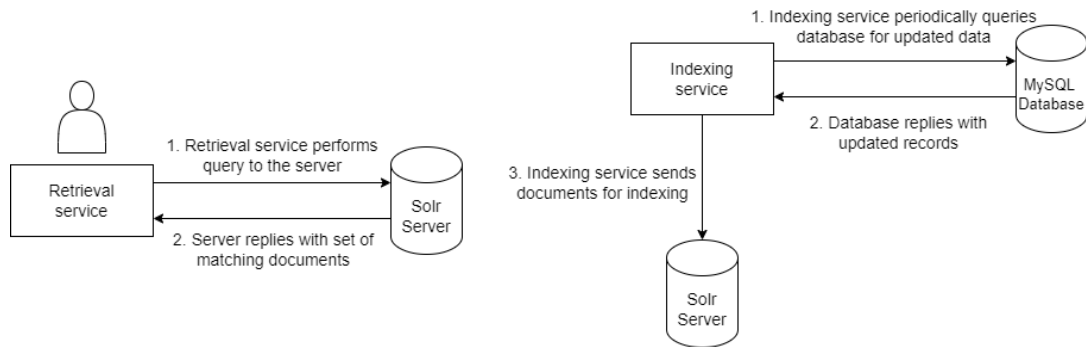


Figure 4.1: System architecture for retrieval and indexing processes and main execution flow steps.

The zerozero Solr server has an index for each of the nine defined entities (detailed in Section 2.4) plus a general one (implemented as Solr cores). An entity index contains all of that entity's documents, and it is used when the search is made by entity, i.e., when the user uses the entity filter or opens the specific entity search page². In contrast, the general index contains all documents from all entities. It is used upon a search that does not specify the entity, which happens in the search bar and in the SERP upon a query submission at the search bar.

In the indexing process, the indexing service periodically fetches the database for updated information every 3 minutes. For this, the service executes a query for records in the database containing a flag about whether or not they have been updated since the last information fetch. This includes records that have been inserted or updated. The database responds with the records, and then the indexing service builds a new document for each of those. Finally, the documents are sent to the index of the respective entity and the generic index in Solr to be used in the search process.

The ranking function used to order results employs six signals: the equivalence between the sport the user is in at the time of the search and the sport of the documents, the equivalence between the query and the documents in terms of whether they are both or neither of the type edition, the score assigned by the Solr parser to the documents given the query, the search index (*indice_pesq_language*) and the internal ranking (*zzranking_language*) of the documents for the domain in which the user is located, as a way of handling past and recent popularity, and an attribute that deals with the merit of the entities (*boost_value*) which, for example, gives more value to international players and to teams that have already won international titles. Firstly, the documents are sorted according to sport, i.e., the documents corresponding to the sport the user is in are displayed first. Next, they are sorted by the *edition* factor, whereby if the query contains numerical terms, the documents of the edition entity type are ranked higher, otherwise they are ranked lower than documents of the other entity types. The third and final ranking criterion consists of a combination of the remaining four signals:

²<https://www.zerozero.pt/jogadores> is the search page for players

$$0.8 * Solr_score + 0.1 * zzranking_language + 0.099 * boost_value + \\ + 0.001 * indice_pesq_language$$

4.3 Solr Framework Impact on zerozero

To analyze the impact of incorporating this framework on the relevance of the results retrieved by the system, we used zerozero query logs referring to a period before the introduction and a period after. After the introduction of Solr, not all of the clicks were being saved in the logs because of the transition (only clicks on player, competition, and edition results were being saved correctly). On the 28th of March 2024, this was corrected, and all clicks resulting from searches in the search bar and on the results page began to be saved. Therefore, our analysis focused on the 15-day period that began on March 29 and ended on April 12.

For a more accurate comparison between systems, we followed the suggestion from the zerozero team and considered the same period in the previous year, 2023, since searches on the platform tend to vary according to the time of year. However, this comparison necessarily has some limitations. The first one has to do with the fact that the pre-Solr system kept logs of the interactions on the search bar exclusively, where only ten items were shown to the users. Upon the introduction of Solr, the logs were also captured for the interactions in the SERP, where 20 results per page are shown (except when there are fewer results retrieved for a query). A direct comparison would harm the version of the system with the Solr framework, so we removed the clicks in positions higher than ten at a given point.

Another limitation has to do with the change in the search interface (as was described in Section 2), introducing, among other things, a different design for the results. In our study on click analysis, we referred to the concept of snippet bias, which, in this case, we can adapt to the result boxes shown, which contain information that can influence the user's click, even though they are not considered snippets.

4.3.1 Dataset

For this analysis, we used the query logs provided by the zerozero team as the dataset, which contain information about the search and the result clicked on. In Table 4.1, we find the description of the query log fields used in this work, and in Listing 4.1, we present an artificial entry simulating its content. This dataset only contains searches made by humans, i.e., the searches identified as being made by bots were removed. This processing was done in advance by the zerozero team.

We will refer to these datasets as the pre-Solr Query Set (PSQS) and Solr Query Set (SQS). The PSQS corresponds to the period between March 29 and April 12, 2023, and originally had 426,464 entries, while the SQS corresponds to the same period in 2024 and had 432,497 entries. Next, we will describe the preparation steps for this data.

Table 4.1: Description of main fields from a zerozero query log.

Field	Description
time	Timestamp representing the moment the query was submitted
page	Page of the clicked result
language	Domain in which the query was made
search_order	Position of the clicked result
search_string	Search query submitted by the user
tp_item	Identifier of the entity type of the clicked result
fk_item	Identifier of the result (unique among its entity type)
geo_location	Geographic location (country) of where the query was made
random	Identifier of the system version (0 is the original zerozero system)

```

1  "id", "time", "page", "language", "search_order", "search_string", "tp_item", "fk_item", "
   geo_location", "random"
2
3  "1", "2024-01-01 00:00:01", "/competicao/liga-portuguesa?search=1", "pt", "1", "liga
   portugal", "4", "9999", "pt", "0"

```

Listing 4.1: Example of query log entry.

4.3.2 Data Preparation

A query log from zerozero contains, among other things, the original query text that the user typed upon a search. However, this is not the query text sent to the Solr server (previously MySQL database), as some data cleaning operations are performed by the zerozero system beforehand. Transforming the initial user query into the final query sent to Solr allows for a consistent grouping of records, so we replicated these procedures.

First, queries with an empty search string were removed as the system does not return results in these cases. Then, all search strings were converted to lowercase. For example, the queries [Cristiano], [cristiano] and [CRISTIANO] are sent to the Solr server as [cristiano]. Next, we replaced specific expressions, such as SQL instructions, and some non-alphanumeric characters with white spaces in the search string. Then, white spaces at the beginning and end of the search strings were removed, and multiple spaces were converted into one. Finally, the characters were converted to their ASCII representation, removing accents, for example. Thus, the queries [Alvaro Pacheco] and [Álvaro Pacheco] are converted to [alvaro pacheco], as in the zerozero system.

It is important to note that the ranking of the results shown changes depending on the domain. For example, for the query [santos], the team Santos³, on February 27, was in the first position

³<https://www.zerozero.pt/equipa/santos?search=1>

Table 4.2: Click ranking distribution comparison.

Ranking	PSQS		SQS		Difference (%)
	Clicks	Percentage	Clicks	Percentage	
1	311087	72.95%	311299	74.36%	1.41%
2	56753	13.31%	45195	10.80%	-2.51%
3	22509	5.28%	20379	4.87%	-0.41%
4	13474	3.16%	12236	2.92%	-0.24%
5	7893	1.85%	7708	1.84%	-0.01%
6	5547	1.30%	6512	1.56%	0.26%
7	3630	0.85%	5602	1.34%	0.49%
8	2702	0.63%	3900	0.93%	0.30%
9	1603	0.38%	3269	0.78%	0.40%
10	1261	0.30%	2538	0.61%	0.31%

in the Brazilian domain while in the Portuguese domain it was in the second position. In order to maintain a consistent analysis and a correct training procedure, we grouped the searches not only by query text but also by domain.

After applying the cleaning operations to both datasets, five entries were removed from the PSQS and none from the SQS.

4.3.3 Search Characterization

Before grouping the searches by query, we gathered some statistics on each dataset. The distribution of clicks by the position of the result is shown in Table 4.2. This considers only the clicks in the first ten positions in the SQS. Overall, the number of clicks decreases as the ranking increases. For the SQS, the click share was more spread out after the first position compared to the PSQS. With Solr, the first position had a slightly higher share, but up to the sixth position, it was lower when compared to the PSQS. From the seventh position onwards, there was a higher click share again. In the SQS, there were a total of 13,859 clicks on results outside the top 10.

The distribution of searches by domain is presented in the Table 4.3. Most searches continue to be carried out via the Portuguese domain, and the Brazilian domain continues to have a significant weight despite the decrease in SQS. These two domains account for the vast majority of searches, while the others register insignificant figures. It should also be noted that the Dutch domain had its first searches. Figures 4.2 and 4.3 show the distribution of queries per day in both query sets.

We then grouped the dataset by the query text and language using the procedure mentioned before. This left us with 102,629 different search strings in the PSQS and 106,862 in the SQS. We also observed the most frequent search strings in both query sets. The top 20 queries are presented

Table 4.3: Click per domain distribution comparison.

Domain	PSQS		SQS		Difference
	Clicks	Percentage	Clicks	Percentage	
pt	262044	61.45%	302164	69.86%	40120
br	145542	34.13%	109672	25.36%	-35870
es	10504	2.46%	9587	2.22%	-917
en	4178	0.98%	6451	1.49%	2273
fr	2985	0.70%	2004	0.46%	-981
us	665	0.16%	1026	0.24%	361
it	391	0.09%	414	0.10%	23
de	144	0.03%	563	0.13%	419
mx	4	0.00%	215	0.05%	211
ar	2	0.00%	395	0.09%	393
nl	0	0.00%	6	0.00%	6

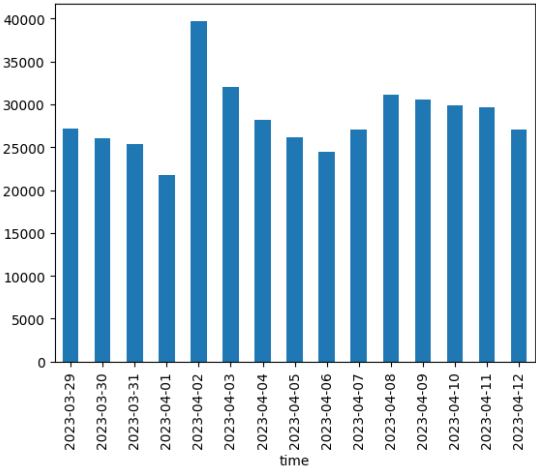


Figure 4.2: Daily search distribution (pre-Solr Query Set).

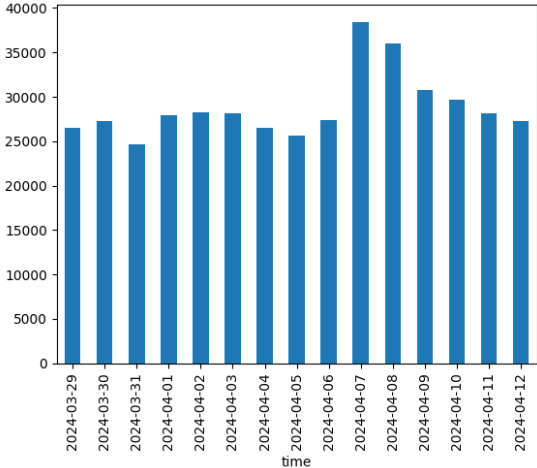


Figure 4.3: Daily search distribution (Solr Query Set).

Table 4.4: Top 20 most frequent queries (pre-Solr Query Set).

Query	Domain	Clicks	Percentage
benfica	pt	2127	0,50%
liga	pt	1673	0,39%
porto	pt	1329	0,31%
sporting	pt	1219	0,29%
ronaldo	pt	1002	0,23%
braga	pt	997	0,23%
botafoogo	br	854	0,20%
inter	pt	838	0,20%
gremio	br	779	0,18%
inter	br	771	0,18%
palmeiras	br	737	0,17%
messi	pt	727	0,17%
atletico	br	703	0,16%
vitoria	pt	679	0,16%
america	br	668	0,16%
bahia	br	650	0,15%
santos	br	644	0,15%
flamengo	br	635	0,15%
palmeiras	pt	607	0,14%
vasco	br	592	0,14%

Table 4.5: Top 20 most frequent queries (Solr Query Set).

Query	Domain	Clicks	Percentage
benfica	pt	3213	0.74%
sporting	pt	2615	0.60%
braga	pt	1645	0.38%
porto	pt	1584	0.37%
ronaldo	pt	1413	0.33%
varzim	pt	1241	0.29%
palmeiras	pt	1125	0.26%
flamengo	br	959	0.22%
arouca	pt	874	0.20%
leixoes	pt	861	0.20%
alverca	pt	847	0.20%
liverpool	pt	838	0.19%
palmeiras	br	833	0.19%
estoril	pt	803	0.19%
vitoria	pt	797	0.18%
boavista	pt	795	0.18%
portugal	pt	792	0.18%
botafoogo	br	790	0.18%
belenenses	pt	770	0.18%
famalicao	pt	759	0.18%

in tables 4.4 and 4.5. In SQS, the weights have increased, indicating a greater concentration of clicks on the most frequent queries.

4.3.4 Evaluation Process

In order to evaluate and compare the different versions of a system, different metrics were tested for both query logs to draw conclusions. Therefore, we used some of the metrics already presented in Chapter 3: *Average Click Position (ACP)*, *Mean Reciprocal Rank (MRR)*, *weighted Mean Reciprocal Rank (wMRR)*, and *Success at N (S@N)*. As the search bar of zerozero displays up to 10 results and only four are visible at any given time, we wanted to register this metric for the top results ($N \in [1,5]$) in order to examine its influence.

To calculate the *Reciprocal Ranks (RR)*, we needed to annotate the correct answers for each query. In their study, Liu et al. [19] explored the accuracy of automatically annotating the most clicked result as the correct answer for navigational queries. Their findings revealed that over 96% of the annotated answers were correct, suggesting that it is feasible to automatically annotate the most clicked result as the correct answer for this type of queries. Navigational queries are those where the intent is to locate a particular resource [9]. In the context of zerozero, the queries are expected to be mainly navigational as users tend to search for a precise entity [4]. Therefore, we considered the most clicked result as the correct answer for a query. When two or more results had the same number of clicks for a given query, we considered the highest ranked result to be the correct answer. Before that, for the sole purpose of this study of the impact of Solr on the zerozero search system, we removed records where the click was on a position lower than 10 in the SQS.

These metrics were also tested for the set of the 200 most frequent queries of each query set, as this represents 16.60% of the PSQS and 20.49% of the SQS.

4.3.5 Results

The evaluation results are presented in Table 4.6. The first line shows the results obtained from the evaluation of the PSQS. The ACP was 1.625, indicating that, on average, the users clicked between the first and second position results. The MRR of the queries was 0.823, while the wMRR was slightly lower at 0.806. 72.1% of queries found success on the first result. The addition of a second result led to an increase of over 15%. With the top 3 results, the system achieved a success rate of 90.9%. As the number of results is increased, the gain decreases. With the first five results, the system provides the correct response to over 96.0% of queries. The pre-Solr system answered 94.0% of queries correctly with the first four results, which rises to 98.5% if we only consider the 200 most frequent queries. The reduction in the number of results shown simultaneously in the search bar is therefore not detrimental since without the effort associated with scrolling, 94% of queries would have the correct result visible. In the set of the 200 most frequent queries, the first result satisfies 88.5% of queries, and adding the second increases this value to over 94%.

The SQS has an ACP of 1.707, slightly worse than PSQS but still between the first and second positions. The MRR and wMRR were slightly better, as was the S@1. From there on, the PSQS

Table 4.6: Evaluation metrics results.

System	ACP	MRR	wMRR	S@1	S@2	S@3	S@4	S@5
Pre-Solr	1.625	0.823	0.806	0.721	0.854	0.909	0.940	0.960
Solr	1.707	0.828	0.814	0.732	0.853	0.906	0.937	0.955
Pre-Solr most frequent	-	0.930	0.879	0.885	0.945	0.975	0.985	0.990
Solr most frequent	-	0.948	0.892	0.910	0.970	0.985	0.990	0.990

had higher values. By showing four results at the search bar, the SQS provides a correct result for 93.7% of queries and 99.0% of the 200 most frequent searches from this query set. When considering only that set of queries, which is more significant in this system, SQS achieved better results.

It is important to emphasize that by showing more results, users are tempted to pay less attention to the top items, which may explain the wider distribution of clicks by position from the sixth result onwards. It is also worth noting that when submitting a query in this new system (and not clicking directly on a result), users are redirected to the SERP instead of going directly to the first result page. This way, they are forced to confirm their choice, which makes the evaluation results of the SQS more solid. By showing only four results at once in the search bar, instead of the eight that were shown in the previous system, there should be a tendency for clicks to be more concentrated on the first four results. However, there is no noticeable difference in the results that would allow us to conclude that users have been influenced by this situation.

For the set of all queries, SQS obtained better values for three metrics, but only for S@1 was the improvement greater than one hundredth (+0.011). It achieved lower values for the remaining metrics. For the set of the 200 most frequent queries, the introduction of the framework led to improvements, with SQS obtaining better values for all metrics except for S@5, where the results were the same.

4.4 Problem

During a search, Solr receives an information need expressed as a query and returns the documents that match the main text fields, such as the name of the entity, the name of the team it represents, or the name of the country of its nationality. For example, for the query 'Portugal', all Portuguese entities will be retrieved.

In addition to the match between the query and the document, the ranking function used by the zerozero search system incorporates other signals. By presenting the results corresponding to the sport where the user is engaged at the top, relevant results from other sports may be devalued. It also encompasses attributes relating to the popularity of the entity in the zerozero network (according to the number of accesses to its web pages), which include recent and older clicks. Prioritizing past popularity may result in entities that were once relevant but are no longer so being

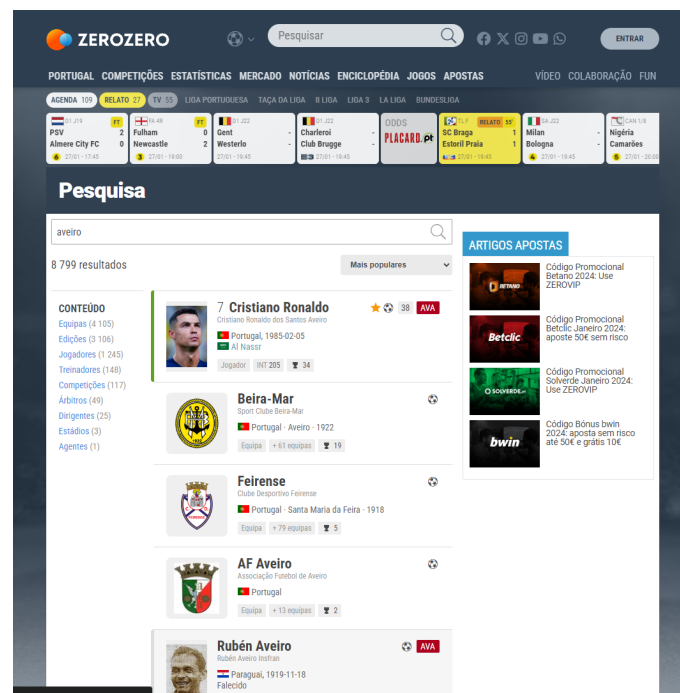


Figure 4.4: Search Engine Results Page for the query [aveiro] in zerozero.pt.

ranked higher than others that are currently more relevant. Furthermore, the attribute relative to sporting merit is computed based on some predefined achievements. For players, it assigns value to international players and Ballon d’Or winners, but there are other merit signals that could help discern the relevance of sports entities.

Besides that, entities that are returned as a result of the textual matching of fields other than their name but which are then well valued in other signals of the ranking function may be better ranked than entities that match on the name. An example is the query [aveiro] (see Figure 4.4) in an attempt to find AF Aveiro⁴. However, the first result shown is Cristiano Ronaldo⁵ (Cristiano Ronaldo dos Santos Aveiro), who only contains the term in his full name and is not even known by same.

These problems arise from manually weighting attributes in the ranking function, leading to unsatisfactory results for some queries. It is challenging to manually balance these weights to achieve a function capable of delivering relevant results across a wide range of queries. This complexity often results in either overvaluing or undervaluing certain signals. On the other hand, the current ranking process relies heavily on predefined criteria and assumptions about what constitutes relevance. Given the size and completeness of the zerozero data collection, more signals could be included in this process to help assess relevance.

⁴<https://www.zerozero.pt/equipa/af-aveiro/84454> (accessed January 2024)

⁵<https://www.zerozero.pt/jogador/cristiano-ronaldo> (accessed January 2024)

4.5 Proposed Solution

Given that the ranking function was constructed manually, our solution was to incorporate an automatic function. To this end, we explored the concept of Learning to Rank, which involves applying machine learning techniques to information retrieval in order to construct ranking models using training data [18]. Since the search system is based on the Solr framework, we explored and incorporated the Learning to Rank module of this framework into the search pipeline. This included applying a re-ranking query to the initially retrieved documents to order them based on a ranking function learned from the query logs, which served as the training data.

The original ranking function of zerozero enables the identification of three key concepts that were employed to assess the relevance of documents for a query. These were sporting merit, popularity, and geographic location (in this case, the domain in which the query was made). Furthermore, there were other signals that could be associated with these concepts, which can help express the relevance of entities that weren't being incorporated into the ranking process. In Section 3.2, we explored some ways of obtaining relevance feedback from user interaction with the retrieval system, including click data and geographical information from the query. Our solution also involved incorporating additional signals other than those employed by the original zerozero system. In collaboration with the zerozero team, we delineated a set of additional attributes associated with merit, including the number of trophies and awards won, values indicative of popularity, such as the number of clicks per domain and per time frame, in addition to total clicks, and also signals associated with geographical location, namely the match between the location of the query and the nationality of the entities or clubs they represent or have represented in the past.

We applied our solution in the Portuguese and Brazilian domains, which account for the vast majority of search traffic on the zerozero network. To assess its performance, the system was adapted to run in an A/B test configuration, as outlined in Section 3.4.1. This enabled the solution to be used online on the zerozero platform and to be directly compared with the original engine, allowing for the gathering of consistent click-through data, which was then employed to evaluate the systems in question using the metrics explored in Section 3.3. Furthermore, it permitted the testing of not just one but several systems simultaneously. Figure 4.5 shows a conceptual view of the proposed solution.

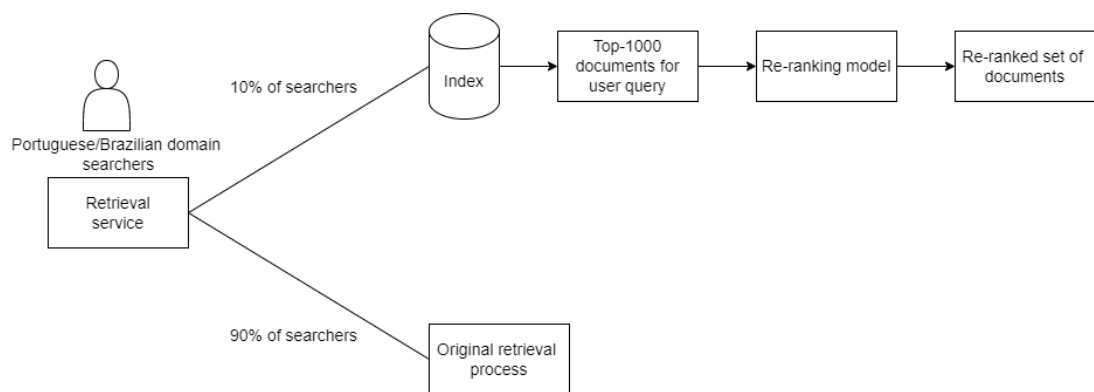


Figure 4.5: Proposed solution conceptual view.

Chapter 5

Learning to Rank in Sports Search

In this chapter, we will cover the implementation of a Learning to Rank solution for the zerozero search system. We will start by discussing the Learning to Rank module in Apache Solr, an open source search framework used in the zerozero system, and examine the most common approaches to Learning to Rank. We then describe the implementation process of the Learning to Rank module of Solr within the zerozero system, from the definition of signals to the model training process and deployment details.

5.1 Learning to Rank

Upon inputting a query, a user expresses their information need and initiates the retrieval process [1]. This involves the search engine retrieval algorithms analyzing the collection of documents to select the ones that match the query. The results are then ranked according to their relevance to the query, with the most relevant results presented to the user. The results may be deemed unsatisfactory in terms of relevance, leading to the need to optimize the algorithms: "*Ranking is a central problem in IR*" [18].

To improve the ranking algorithms, parameters can be manually tuned (for example, by changing the weights of fields), but a perfectly tuned model that performs great on a given query or validation set may perform poorly on unseen queries (over-fitting) [18], i.e., the values that work for a given set of queries may not improve (or may even decrease) the relevance of results for another set. Problems may also arise when indexing unseen documents. The ranking models may fail to assess the relevance of new documents due to being over-fitted to the validation set. This approach not only proves to be inefficient for systems that update their document collections or that receive high amounts of queries but is also time-consuming [26].

On the other hand, machine learning (ML) algorithms are effective in automatically tuning parameters and avoiding over-fitting. Learning to Rank (LTR) is the task of constructing a ranking model using training data [18]. It consists of ML algorithms applied to information retrieval that help assess the relevance of documents for a query to improve the quality of search results.

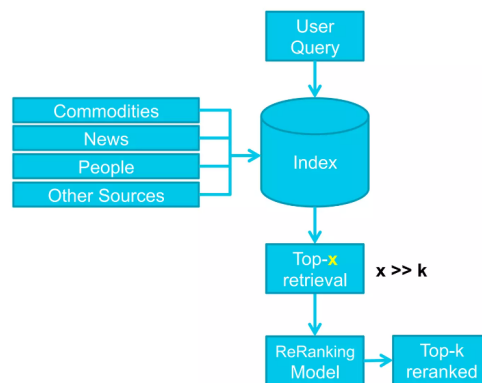


Figure 5.1: Search pipeline of a system that uses Learning to Rank [26].

5.1.1 Learning to Rank in Solr

The first attempt to use LTR in Solr dates back to 2015, when software engineers from Bloomberg¹ introduced a plug-in, showcasing its functionality in a comprehensive recorded presentation. Subsequently, this plug-in was incorporated into Solr at the beginning of 2017, thanks to the collaborative efforts of two distinct engineering teams within the company. Next, we describe how the Solr Learning to Rank module works, drawing inspiration from the pioneering work conducted by Bloomberg [26].

As is illustrated in Figure 5.1, instead of retrieving the top- K results, the search engine retrieves the top- X results (where X is much larger than K , for example, $K = 10$, $X = 1000$). Then, these top- X results are sent to the LTR module, where they are reordered by the ML-trained models in what is called the re-ranking process. Not only is it expected that the top- X retrieved results contain the relevant documents for a query, but also LTR algorithms are more costly than the first ranking process, which is why a smaller set of documents is used in this stage. Finally, the top- K retrieved results from the LTR model are presented to the user.

The ranking model needs to be trained offline before being applied. This requires training data like in any ML model, and it is the only part that needs to be done outside the Solr framework [34]. The process can be followed in Figure 5.2. LTR models use a set of user queries and the list of documents retrieved by the system for that query. Query-document pairs with relevance labels such as "excellent match" or "good match" [2] can be used to assess the relevance of a document for a given query.

Each retrieved result is characterized by a vector of features. A feature is a property of the document or the query that assists the model in assessing its relevance for a specific query. The popularity, the length or the age of the document, or the number of clicks it had in the past month are some examples of features that can be used to characterize documents, while the country of origin of the query is an example of a query feature. Subsequently, to score and rank the documents, LTR models use a parameterized function of multiple features computed based on the document and the query [14]. The learning algorithm goal is to find the best combination of

¹<https://www.bloomberg.com/company/>

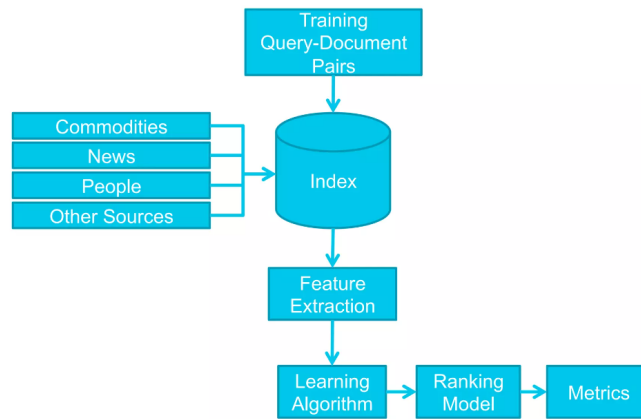


Figure 5.2: Training pipeline of a system that uses Learning to Rank [26].

features and their contribution (weights) that build the best ranking function for a given metric. A critical step is evaluating the combinations of features, and relevance metrics must be used. There are many learning algorithms applied to LTR, which can vary from linear models (as is the case of Ranking Support Vector Machine [10]) to Multiple Additive Trees (such as LambdaMART [38]) and Neural Networks, like RankNet [2].

5.1.2 Learning to Rank Approaches

Learning to rank aims to order a set of documents based on their relevance or importance to a particular query or user. In his survey, Tie-Yan Liu [18] proposed a categorization into three approaches for learning to rank algorithms: the pointwise approach, the pairwise approach, and the listwise approach.

The pointwise approach in LTR treats the ranking problem as a regression or classification task, where each query-document pair, expressed as a feature vector, is considered independently. This approach predicts the exact degree of relevance of each document, and these scores are then used to produce the final ordered list. This method is characterized by its simplicity but also by its inability to capture the relative importance or preference between different documents within a query context, and ranking involves predicting relative order rather than accurate relevance scores.

On the other hand, the pairwise approach focuses on relative order by directly comparing pairs of documents for a given query, both represented as feature vectors, which brings it closer to the concept of ranking. This can be seen as a classification problem, where the goal is to minimize the number of misclassified document pairs. The pairwise method implicitly captures the notion of preference and allows for more refined ranking decisions. Implicit feedback (such as click data) can be used to infer which of the two documents is more relevant. Nevertheless, Liu [18] acknowledges scalability and computational complexity challenges, particularly because the number of pairwise comparisons grows quadratically with the number of documents.

Finally, the listwise approach considers the entire list of documents associated with a query.

Rather than decomposing the problem into individual instances or pairwise comparisons, the listwise method aims to optimize the quality of the entire ranked list. By exploiting information about the entire ranking context, the listwise approach aims to produce more coherent and globally optimal rankings. According to Liu [18], the performance of this method is indeed better than previous approaches.

5.2 Applying Learning to Rank in zerozero

In this section, we will explore the implementation of an LTR module in the zerozero search system. We will begin with the configuration of a test system, then proceed to the definition of the features to be used by the model before moving on to the feature engineering process, which involves the construction of new features from existing data. Finally, it will conclude with a description of the process of training the models using the basic example provided by Solr. The query logs provided by the zerozero team were used not only for characterizing searches and evaluating systems but also for the process of training the LTR models.

5.2.1 Test Setup

Before implementing the LTR models in the zerozero system, it was necessary to configure the system in an A/B configuration. The objective was to have different system versions running simultaneously, with a distribution of the platform's traffic. To this end, an A/A test, also known as a null test, was initially implemented in line with the recommendations of Kohavi and Longbotham [16] to test the system's correctness regarding randomly redirecting users. Additionally, it was necessary to test the integration of re-ranking queries using the Solr LTR module. To do so, we defined a system that utilized a model with features that replicated the ranking function of the original system, aligning with the concept of A/A test. In this experimental phase, it was decided to redirect 1% of the sessions in the Portuguese domain to the system with the re-ranking query. In order to differentiate the system being used in the searches, the zerozero team offered to add an extra field that distinguished the old system from the new one in the database that stores the query logs. This initial system configuration provided a solid basis for further development.

5.2.2 Feature Definition

As previously stated, features are characteristics of documents or queries, and their purpose is to help assess the relevance of a document to a given query. To do so, they must be defined rigorously and consider the context in order to positively affect their use in the ML model. In the case of the zerozero system, the context is sports and encompasses multiple modalities and entities. Before defining them, we studied the data associated with each entity, whether in Solr documents, database records, or search logs. Considering this data and after several discussions with the zerozero team, we defined a set of features, which included text match, entity attributes, and query-document attribute value match, that can help infer the relevance of results in the search.

Table 5.1 provides a detailed description of each feature, accompanied by an indication of the specific iterations in which they were employed (in this work, we carried out two iterations of the systems, the second aimed at addressing and correcting the gaps identified in the first).

Table 5.1: Description of the features used to assess relevance.

Feature Name	Description	Iteration
original_score	Solr's score for the original query relative to text matching	1,2
sports_match	Whether the entity is of the same sport as the one chosen in the zerozero search	-
edition_check	Whether the entity and the query are both or none of the type <i>edition</i> . This scores editions positively when the user is looking for them (when the user types numbers) and negatively when they are not	1,2
indice_pesq_language	Zerozero defined attribute for handling popularity	1,2
zzranking_language	Zerozero defined attribute for handling popularity	-
boost_value	Zerozero defined attribute for handling sports merit. Assigns more value to players and teams that won international trophies, international players, and teams that won national trophies and play in the first division	1,2
is_glory	Attribute to indicate if the entity is considered a glory	1,2
is_international	Attribute for international status	2
number_international_matches	Attribute for the number of international matches played	1,2
is_retired	Indicator for the entity status (retired)	1,2
is_deceased	Indicator for the entity status (deceased)	1,2
is_recently_deceased	Indicator for recently deceased entity (if less than seven days have passed since the date of decease)	2
total_international_titles	Attribute for the number of international titles won	2

total_national_titles	Attribute for the number of national titles won	2
total_awards	Attribute for the number of awards won	2
level	Attribute for the level of a team, manager, competition, or edition. Main competitions/editions have higher values. Senior teams have higher values when compared to other tiers. Managers that coach senior teams have higher values as well	2
level2	Attribute for the level of a team. Main teams have higher values than B teams	2
ambito	Attribute for the ambit of a competition/edition (intercontinental, continental, national, regional, etc.)	2
clicks_total	Attribute for the total number of clicks in that entity's page	1,2
clicks_language	Attribute for the total number of clicks on a specific domain	1,2
clicks_last_week_all	Attribute for the number of clicks in the last 7 days across all domains	1,2
clicks_2_weeks_ago_all	Attribute for the number of clicks from 7 to 14 days across all domains	1,2
clicks_last_week_language	Attribute for the number of clicks in the last 7 days on a specific domain	1,2
clicks_2_weeks_ago_language	Attribute for the number of clicks 7 to 14 days on a specific domain	1,2
country_match	Whether the country of the entity and the origin of the query are equal	1,2
current_team_country_match	Whether the country of the entity's current team and the origin of the query are equal	1,2
all_teams_country_match	Whether the country of, at least, one of the entity's past teams and the origin of the query are equal	1

Among the features shown in Table 5.1, the first six correspond to signals used to order results in zerozero's original system. The rest were introduced as part of this work. At this stage, the zerozero team added the necessary fields to the documents on Solr regarding the number of clicks

on the entity's pages and the countries of past teams. While some features simply correspond to the value of a particular field present in the Solr documents, others are more complex and require data processing in order to be used correctly. The following section describes these operations.

During this process, we encountered a limitation concerning the fact that the search logs do not store information about the sport that is selected when the user clicks on a result and, therefore, it is not possible to train the model with the *sports_match* feature.

5.2.3 Feature Engineering

Feature engineering is a crucial task in preparing data to be used by machine learning models. It involves transforming raw data into features that enhance predictive performance [25]. In this section, we explain the various techniques employed to exploit the available data for use in the model.

To begin with, we address the categorical variable *status*, which can have the values "active", "retired", or "deceased". Employing one-hot encoding, we translated this variable into the binary features *is_retired* and *is_deceased*, with values of 0 or 1. Notably, the absence of both flags denotes an "active" status.

The fields available that provide geographical information are the country related to an entity (which is the nationality in the case of people and the country of competitions, teams, and stadiums) and the country of the teams related to an entity at the moment or in the past (for players and managers this corresponds to the teams they represent or represented). Thus, we defined the features *country_match*, *current_team_country_match*, and *all_teams_country_match* where we used the Solr filter query functionality to match these document fields with the country from where the query was made. These features have a value of 1 if there is a match and 0 otherwise.

Next, we will outline the processing steps for the features related to clicks and recently deceased entities.

5.2.3.1 Click Data

As reported in Section 3, even though click data can be biased, it is still a meaningful way of assessing relevance through implicit user feedback. Taking advantage of the fact that zerozero stores data on user clicks when searching on their website, we created features that use this information as a way of handling popularity.

Using a Python script, we extracted the query logs from the database and processed them to calculate the values needed for our features. To do so, we grouped the clicks by entity, considering the domain and the time at which they were made. Listing 5.1 shows one example record from the output of the processing. We only calculated these fields for the Portuguese and the Brazilian domains as our work only addresses these domains.

```

1  "tp_item","fk_item","clicks_total","clicks_2_weeks_ago_all","clicks_last_week_all
   ", "clicks_2_weeks_ago_pt", "clicks_last_week_pt", "clicks_2_weeks_ago_br", "
   clicks_last_week_br", "br", "pt"
2  "4", "9999", "10000", "1000", "1050", "935", "965", "40", "50", "400", "9350"

```

Listing 5.1: Example record from the processing of clicks.

This data was then stored in a new table in the MySQL database, and the entities that had their fields updated were flagged to be re-indexed to Solr. Upon their indexing, the up-to-date click information is added to each document to be used by the LTR model in multiple features.

The use of click data to infer relevance has a limitation, as newly introduced documents are devalued because they do not have any clicks on their pages.

5.2.3.2 Decease Date

After carefully analyzing the search logs and discussing with the zerozero team, we concluded that a recent death increases the volume of searches for the deceased entity. We, therefore, studied the impact of Artur Jorge's² death on searches and visits to his player page on the zerozero network (at that time clicks on manager pages were not yet captured in the logs after the introduction of Solr). The former Portuguese player and manager passed away on 22 February 2024, and, as we can see in Figure 5.3, there was a significant increase in the number of accesses compared to what was usual. This peak in visits lasted until the 26th of the same month, but in the following two days, the number of visits was still much higher than usual before the decease. Therefore, we decided to consider that the relevance of this entity would be higher than usual for a period of seven days. So the *is_recently_deceased* feature has a value of 1 if less than seven days have elapsed between the current date and the date of the entity's decease, and 0 otherwise. In the event of the entity's passing, the document of the respective entity is edited, adding the decease date.

5.2.4 Model Training

Training the model is the only step that needs to be done outside of Solr. However, a practical training example [33] is provided, which takes a log of clicked items (query-document pairs) and the number of clicks used to judge relevance, extracts the values of the features from Solr for that query-document pair, and shapes them so that they can be used by an ML model training library. The provided example uses a pairwise approach and the large linear classification library LIBLINEAR [5] and produces a linear model compatible with Solr's LinearModel Class³. Given the lack of practical examples, we followed this recommendation and adjusted the available code according to our needs, which mainly were to minimize the number of queries made to Solr to extract the features, given it is the most time-consuming part considering a query log with tens of thousands of entries.

²<https://www.zerozero.pt/jogador/artur-jorge/15199?search=1> (accessed April 2024)

³https://solr.apache.org/docs/8_7_0/solr-ltr/org/apache/solr/ltr/model/LinearModel.html

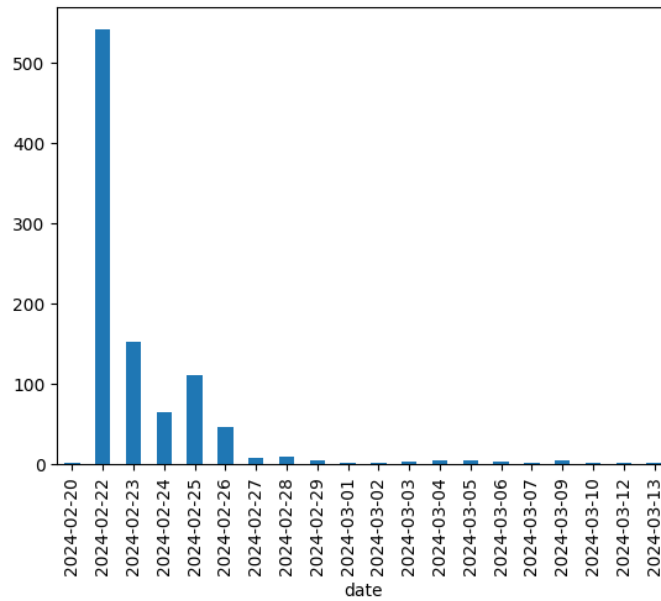


Figure 5.3: Distribution of daily accesses to Artur Jorge’s player page on the zerozero network.

The number of clicks for each query-document pair was calculated from the query logs provided by the zerozero team after the data preparation process described in the previous section had been applied. Subsequently, to facilitate the utilization of the training example provided by Solr, the data was converted into a format that could be directly used as input. In Listing 5.2, we provide example records of the training input entries used to train the model. These specific examples would represent that for the normalized query [ronaldo], the document *player-1* had 15000 clicks from Portugal (geographic location, not the Portuguese domain) while the document *player-131* had 4000.

```

1      query | docID | score | source
2      ronaldo | player-1 | 15000 | pt
3      ronaldo | player-131 | 4000 | pt

```

Listing 5.2: Example records from the training entries used as input for model training.

The program would then extract the feature values for each document given the query text (query-document pair) by directly querying the defined Solr core. With these feature values, a pairwise approach is followed. The program will proceed to compare the pairs of documents for a query and calculate the differences between them. For example, let us consider the following documents and their corresponding feature values:

- *player-1*: is_glory: 1, number_international_matches: 206;
- *player-131*: is_glory: 0, number_international_matches: 6.

Table 5.2: Example of geographic location features for *player-1*'s document and query text [ronaldo].

Query		Features	
origin	country_match	current_team_country_match	all_teams_country_match
pt	1	0	1
br	0	0	0
es	0	0	1
it	0	0	1

For this pair of documents and this query, let us also consider the number of clicks in Listing 5.2. The output of the differences calculated by the program will be "1:+1 2:+200", meaning that the more relevant document (in this case *player-1* because it has more clicks) has a positive difference of 1 in the first feature and 200 in the second. Then, the computed differences for all pairs of clicked documents for the same query text are sent to the LIBLINEAR training library to generate the feature weights that will result in our linear model.

When training the model with the geographic location features, we obtain the values for the features through a query containing the query text and the source. This value is then compared with the defined document fields, with a feature value of 1 being assigned if the countries match and 0 otherwise. Documents with the same query-document pair and the same set of values for all features are grouped, with the number of clicks being summed to produce an authentic relevance judgment. Listing 5.3 provides a concrete example of a player, designated as *player-1*, who has played in teams from Portugal (pt), Spain (es), and Italy (it) but currently plays in a team from Saudi Arabia. Table 5.2 illustrates the values of the features for each query-source pair. The program will then merge the pairs *ronaldo-es* and *ronaldo-it*, as they have identical feature values. The remaining features will also be identical, as they depend solely on the document or query-document pair, which is identical in this case. Consequently, the model training will have three instances, as shown in Listing 5.4 ordered by relevance. In the absence of geographic location features in the training data, the source is ignored.

```

1      query | docID | score | source
2      ronaldo | player-1 | 15000 | pt
3      ronaldo | player-1 | 3000 | br
4      ronaldo | player-1 | 2000 | es
5      ronaldo | player-1 | 1500 | it

```

Listing 5.3: Example records from the training entries for *player-1*.

```

1      Relevance, {Features Map}

```

```

2   15000, {..., country_match: 1, current_team_country_match: 0,
    all_teams_country_match: 1}
3   3500, {..., country_match: 0, current_team_country_match: 0,
    all_teams_country_match: 1}
4   3000, {..., country_match: 0, current_team_country_match: 0,
    all_teams_country_match: 0}

```

Listing 5.4: Example relevance-features map pair for *player-1*.

It is important to note that there are two limitations to this approach. The first relates to the fact that documents that are not clicked on are therefore not present in the query logs, so they will also not take part in the comparisons. If we take the previous example again and consider that for the query [ronaldo] the documents *player-1* and *player-131* are retrieved in the first and fourth positions respectively, our method will only consider the document in the first position to be more relevant than the one in the fourth position, but it will not consider that the last one would also be more relevant than the documents in the second and third positions that were not even clicked. The second limitation is that the training does not take into account the size of the difference in the number of clicks. In other words, whether a document receives significantly more clicks, as in the example, or just one additional click, our approach will classify it as more relevant without deeper consideration.

Since we were using the LIBLINEAR library [5], we decided to analyze the available options for the type of solver and test the most suitable ones on our training set through a 5-fold cross-validation. In the end, we chose the L2-regularised L2-loss support vector classification that solves the primal problem, as it was the one that obtained the best cross-validation accuracy with the set of all features we defined in the first iteration. For the other options, we used the default settings to reduce the risk of over-fitting.

The next step was to train the model with these configurations to obtain the feature weights. To do so, we first had to select the features to be included in each model and apply the training. In the next section, we will address the specific features employed in each model. For the training data, we used the query logs referring to clicks in the 30 days prior to training the models. We also separated them by domain, i.e., the logs for the Portuguese domain were only used to train the models for the Portuguese domain, and the logs for the Brazilian domain were only used to train the models relate to the Brazilian domain. This separation is essential to maintain the geographical context already applied in the zerozero search engine. Therefore, the weights of the same features in the same model version would be expected to vary from the Portuguese to the Brazilian version.

5.3 Solution Deployment

Having established the A/A test configuration, in which system A represented the current zerozero production engine, we replaced the second A variant with the systems that use LTR models to re-rank the results we trained. In this work, we considered three main concepts for assessing relevance: popularity, merit, and geographical location. Consequently, we built three models,

each focusing on a single concept, adding the respective new features to the ones that were already used in the zerozero system. The models have been designated as Pop, Mer, and Geo, respectively, in order to facilitate their referencing. Each of the systems was deployed receiving 10% of the user sessions, which was the maximum proposed by Manning et al. [21], and the re-rank query was applied to the top-1000 documents retrieved from the initial query made by the original zerozero system. It is essential to highlight that this re-ranking query is exclusively applied to the general index of the Solr server, which is used in the search bar and on the generic search page.

For the implementation of these systems, the deployment was carried out in phases, with the aim of conducting tests to verify the integrity of the process and the results. System Pop was deployed on the 12th of April 2024 using the popularity features, i.e., those concerning clicks. Five days later, on the 17th, we implemented system Mer with features related to merit that were available at that time. These included *is_glory*, *number_international_matches*, *is_retired*, and *is_deceased*. Finally, on the 23rd, system Geo was introduced with the three features related to geographical location. The specific features of the models used in each system and the associated weights are presented in Appendix A. The training data comprised query logs from 12 March to 11 April 2024.

The weights of the features correspond to their influence on the ranking function. As mentioned before, it was expected that the weights for the same features would differ from the Portuguese domain to the Brazilian domain since we applied a geographical split by training each model using only the data from the respective domain. On the other hand, in some cases, negative weights are used in the models, meaning that the inverse of the value of these features is considered to determine the order of the pairs of documents.

In the Pop model, some features have negative weights, which was unexpected since clicks can be used to assess relevance and not the opposite. Three of the features corresponding to clicks from two weeks ago have negative values, and the other (*clicks_2_weeks_ago_br*) has an insignificant weight. The feature *clicks_total* also has a small weight in the Portuguese model and is negative in the Brazilian one. The same happens with *clicks_last_week_language*. For the Mer model, we expected to see the features *is_retired* and *is_deceased* with negative weights, as retired and deceased players are less searched for. The biggest surprise was the *is_glory* feature, which has a small value in the Brazilian domain and a negative weight in the Portuguese domain. The Geo model considers *country_match* as the most relevant location match feature, followed by *current_team_country_match*. According to the trained model, the *all_teams_country_match* is the least influential of these three features in the ranking function and is even negative in the Brazilian domain.

5.4 Summary

In this section, we presented a conceptual overview of the Solr LTR module, which encompasses the search and training pipelines constituting a search system. The re-ranking of documents uses ML models as an essential component of the overall ranking process, enhancing the relevance

of the retrieved results presented to the user. Solr provides a detailed guide, accompanied by illustrative examples [34], outlining the configuration and execution of LTR in a real system. It emphasizes that the only thing that needs to be done outside Solr is to train the model. Furthermore, we have provided an overview of the main approaches to learning to rank, providing insight into their different methodologies and implications for ranking tasks in information retrieval.

We then described the implementation of LTR within the zerozero search system. Our initial step entailed establishing a test system, which proved to be a pivotal element in the subsequent stages of development. We then delineated the features to be utilized in the models we had defined. Subsequently, we described the process of feature engineering for the click-through data and the decrease date of an entity. Finally, we presented the model training process, providing a detailed overview of the most critical steps. We followed and adapted the example provided by Solr and used the LIBLINEAR library, following a pairwise approach. For the training data, we used the query logs provided by zerozero.

Finally, we developed three systems, each addressing a specific concept associated with relevance: popularity, sporting merit, and geographical location. For that, we took advantage of the A/B tests architecture, which provided a solid foundation for deployment. The training process yielded ranking functions with some negatively weighted features, such as clicks, which were not anticipated.

Chapter 6

Solution Evaluation

Following the implementation of the systems and a period of gathering results, the results were evaluated using the set of metrics explored in Chapter 3, which were also used in the analysis of the impact of Solr in the zerozero system. The most clicked result was still considered the correct answer for a query.

In addition to the previously described data processing steps, we implemented a further procedure specifically designed for this evaluation process. This involved the removal of all entries from the logs where the click position exceeded 500. This was done to prevent any potential negative impact on the results, since the selected metrics are relative to the click position. Although there are only a few cases, they significantly increase the ACP value, and upon closer examination, we can conclude that these cases do not correspond to navigational queries. One illustrative example is searching for the name of a club where the clicks are on the last available pages in results for players representing that club at the youth level. This indicates that the query is the name of a club but that the information need is a set of players. Alternatively, there may be no defined information need. Another example is the search for a specific calendar year (e.g., 1980), and there are clicks on multiple results of competition editions from that year. It is unreasonable to expect that all of these results will be highly ranked, and, therefore, the click position is significantly high. These types of searches present a challenge in fulfilling the search while still maintaining the most relevant results at the top of the ranking. Consequently, we have decided to remove them from our analysis. Furthermore, since these are uncommon occurrences, not all systems may have been subjected to these types of queries, which would result in a discrepancy in the evaluation results for only a few systems.

Later in this chapter, we will address the second iteration of the systems, in which we aim to improve the first version of the systems by applying the conclusions drawn from the evaluation process.

Table 6.1: Evaluation metrics from first iteration systems.

System	ACP	MRR	wMRR	S@1	S@2	S@3	S@4	S@5
A	2.382	0.789	0.772	0.679	0.820	0.881	0.914	0.931
Pop	2.274	0.789	0.772	0.676	0.823	0.883	0.914	0.931
Mer	2.389	0.772	0.756	0.655	0.802	0.868	0.903	0.925
Geo	2.372	0.785	0.769	0.675	0.815	0.878	0.910	0.928
A most frequent	–	0.921	0.866	0.871	0.947	0.964	0.972	0.978
Pop most frequent	–	0.909	0.848	0.848	0.932	0.965	0.990	0.990
Mer most frequent	–	0.880	0.826	0.806	0.910	0.945	0.955	0.977
Geo most frequent	–	0.903	0.857	0.842	0.932	0.952	0.968	0.984

6.1 First Version Systems Evaluation

Having implemented the systems, we conducted an evaluation process to assess their performance in the zerozero environment. Therefore, the query logs corresponding to the period when the three systems were in production - between the 23rd of April 2024, following the implementation of system Geo, and the 2nd of May - were analyzed.

As can be seen in Table 6.1 and Figure 6.1, which show the results of the systems for the metrics concerned, although systems Mer and Geo have worse results, system Pop is competitive compared to the original system (system A), achieving even better results in the ACP and differing by less than one hundredth in the other metrics.

An analysis of the tables about the distribution of clicks per position (Table 6.2) and the distribution of the correct answer per position (Table 6.3) reveals that the Pop system remains competitive with the original production engine. While the original system exhibited a higher click-through rate in the first position, system Pop demonstrated a higher percentage of clicks from the second position onwards. From position seven onwards, the situation is reversed, although the differences are less significant. This reflects system Pop's greater ability to place relevant results in the top six positions. The other systems are less effective, with the difference being more noticeable in the top positions. Considering the distribution of the correct answer per position, systems A and Pop exhibit comparable values, yet alternate in which one answers more queries depending on the position. It is noteworthy that the values in the column with the cumulative percentage correspond to the concept of the S@N metric (i.e., the cumulative value of position five and the S@5 metric correspond to the percentage of queries that are answered, showing only the first five results). Therefore, this table is only useful for analysis from the sixth click position onwards.

Given the similarity in the results of the evaluation of the systems, we proceeded to examine the results in greater detail. This involved preparing Table 6.4, which presents the distribution of clicks per entity and the respective ACP. This analysis reveals that players accomplished a better value. At the same time, the entities team, manager, and competition, which collectively represent

Table 6.2: Click position distribution from first iteration systems.

Rank.	System A		System Pop		System Mer		System Geo	
	Percen.	Cumul.	Percen.	Cumul.	Percen.	Cumul.	Percen.	Cumul.
1	69.06%	69.06%	68.68%	68.68%	65.29%	65.29%	67.47%	67.47%
2	12.60%	81.67%	13.28%	81.96%	14.52%	79.81%	13.21%	80.68%
3	5.43%	87.10%	5.59%	87.55%	6.10%	85.91%	5.76%	86.44%
4	3.03%	90.13%	3.02%	90.57%	3.34%	89.25%	3.36%	89.81%
5	2.04%	92.17%	1.79%	92.36%	2.21%	91.45%	2.14%	91.95%
6	1.70%	93.87%	1.68%	94.04%	1.79%	93.24%	1.61%	93.56%
7	1.20%	95.06%	0.97%	95.01%	1.29%	94.53%	0.97%	94.53%
8	0.95%	96.02%	0.87%	95.88%	0.98%	95.51%	0.96%	95.49%
9	0.78%	96.80%	0.80%	96.68%	0.90%	96.40%	0.89%	96.38%
10	0.62%	97.42%	0.72%	97.40%	0.68%	97.08%	0.70%	97.08%
11+	2.58%	100.00%	2.60%	100.00%	2.92%	100.00%	2.92%	100.00%

Table 6.3: Correct answer position distribution from first iteration systems.

Rank.	System A		System Pop		System Mer		System Geo	
	Percen.	Cumul.	Percen.	Cumul.	Percen.	Cumul.	Percen.	Cumul.
1	67.87%	67.87%	67.65%	67.65%	65.52%	65.52%	67.47%	67.47%
2	14.15%	82.02%	14.68%	82.33%	14.70%	80.22%	14.02%	81.50%
3	6.08%	88.10%	5.93%	88.26%	6.53%	86.75%	6.26%	87.76%
4	3.27%	91.37%	3.09%	91.35%	3.58%	90.33%	3.25%	91.01%
5	1.77%	93.14%	1.73%	93.09%	2.13%	92.46%	1.79%	92.80%
6	1.41%	94.55%	1.49%	94.58%	1.66%	94.12%	1.38%	94.19%
7	1.12%	95.67%	0.98%	95.55%	1.17%	95.29%	1.02%	95.20%
8	0.85%	96.52%	0.78%	96.33%	0.93%	96.22%	0.97%	96.18%
9	0.68%	97.20%	0.78%	97.11%	0.67%	96.89%	0.80%	96.98%
10	0.60%	97.81%	0.57%	97.68%	0.58%	97.48%	0.66%	97.64%
11+	2.19%	100.00%	2.32%	100.00%	2.52%	100.00%	2.36%	100.00%

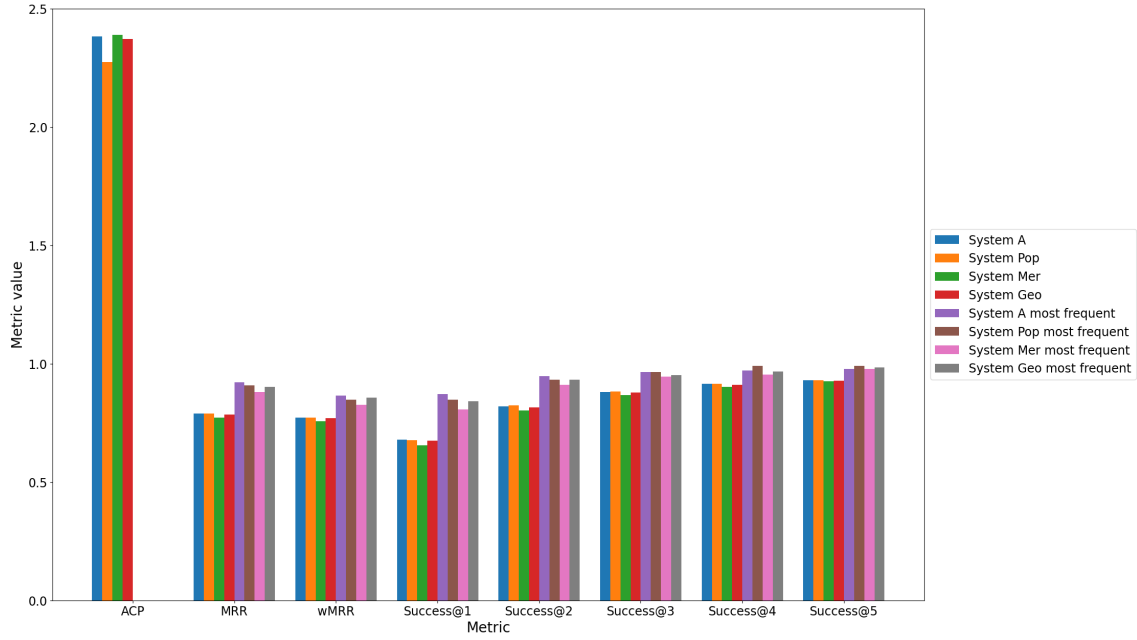


Figure 6.1: Evaluation metrics from first iteration systems.

approximately 60% of the searches, exhibited a higher ACP in the developed systems. This could be due to the fact that the training process uses the number of clicks as the relevance factor. The training process utilized data from a 30-day period during which, for a portion of that period, only clicks on player, competition, and edition results were recorded. This resulted in the machine learning model being biased towards the documents of these three types of entities. For instance, the player documents received more clicks over a longer period than the team documents. Furthermore, the features employed are subject to certain limitations. In model Pop, the number of clicks associated with each document presents the same issue, namely that the values of these attributes in the documents of players, competitions, and editions could be higher than in those of the other entities since the number of clicks has been counted for a more extended period. In model Mer, certain features are assigned exclusively to players (*is_glory* and *number_international_matches*). Also, in model Geo, the *current_team_country_match* and *all_teams_country_match* features can only have a value other than 0 in the player and manager documents. In addition, all models utilize the *boost_value* feature, which is exclusively assigned to players and teams.

6.2 System Improvements

Although the ACP for the players had improved with the introduction of the LTR models, it would be necessary to take greater account of the rest of the entities in order to obtain more satisfactory results for the system as a whole. To this end, we adapted the Mer system model by creating new merit features, which were not only aimed at players but also other entities. The *boost_value* feature corresponds to an attribute created by the zerozero team to deal with merit. This attribute assigns greater value to international players who have won specific trophies and to clubs and

Table 6.4: Clicks per entity distribution in the first iteration systems.

Entity	System A		System Pop		System Mer		System Geo	
	Percent.	ACP	Percent.	ACP	Percent.	ACP	Percent.	ACP
Team	52.15%	1.98	52.63%	2.01	53.72%	2.22	52.51%	2.11
Player	39.31%	2.81	39.38%	2.57	38.44%	2.61	39.50%	2.67
Manager	4.13%	2.19	3.91%	2.67	3.53%	2.43	3.72%	2.44
Competition	3.08%	2.27	3.20%	2.35	3.19%	2.34	3.20%	2.58
Edition	0.88%	8.51	0.59%	2.73	0.80%	3.43	0.76%	3.92
Stadium	0.23%	2.28	0.16%	1.63	0.13%	1.70	0.13%	2.08
Referee	0.12%	2.25	0.07%	4.14	0.07%	1.95	0.08%	2.28
Director	0.07%	2.03	0.04%	2.69	0.12%	2.16	0.07%	3.24
Agent	0.02%	1.35	0.01%	4.75	0.01%	1.00	0.02%	1.00

national teams that participate in the most competitive competitions. It also rewards those who have won national and continental titles as well as international and intercontinental titles. Each of these signals contributes to this attribute with a weight predefined by the zerozero team. However, we believe it would be advantageous to have a feature for each of these signals and let the LTR model define their weights according to data from the query logs. In order to achieve this, we have created features for the number of national (continental in the case of national teams) and international (intercontinental) titles that are associated with the documents of player, team, and manager entities. Furthermore, the number of awards has been incorporated as a feature that can be applied to players and managers. Additionally, the fact that a player is international was used in the *boost_value* field and we also considered it as a feature. Competitions and editions account for nearly 4% of searches and have associated levels. The *level* field refers to the level of the competition. For example, the UEFA Champions League level is higher than the UEFA Europa League. These fields also exist for teams and managers, with those for the senior level having a higher value. Additionally, the *ambito* attribute categorizes competitions and editions according to their scope, specifically whether they are intercontinental, continental, national, etc. The *level2* field is solely utilized to differentiate between main teams and B teams, as well as other non-main teams that may exist. We hypothesize that by utilizing these fields, the model will be able to more accurately identify the relevance of documents from all entities. Furthermore, we introduced the *is_recently_deceased* feature. All of these features were presented in Chapter 5 in Table 5.1 but were only used in the second iteration of the Mer model.

In the Pop model, no new features have been introduced. In fact, the *indice_pesq_language* feature, which was defined by the zerozero team as a way of dealing with document popularity, has been removed. Given that this model yielded the most satisfactory results in the initial iteration and that the weight of the *indice_pesq_language* feature is relatively insignificant, we elected

to retain only the features associated with clicks, entirely replacing the approach of the original zerozero system. Similarly, no additional features were introduced into the Geo model; instead, the *all_teams_country_match* feature was removed, given its minimal impact on the Portuguese domain model and its negative weight in the Brazilian one.

Finally, a further model incorporating signals from the three identified concepts, designated All, was defined and trained, incorporating popularity (all the features associated with clicks), merit (all the merit features used in the Mer version 2 model), and geographical location (location features from the Geo version 2 model, *country_match* and *current_team_country_match*) features. The model was trained exclusively in the Portuguese environment. However, it was also deployed in the Brazilian domain with the same weights as in the Portuguese, replacing the language features with the equivalent Brazilian features (for example, *clicks_pt* to *clicks_br*).

Having defined the features to be used, the models were trained using the same process but with more recent data from the 30 days before 9 May 2024. As previously described, using data from a period when clicks on all entities were recorded ensures a more reliable and consistent training process. On 9 May, the three previous models were replaced with their updated versions, and a new model, All, was introduced. The same proportion (10%) of user sessions were redirected to each system. Appendix B illustrates the features and respective weights employed in each model in this second iteration.

6.2.1 Model Analysis

An analysis of the feature weights computed by the LTR models reveals that, in model Pop, the same phenomenon observed in the previous iteration occurred, with some click features exhibiting negative values. This time, the *clicks_total* and the *clicks_2_weeks_ago_language* are negative in both domains. Furthermore, the *clicks_language* is negative only in the Portuguese and *clicks_last_week_language* only in the Brazilian. In model Mer, only *ambito* in the Portuguese domain and *is_glory* in the Brazilian had unexpected negative weights. Model Geo is composed of all positive weights, which is consistent with the features used. In the model All, *clicks_2_weeks_ago* and *total_awards* were the only unexpected feature weights being negative.

6.2.2 Evaluation

Following the implementation of the four systems from the second iteration, the data collection period commenced on 9 May and concluded on 18 May 2024. We then replicated the analysis process of the previous iteration.

Table 6.5 gathers the results for the selected metrics and shows that, although in the initial version, system Pop appeared to be the most competitive with system A, in the second iteration, all the systems with an LTR model demonstrated equal or superior performance to the original system across all metrics. System Mer achieved the best values in the ACP, MRR, wMRR, and S@1 metrics. In contrast, systems Geo and All exhibited the most notable performance in the other success metrics. When considering only the 200 most frequent queries, it was system Geo

Table 6.5: Evaluation metrics from second iteration systems.

System	ACP	MRR	wMRR	S@1	S@2	S@3	S@4	S@5
A	2.334	0.792	0.775	0.684	0.820	0.881	0.912	0.930
Pop	2.183*	0.795	0.780*	0.690	0.823	0.882	0.912	0.930
Mer	2.127*	0.799*	0.785*	0.697*	0.825	0.885	0.914	0.932
Geo	2.201	0.798*	0.783*	0.692	0.826	0.888*	0.919*	0.936*
All	2.157	0.797*	0.782*	0.690	0.829*	0.888*	0.918*	0.937*
A most frequent	–	0.905	0.869	0.850	0.927	0.955	0.960	0.966
Pop most frequent	–	0.907	0.869	0.850	0.938	0.947	0.966	0.975
Mer most frequent	–	0.889	0.858	0.818	0.920	0.952	0.971	0.978
Geo most frequent	–	0.914	0.875	0.860	0.940	0.962	0.971	0.987
All most frequent	–	0.893	0.849	0.820	0.930	0.959	0.981	0.984

Table 6.6: Cumulative click position distribution from second iteration systems.

Ranking	System A	System Pop	System Mer	System Geo	System All
1	69.89%	70.55%	71.08%	70.03%	69.91%
2	82.41%	82.98%	83.19%	82.67%	83.56%
3	87.95%	88.49%	88.74%	88.21%	88.92%
4	90.75%	91.29%	91.69%	91.08%	91.86%
5	92.53%	93.07%	93.32%	93.10%	93.60%
6	94.15%	94.45%	94.70%	94.62%	94.76%
7	95.27%	95.58%	95.86%	95.57%	95.82%
8	96.18%	96.29%	96.66%	96.39%	96.66%
9	96.94%	96.92%	97.32%	97.14%	97.32%
10	97.54%	97.54%	97.81%	97.70%	97.79%
11+	100.00%	100.00%	100.00%	100.00%	100.00%

Table 6.7: Cumulative correct answer position distribution from second iteration systems.

Ranking	System A	System Pop	System Mer	System Geo	System All
1	68.37%	69.02%	69.68%	69.15%	69.00%
2	82.03%	82.34%	82.49%	82.55%	82.92%
3	88.15%	88.17%	88.47%	88.79%	88.76%
4	91.17%	91.23%	91.44%	91.90%	91.85%
5	93.02%	93.01%	93.18%	93.62%	93.68%
6	94.51%	94.34%	94.71%	95.02%	94.90%
7	95.60%	95.43%	95.80%	95.98%	95.87%
8	96.53%	96.22%	96.63%	96.89%	96.68%
9	97.29%	96.96%	97.25%	97.58%	97.36%
10	97.92%	97.66%	97.80%	98.11%	97.88%
11+	100.00%	100.00%	100.00%	100.00%	100.00%

Table 6.8: Average click position per entity in the second iteration systems.

Entity	System A	System Pop	System Mer	System Geo	System All
Team	2.08	2.01	1.80	2.04	1.91
Player	2.58	2.33	2.43	2.36	2.43
Manager	2.45	2.30	2.28	2.01	2.21
Competition	2.27	2.24	2.59	2.44	2.02
Edition	3.93	4.54	3.97	3.80	2.92
Stadium	2.46	1.44	5.31	1.88	2.40
Referee	4.17	1.58	4.88	2.18	3.28
Director	2.48	3.12	2.26	3.83	1.78
Agent	1.94	1.29	1.20	1.25	1.29

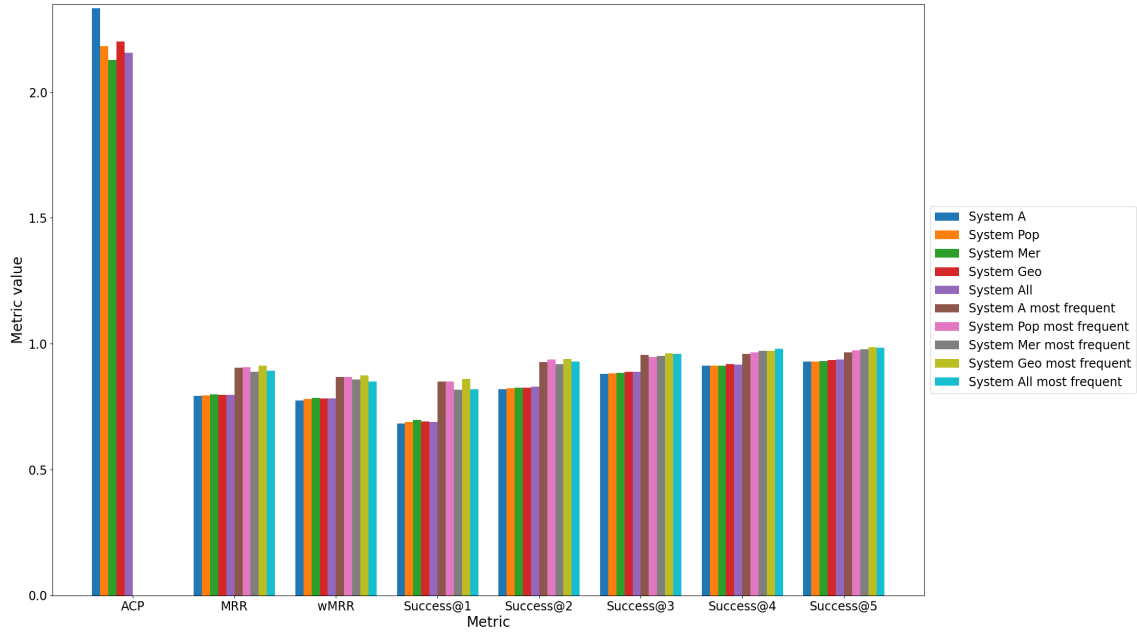


Figure 6.2: Evaluation metrics from second iteration systems.

that demonstrated the most favorable results for all metrics, except for S@4, where system All was the superior performer. Figure 6.2 complements the information presented in the table by providing a graphical representation of the metric values for all the systems.

An analysis of the cumulative distribution of clicks by result position, as shown in Table 6.6, reveals that system A consistently has the worst cumulative value in the first eight positions. Conversely, system All has the highest click-through rate for results in the first two, three, four, five, and six positions despite having a lower click-through rate in the first position than the other developed systems. With regard to the cumulative distribution of correct answers by position, as illustrated in Table 6.7, it is essential to recall that the cumulative value corresponds to the S@N metric. Upon analysis of the data from the sixth line onwards, it can be observed that system Geo consistently obtains the best value, while system Pop consistently obtains the lowest value.

Table 6.8 presents the ACP for each entity type in each system. It can be observed that for entities of the type team, which is the most frequently clicked on, system Mer achieved the best result with an average position of 1.80. This improvement reflects the fact that the features introduced in this model are not only aimed at players. This also demonstrated superior performance compared to System A with respect to managers, although it did not achieve the best result among all the systems. In terms of competitions, system Mer demonstrated the worst overall performance for this metric. Furthermore, it was found to be less effective than the original zerozero system for editions. Conversely, systems Geo and All demonstrated highly satisfactory outcomes for the most significant entities.

In contrast to the previous iteration, no system emerged as a clear leader in all the areas of analysis that we focused on. However, it is evident that the systems with an LTR model have outperformed the original zerozero system in terms of the values of these metrics. Among the

Table 6.9: Mann-Whitney U test *p-values*.

System	ACP	MRR	wMRR	S@1	S@2	S@3	S@4	S@5
Pop	0.006*	0.174	0.003*	0.117	0.354	0.934	0.803	0.976
Mer	1.77e-6*	0.004*	7.84e-6*	0.001*	0.167	0.256	0.284	0.457
Geo	0.366	0.035*	0.001*	0.060	0.125	0.025*	0.004*	0.008*
All	0.109	0.0497*	0.0004*	0.122	0.008*	0.032*	0.006*	0.003*
Pop most frequent	–	0.974	0.069	0.992	0.574	0.635	0.723	0.497
Mer most frequent	–	0.302	0.069	0.269	0.765	0.875	0.442	0.369
Geo most frequent	–	0.671	0.054	0.713	0.499	0.648	0.437	0.074
All most frequent	–	0.347	0.444	0.285	0.848	0.797	0.119	0.138

developed systems, Pop, which was the most competitive in the first iteration, is now the least competitive in most metrics, although the differences are not significant. To provide further insight, statistical tests were employed to enhance comprehension of the results and their statistical significance. The objective was to determine whether the enhancements observed in the developed systems relative to the original zerozero system were statistically significant.

To this end, we first conducted the Shapiro-Wilk normality test [31] to find out if the data follows a normal distribution. We adopted a *p-value* of 0.05, which is the standard value. The results demonstrated that, for all the metrics in all the systems, the *p-value* was much lower than this reference, indicating that none follow a normal distribution for these metrics. This led us to employ non-parametric tests to compare the distributions of the metrics between the different systems.

Consequently, we used the Mann-Whitney U test [20], a non-parametric statistical test designed to compare two independent samples. The null hypothesis (H_0) of this test states that the two samples are derived from the same distribution, indicating that there is no statistically significant difference between the distributions of the two variables being compared. The objective was to identify which combinations rejected the null hypothesis. Once more, the reference *p-value* of 0.05 was considered. Table 6.9 presents the results of the *p-value* obtained when comparing each of the developed systems with the A system for the respective metric. The results below the threshold value are marked with an asterisk (*). In Table 6.5, the asterisk marks the statistically significant values. The results indicate that:

- There is a statistically significant difference between systems A and Pop for the ACP and wMRR metrics.
- There is a statistically significant difference between systems A and Mer for the ACP, MRR, wMRR, and S@1 metrics.
- There is a statistically significant difference between systems A and Geo for the MRR, wMRR, S@3, S@4, and S@5 metrics.

- There is a statistically significant difference between systems A and All for the metrics MRR, wMRR, S@2, S@3, S@4, and S@5.

In the remaining metrics for these systems, it is not possible to conclude that there was, in fact, a difference compared to the original system. For the top 200 most frequent queries, none of the systems showed statistically significant differences in any of the metrics. Therefore, system A performs similarly to the developed systems in the context of the most common queries. The wMRR metric showed a statistically significant difference in all comparisons, indicating consistency in the performance differences of the developed systems concerning this metric. Moreover, the discrepancies observed in some of the S@N metrics indicate that the capacity of systems Geo and All to return relevant results in the top positions differs from that of system A. It can also be concluded that in the three systems where a statistically significant difference was observed for the MRR metric, the order of the results presented by the ranking algorithm differed significantly. Nevertheless, no system exhibited statistically significant differences in the values for all metrics. The closest systems to this outcome were Geo and All, with five and six metrics demonstrating statistically significant differences, respectively.

6.3 Summary

The results obtained about the performance of the first version of the systems showed that, for the metrics we used, one of the systems, system Pop, demonstrated a competitive performance against the original zerozero system. In contrast, the remaining systems obtained worse values in almost all the metrics. Upon a more detailed analysis, we observed that the implemented systems obtained a better ACP value than the original system in the player results but a worse value in the other three most clicked entities. Consequently, we concluded that this was due to the selected features favoring players.

We then developed a second version of the models with the objective of addressing the gaps that had been identified previously. In the Mer system, we also introduced features focused on teams, managers, competitions, and editions. In systems Pop and Geo, we removed some features with less significant weight. We also implemented a model with features from all the concepts mentioned above. The results showed that the four systems obtained better values than the original zerozero system for the metrics used. Systems Mer, Geo, and All were the ones that stood out the most, in contrast to what happened in the previous iteration, where Pop was the most notable system. Upon the analysis of clicks per entity, the ACP value for teams and managers was lower than in system A and still lower for players.

Given this overall improvement of the implemented systems compared to the original system, we employed statistical tests in order to understand the statistical significance of the results obtained. The results demonstrated statistically significant differences in the wMRR metric for all systems. Additionally, the results indicated the capacity of systems Geo and All to return relevant results in the top positions. Moreover, it can be concluded that systems Mer, Geo, and All

presented statistically significant differences in the ordering of results compared to the original system.

In conclusion, the evaluation process has demonstrated that systems with a learning to rank model can retrieve more relevant results in the top positions in the context of the zerozero search system.

Chapter 7

Conclusions

The main objective of this work was to analyze and improve the search system of the Portuguese sports network zerozero, using Information Retrieval (IR) techniques, specifically by incorporating multiple signals into the ranking algorithm, which is done manually in the original system.

We started by analyzing the search system of the most visited sports-related websites in the world. Regarding the search interface, we observed that most platforms had very limited interfaces, particularly in terms of filtering functionality and sorting results. However, the zerozero platform stood out for its coverage of entities, control features, and detailed information in the results, thereby providing a complete search experience. Furthermore, we analyzed the ranking of results on some of these websites to investigate the impact of sporting events on them. One of the websites (flashscore.com) showed justified changes in the results for a set of queries we defined.

Our analysis of the zerozero system allowed the identification of three concepts in the original ranking function: popularity, sporting merit of the entities, and geographical location. We also identified the problems resulting from manually incorporating these signals, namely the unsatisfactory results for some queries. The Solr framework was introduced into the zerozero system in 2023, and a comparative analysis of the results, extracted from the query logs, showed that there were no notable differences compared to the period before the integration. In order to evaluate the systems in question, we used metrics for ordered sets of documents, namely ACP, MRR, wMRR, and Success.

Learning to Rank (LTR) consists of integrating machine learning algorithms into document ranking functions. With the aim of developing an automatic ranking function, we integrated Solr's LTR module into the zerozero search system in order to return more relevant results in the top positions. To this end, we defined signals that could help discern the relevance of the results in the form of features, which were then used in the training of the LTR models and in the ranking functions produced. Some of these signals include implicit feedback techniques that have been studied, such as click information and the geographical location of the query. The training process was based on an example provided by Solr that uses the LIBLINEAR library [5] for a pairwise approach, and we used the zerozero query logs as training data. We developed four systems, three of which were based on one of the following signals - popularity, merit, geographic location - and

the other included all of them.

The integration of the solution into the zerozero system was achieved through the implementation of an A/B tests architecture for conducting tests, where a small part of the search traffic (10% of sessions) was redirected to each of the developed systems. This technique is effective for evaluating the impact of changes, as it allows for the comparison of variants of systems implemented simultaneously in a common environment. In this case, the Portuguese and Brazilian domains of the zerozero platform were the environment of our implementation.

The evaluation of the first implementation of the systems revealed the competitiveness of one of the systems (the one that included the popularity features) compared to the original one according to the selected metrics. Furthermore, the evaluation highlighted the need to adjust the selected features, particularly to achieve a more balanced relevance between the different types of entities with the highest search volume (players, teams, managers, and competitions). Consequently, in the second iteration, we changed the models by introducing more comprehensive features, which resulted in a substantial improvement in the results, especially in the MRR and wMRR metrics, compared to the original zerozero system. From this analysis, we can conclude that three of the systems presented statistically significant differences in the ordering of results, and two of them retrieved more relevant results in the top positions. The systems that exhibited the most notable performance were Mer (the system that includes the merit features), Geo (geographical location features), and All (features from all concepts). System Mer achieved improvements of 0.207 points in ACP, 0.7 percentage points (p.p.) in MRR, 1 p.p. in wMRR, and 1.3 p.p. in S@1.

We, therefore, conclude that the incorporation of multiple signals capable of assessing the relevance of results and their automatic weighting in the ranking algorithm has a superior capacity to produce more relevant results when compared to the original zerozero system, which included fewer signals and weighted them manually.

7.1 Main Contributions

This work had significant contributions to the search system of the zerozero network. The developed systems represented improvements in the quality of results and have already been deployed in production on their platform. We also implemented an architecture for A/B tests, allowing for future optimizations by directly comparing different system versions.

The conclusions of this study also have implications for the implementation and evolution of IR systems in specific contexts, in particular on sports platforms. We highlight three other main contributions. Firstly, we conducted a comprehensive analysis of the search interfaces of several sports platforms in terms of input, control, and informational functionalities. In addition, we emphasized that the zerozero interface proved to be more complete, which provides a basis for developing best practices and design standards for search interfaces on sports platforms.

Secondly, in addition to our study of techniques for obtaining implicit feedback on relevance, we also applied some of these techniques to the ranking function using features, something that

had not been explored in the sports domain. Through the processing of query logs, we incorporated these features into the documents to assess their relevance to a given query.

Thirdly and lastly, in addition to the satisfactory results obtained by our solution, we have also described in detail the process of building an algorithm for ranking results using LTR techniques. Moreover, we discussed its direct application in the Solr framework using the module provided, something that has been very little and incompletely explored.

7.2 Limitations

Throughout this document, we have identified the limitations found at each stage. The most significant limitation of this work was that there were few practical examples of the application of LTR, particularly in training and building models. Consequently, we had to use the example provided by Solr, which was not fully documented, and used a pairwise approach, which, in terms of performance, would fall short of a listwise approach [18].

Furthermore, we used click-through data in multiples features and as a relevance judgement in the model training process. In general, implicit feedback has limitations, such as being difficult to interpret and noisy. In addition, click-through data has its own limitations, such as trust and snippet biases, so our solution inherently shares these limitations. Also, in the training process, non-clicked documents are not included, and the size of the difference in the number of clicks is not considered, only whether an entity has more, the same, or fewer clicks. Another limitation was the fact that the sport the user is in at the time of the search was not stored in the query logs, making it impossible to use the *sports_match* feature to train the models. This represents the removal of a signal that was used in the original zerozero ranking function.

7.3 Future Work

We started this work by analyzing and comparing the search interface of some sports platforms according to input, control, and informational functionalities. This analysis can serve as a basis for future comparisons of any platform with a search functionality.

In addition, this work provides a basis for further developments in IR systems based on LTR solutions. The use of alternative approaches, such as listwise, the use of different signals or techniques to assess relevance, or the adoption of different algorithms to create models can be explored and implemented following the pipeline we have adopted in this work, taking advantage of A/B testing and the evaluation process. The training process we have detailed can also be adapted to overcome the identified limitations, namely by including the documents of non-clicked entities and by taking into account the size of the difference in the number of clicks.

We also proposed to the zerozero team that they store the sport modality in which the user is searching in the query logs. This will allow the training of models with this feature, which we could not exploit in our work.

References

- [1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [2] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 89–96, New York, NY, USA, 2005. Association for Computing Machinery.
- [3] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Trans. Inf. Syst.*, 30(1), mar 2012.
- [4] João Paulo Madureira Damas. Building a domain-specific search engine that explores football-related search patterns. Master's thesis, Faculdade de Engenharia da Universidade do Porto (FEUP), 2020.
- [5] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIB-LINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [6] Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. Beyond clicks: query reformulation as a predictor of search satisfaction. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 2019–2028, New York, NY, USA, 2013. Association for Computing Machinery.
- [7] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. A probabilistic method for inferring preferences from clicks. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 249–258, New York, NY, USA, 2011. Association for Computing Machinery.
- [8] Jeff Huang and Efthimis N. Efthimiadis. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 77–86, New York, NY, USA, 2009. Association for Computing Machinery.
- [9] Jim Jansen, Danielle Booth, and Amanda Spink. Determining the informational, navigational, and transactional intent of web queries. *Information Processing & Management*, 44:1251–1266, 05 2008.
- [10] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 133–142, New York, NY, USA, 2002. Association for Computing Machinery.

- [11] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, page 154–161, New York, NY, USA, 2005. Association for Computing Machinery.
- [12] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2):7–es, apr 2007.
- [13] Christopher Jones and Ross Purves. Geographical information retrieval. *International Journal of Geographical Information Science*, 22:219–228, 03 2008.
- [14] Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. On application of learning to rank for e-commerce search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 475–484, New York, NY, USA, 2017. Association for Computing Machinery.
- [15] Diane Kelly and Nicholas J. Belkin. Display time as implicit feedback: understanding task effects. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 377–384, New York, NY, USA, 2004. Association for Computing Machinery.
- [16] Ron Kohavi and Roger Longbotham. *Online Controlled Experiments and A/B Testing*, pages 922–929. Springer US, 01 2017.
- [17] Apostolos Kritikopoulos and Martha Sideri. The compass filter: search engine result personalization using web communities. In *Proceedings of the 2003 International Conference on Intelligent Techniques for Web Personalization*, ITWP'03, page 229–240, Berlin, Heidelberg, 2003. Springer-Verlag.
- [18] Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, mar 2009.
- [19] Yiqun Liu, Yupeng Fu, Min Zhang, Shaoping Ma, and Liyun Ru. Automatic search engine performance evaluation with click-through data analysis. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 1133–1134, New York, NY, USA, 2007. Association for Computing Machinery.
- [20] Henry B. Mann and Donald R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50 – 60, 1947.
- [21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [22] David Massimo and Francesco Ricci. Popularity, novelty and relevance in point of interest recommendation: an experimental analysis. *Information Technology & Tourism*, 23:1–36, 12 2021.
- [23] João Miguel Nascimento Monteiro. Learning to Rank Retrieval Results for a Geographically Constrained Search Queries. Master's thesis, Instituto Superior Técnico, June 2012.

- [24] Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, page 272–281, Berlin, Heidelberg, 1994. Springer-Verlag.
- [25] Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana, Elias B. Khalil, and Deepak Turaga. Learning feature engineering for classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 2529–2535. AAAI Press, 2017.
- [26] Michael Nilsson, Diego Ceccarelli, and Joshua Pantony. Learning to rank for solr. <https://pt.slideshare.net/lucidworks/learning-to-rank-in-solr-presented-by-michael-nilsson-diego-ceccarelli-bloomberg-lp>, 2015. SlideShare, [Online; accessed January 2024].
- [27] Filip Radlinski and Nick Craswell. Optimized interleaving for online retrieval evaluation. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, page 245–254, New York, NY, USA, 2013. Association for Computing Machinery.
- [28] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, page 43–52, New York, NY, USA, 2008. Association for Computing Machinery.
- [29] World Population Review. Most popular sport by country 2024. <https://worldpopulationreview.com/country-rankings/most-popular-sport-by-country>, 2024. [Online; accessed June 2024].
- [30] Mark Sanderson and Janet Kohler. Analyzing geographic queries. In *SIGIR Workshop on Geographic Information Retrieval*, 09 2004.
- [31] Samuel S. Shapiro and Martin B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.
- [32] Si Shen, Botao Hu, Weizhu Chen, and Qiang Yang. Personalized click model through collaborative filtering. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, page 323–332, New York, NY, USA, 2012. Association for Computing Machinery.
- [33] Solr. LTR training example. <https://github.com/apache/lucene-solr/tree/releases/lucene-solr/8.7.0/solr/contrib/ltr/example>, 2017. [Online; accessed April 2024].
- [34] Solr. Learning to rank. https://solr.apache.org/guide/8_7/learning-to-rank.html, 2020. [Online; accessed January 2024].
- [35] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, page 449–456, New York, NY, USA, 2005. Association for Computing Machinery.

- [36] Walter Underwood. Measuring search relevance with MRR. <https://observer.wunderwood.org/2016/09/12/measuring-search-relevance-with-mrr>, 2016. [Online; accessed February 2024].
- [37] Max L. Wilson. *Search User Interface Design*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2011.
- [38] Qiang Wu, Christopher Burges, Krysta Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Inf. Retr.*, 13:254–270, 06 2010.
- [39] Yang Wu, Masayuki Mukunoki, Takuya Funatomi, Michihiko Minoh, and Shihong Lao. Optimizing mean reciprocal rank for person re-identification. In *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 408–413, 2011.
- [40] Jun Zhai and Kaitao Zhou. Semantic retrieval for sports information based on ontology and sparql. In *2010 International Conference of Information Science and Management Engineering*, volume 1, pages 395–398, 2010.
- [41] Huaiyu Zhu, Sriram Raghavan, Shivakumar Vaithyanathan, and Alexander Löser. Navigating the intranet with high precision. In *Proceedings of the 16th International Conference on World Wide Web, WWW '07*, page 491–500, New York, NY, USA, 2007. Association for Computing Machinery.

Appendix A

Search User Interface Features Comparison

Table A.1: Search User Interface Input features in sports-related websites.

Website	Search box	Query typing	Query submission	Query text
espn.com	Icon	Nothing	Page with results	Visible
cricbuzz.com	Icon	10 suggestions	1st result page	Visible
espncricinfo.com	Icon	Nothing	Page with results	Visible
mlb.com	Icon	8 suggestions	Page with results	Visible
flashscore.com	Icon	20 suggestions	Not allowed	Visible
caughtoffside.com	Icon	Nothing	Page with results	Visible
livescore.com	Icon	30 suggestions	Not allowed	Visible
nfl.com	Hidden on menu	Nothing	Page with results	Visible
lequipe.fr	Hidden on menu	Text suggestions	Page with results	Visible
sportskeeda.com	Icon	Text suggestions	Page with results	Not visible
sports.yahoo.com	Visible	12 suggestions	Page with results	Visible
zerozero.pt (old)	Visible	10 suggestions	1st result page	Visible
zerozero.pt (new)	Visible	10 suggestions	Page with results	Visible

Table A.2: Search User Interface Control features in sports-related websites.

Website	Filters	Ordering
espn.com	By entity (teams, players, news)	No
cricbuzz.com	By entity (teams, news, series, venues)	No
espncricinfo.com	By entity (players, matches, news, videos)	No
mlb.com	Filter by news/videos	By date and relevance
flashscore.com	By sports	No
caughtoffside.com	No filters	No
livescore.com	By entity (regions, teams, competitions)	No
nfl.com	Active/retired	No
lequipe.fr	By sports	By date and relevance
sportskeeda.com	By entity (teams, tournaments, sports, players)	By date and relevance
sports.yahoo.com	Filter by images/videos/news	No
zerozero.pt (old)	First by entity, then specific filters	By relevance and alphabetical order
zerozero.pt (new)	First by entity, then specific filters	By relevance and alphabetical order

Table A.3: Search User Interface Informational features in sports-related websites.

Website	Pagination	Volume of results
espn.com	5 per entity	No
cricbuzz.com	5 per entity	No
espncricinfo.com	5 per entity	Yes
mlb.com	10 per page	Yes
flashscore.com	No	No
caughtoffside.com	10 per page	Yes
livescore.com	No	No
nfl.com	Infinite scroll	Yes
lequipe.fr	Infinite scroll	Yes
sportskeeda.com	10 per page	Yes
sports.yahoo.com	7 per page	Yes
zerozero.pt (old)	10 per page	Yes
zerozero.pt (new)	20 per page	Yes

Appendix B

Learning to Rank Models Used in the First Iteration

```
1 Pop model for Portuguese domain
2 {
3     "original_score":0.03949292619432868,
4     "indice_pesq_pt":9.196070209817321E-6,
5     "boost_value":0.005995384371146846,
6     "edition_check":0.6778796507888989,
7     "clicks_total":9.513577119123694E-5,
8     "clicks_pt":1.2762561609653314E-4,
9     "clicks_last_week_all":0.0011389528054647404,
10    "clicks_2_weeks_ago_all":-5.531901762740259E-4,
11    "clicks_last_week_pt":8.491672759771325E-4,
12    "clicks_2_weeks_ago_pt":-5.343706482068831E-4
13 }
14
15 Pop model for Brazilian domain
16 {
17     "original_score":0.02294308540286844,
18     "indice_pesq_br":2.9423752069232678E-5,
19     "boost_value":0.007875045889416803,
20     "edition_check":0.5510378136430166,
21     "clicks_total":-4.2357965770620996E-4,
22     "clicks_br":0.0040947628774548476,
23     "clicks_last_week_all":0.002319396467198114,
24     "clicks_2_weeks_ago_all":-0.0019370995141555197,
25     "clicks_last_week_br":-0.004826058938809012,
26     "clicks_2_weeks_ago_br":2.3429546477151167E-4
27 }
```

Listing B.1: Model Pop features and weights.

```

1 Mer model for Portuguese domain
2 {
3     "original_score":0.025469009128785317,
4     "indice_pesq_pt":3.40713523618487e-05,
5     "boost_value":0.006109655027279283,
6     "edition_check":0.667151882572916,
7     "is_glory":-0.0007652726873223852,
8     "number_international_matches":0.0034958249645979994,
9     "is_retired":-0.18169715324393043,
10    "is_deceased":-0.2192921233821423
11 }
12
13 Mer model for Brazilian domain
14 {
15     "original_score":0.02279917174511021,
16     "indice_pesq_br":7.238680663087223e-05,
17     "boost_value":0.007194256925164731,
18     "edition_check":0.5885739779414314,
19     "is_glory":0.000977833755108434,
20     "number_international_matches":0.00525533777397163,
21     "is_retired":-0.24538820994067911,
22     "is_deceased":-0.44669381579236866
23 }

```

Listing B.2: Model Mer features and weights.

```

1 Geo model for Portuguese domain
2 {
3     "original_score":0.02553199482390007,
4     "indice_pesq_pt":2.768542501130771E-5,
5     "boost_value":0.0072299600816608444,
6     "edition_check":0.6748922335098385,
7     "country_match":0.2722827811894627,
8     "current_team_country_match":0.07663341281126808,
9     "all_teams_country_match":0.0012780623530900819
10 }
11
12 Geo model for Brazilian domain
13 {
14     "original_score":0.021699024930665008,
15     "indice_pesq_br":6.801303714298702E-5,
16     "boost_value":0.009081341034705422,
17     "edition_check":0.5914532897627387,
18     "country_match":0.4185171757818522,
19     "current_team_country_match":0.2702337943698044,
20     "all_teams_country_match":-0.04450261921176278
21 }

```

Listing B.3: Model Geo features and weights.

Appendix C

Learning to Rank Models Used in the Second Iteration

```
1 Pop version 2 model for Portuguese domain
2 {
3     "original_score":0.02720286712469877,
4     "boost_value":0.006001399016290134,
5     "edition_check":0.5177103146378905,
6     "clicks_total":-8.290369294824715e-05,
7     "clicks_pt":-4.859391739230746e-05,
8     "clicks_last_week_all":0.0009456127019633271,
9     "clicks_2_weeks_ago_all":0.0002767289088184554,
10    "clicks_last_week_pt":0.0012944410185025156,
11    "clicks_2_weeks_ago_pt":-0.0001339112407338619
12 }
13
14 Pop version 2 model for Brazilian domain
15 {
16     "original_score":0.024436727756267932,
17     "boost_value":0.007754482680822878,
18     "edition_check":0.3471285107825534,
19     "clicks_total":-0.0006383097400968509,
20     "clicks_br":0.002425778213241583,
21     "clicks_last_week_all":0.0019736996245455374,
22     "clicks_2_weeks_ago_all":0.0011596068695358028,
23     "clicks_last_week_br":-0.003847314249220885,
24     "clicks_2_weeks_ago_br":-0.0032868932892798674,
25 }
```

Listing C.1: Model Pop version 2 features and weights.

```
1 Mer version 2 model for Portuguese domain
```

```

2  {
3      "original_score":0.021675627889265026,
4      "indice_pesq_pt":2.6557009604000073e-05,
5      "edition_check":0.5379419124746097,
6      "is_glory":0.13077023576838673,
7      "is_international":0.3248449409007448,
8      "number_international_matches":0.0003195015334848519,
9      "is_retired":-0.2020678740690713,
10     "is_deceased":-0.23122156471295666,
11     "is_recently_deceased":0.3492687903373676,
12     "total_international_titles":0.009761498465363056,
13     "total_national_titles":0.005521355800908933,
14     "total_awards":0.0011037356597750686,
15     "level":0.253950205184254,
16     "level2":0.3782772137295696,
17     "ambito":-0.023815937018609876
18 }
19
20 Mer version 2 model for Brazilian domain
21 {
22     "original_score":0.01848461047462273,
23     "indice_pesq_br":4.852997124403642e-05,
24     "edition_check":0.053211416965965784,
25     "is_glory":-0.07802404676278527,
26     "is_international":0.4293051109309227,
27     "number_international_matches":0.0005103631401351975,
28     "is_retired":-0.20879185986181314,
29     "is_deceased":-0.25267709558555246,
30     "is_recently_deceased":0.315945192575493,
31     "total_international_titles":0.035947330731844966,
32     "total_national_titles":0.010069403030477894,
33     "total_awards":0.001563655174191876,
34     "level":0.20898333598674354,
35     "level2":0.49823522723423824,
36     "ambito":0.1797300232740337
37 }

```

Listing C.2: Model Mer version 2 features and weights.

```

1  Geo version 2 model for Portuguese domain
2  {
3      "original_score":0.027308606642458193,
4      "indice_pesq_pt":2.8154288027009885e-05,
5      "boost_value":0.006661379889460027,
6      "edition_check":0.5503908113301745,
7      "country_match":0.32773300723160437,
8      "current_team_country_match":0.06251556707599684
9  }

```

```

10
11 Geo version 2 model for Brazilian domain
12 {
13     "original_score":0.024068507447933735,
14     "indice_pesq_br":5.697110416954091e-05,
15     "boost_value":0.008591657947790229,
16     "edition_check":0.37236832781499196,
17     "country_match":0.30290254165983627,
18     "current_team_country_match":0.23308642803776283
19 }

```

Listing C.3: Model Geo version 2 features and weights.

```

1 All model for Portuguese domain
2 {
3     "original_score":0.02299227187968316,
4     "edition_check":0.5453049997004102,
5     "is_glory":0.3159664133270564,
6     "is_international":0.399161396489088,
7     "number_international_matches":0.0003786395002554039,
8     "is_retired":-0.20118797370837568,
9     "is_deceased":-0.26264878327197644,
10    "is_recently_deceased":0.6983819921775215,
11    "total_international_titles":0.007238406932800777,
12    "total_national_titles":0.003635253746630557,
13    "total_awards":-0.01858853612253959,
14    "level":0.23928827786361548,
15    "level2":0.3479975687927479,
16    "ambito":0.40603954710197755,
17    "clicks_total":4.1020602719757035e-05,
18    "clicks_pt":0.0017473343719632577,
19    "clicks_last_week_all":0.006410173424601789,
20    "clicks_2_weeks_ago_all":0.00030803277301152097,
21    "clicks_last_week_pt":0.0004389506162859198,
22    "clicks_2_weeks_ago_pt":-0.00024491273064265317,
23    "country_match":0.1166546128934546,
24    "current_team_country_match":-0.009584166639676345
25 }
26
27 All model for Brazilian domain
28 {
29     "original_score":0.02299227187968316,
30     "edition_check":0.5453049997004102,
31     "is_glory":0.3159664133270564,
32     "is_international":0.399161396489088,
33     "number_international_matches":0.0003786395002554039,
34     "is_retired":-0.20118797370837568,
35     "is_deceased":-0.26264878327197644,

```

```
36     "is_recently_deceased":0.6983819921775215,  
37     "total_international_titles":0.007238406932800777,  
38     "total_national_titles":0.003635253746630557,  
39     "total_awards_l":-0.01858853612253959,  
40     "level":0.23928827786361548,  
41     "level2":0.3479975687927479,  
42     "ambito":0.40603954710197755,  
43     "clicks_total":4.1020602719757035e-05,  
44     "clicks_br":0.0017473343719632577,  
45     "clicks_last_week_all":0.006410173424601789,  
46     "clicks_2_weeks_ago_all":0.00030803277301152097,  
47     "clicks_last_week_br":0.0004389506162859198,  
48     "clicks_2_weeks_ago_br":-0.00024491273064265317,  
49     "country_match":0.1166546128934546,  
50     "current_team_country_match":-0.009584166639676345  
51 }
```

Listing C.4: Model All features and weights.