NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

BSc THESIS

GreekQA: A Crowdsourcing Platform and its Use for Creating a Greek Question Answering Dataset

Efstathios I. Siatras

Supervisors: **Manolis Koubarakis,** Professor
Despina - Athanasia Pantazi, PhD Candidate

ATHENS

October 2022

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

GreekQA: Πλατφόρμα Πληθοπορισμού και η Χρήση της για την Δημιουργία Ελληνικού Συνόλου Δεδομένων Ερωτήσεων και Απαντήσεων

Ευστάθιος Ι. Σιάτρας

Επιβλέποντες: **Μανόλης Κουμπαράκης,** Καθηγητής
Δέσποινα – Αθανασία Πανταζή, Υποψήφια Διδάκτωρ

ΑΘΗΝΑ

Οκτώβριος 2022

BSc THESIS

GreekQA: A Crowdsourcing Platform and its Use for Creating a Greek Question Answering Dataset

Efstathios I. Siatras

S.N.: 1115201600152

SUPERVISORS: **Manolis Koubarakis,** Professor
Despina - Athanasia Pantazi, PhD Candidate

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

GreekQA: Πλατφόρμα Πληθοπορισμού και η Χρήση της για την Δημιουργία Ελληνικού Συνόλου Δεδομένων Ερωτήσεων και Απαντήσεων

Ευστάθιος Ι. Σιάτρας

Α.Μ.: 1115201600152

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Μανόλης Κουμπαράκης,** Καθηγητής
Δέσποινα – Αθανασία Πανταζή, Υποψήφια Διδάκτωρ

ABSTRACT

Teaching machines to comprehend, process, and produce human language has been a perpetual challenge since the first decades of electronic digital programmable computers. In modern times, the progress made in the research area of Natural Language Processing is present in everyday life and facilitates people with an expanding set of conveniences. This field has once again flourished with the recent arrival of increasingly sophisticated and flexible language models. These state-of-the-art models have tackled a plethora of Natural Language Processing tasks bringing significant performance gains. Machine reading comprehension has been one of the cornerstone tasks that benefited from these recent advances. This challenging task requires machines to read a passage of text and answer questions based on the context. Besides the structure of the models, reading comprehension datasets have also played a decisive role in bringing successful results. Motivated by this trend in reading comprehension task, an increasing number of question answering datasets have appeared in English and a specific group of other languages. Regarding the Greek language, there has been no progress on native question answering datasets other than automatically translated ones from other languages.

In light of the above, we present the Greek Question Answering (GreekQA) dataset, a Greek reading comprehension dataset based on Wikipedia articles. GreekQA1.0 dataset consists of 1,000+ questions posed by crowdworkers on curated passages from a set of Wikipedia articles in Greek. For the development of the GreekQA dataset, we also introduce the namesake GreekQA Crowdsourcing Annotation Platform, a web application specifically designed and implemented for crowdsourcing the collection of question and answer pairs for this dataset. We analyze the requirements and the selected technologies of the GreekQA crowdsourcing platform, describe the structure of the implementation, and demonstrate the platform. We describe the procedure of curating passages and the defined guidelines of collecting question and answer pairs. In order to understand the properties of the GreekQA1.0, we analyze the questions and answers as well as the reasoning required to answer the questions based on the corresponding passage. Finally, we evaluate the Human Performance as a baseline for future experimental evaluation of language models using this dataset.

SUBJECT AREA: Computation and Language, Natural Language Processing, Machine Learning, Web Development

KEYWORDS: Machine Reading Comprehension, Question Answering, Dataset Collection, Data Annotation, Crowdsourcing Platform

ΠΕΡΙΛΗΨΗ

Η εκμάθηση των μηχανών να κατανοούν, να επεξεργάζονται και να παράγουν ανθρώπινη γλώσσα αποτελεί μια διαρκή πρόκληση από τις πρώτες δεκαετίες των ηλεκτρονικών ψηφιακών προγραμματιζόμενων υπολογιστών. Στη σύγχρονη εποχή, η πρόοδος που έχει σημειωθεί στον ερευνητικό τομέα της Επεξεργασίας Φυσικής Γλώσσας είναι παρούσα στην καθημερινή ζωή και διευκολύνει τους ανθρώπους με ένα αυξανόμενο σύνολο ανέσεων. Αυτός ο τομέας για άλλη μια φορά άνθησε με την πρόσφατη άφιξη ολοένα και πιο εξελιγμένων και ευέλικτων γλωσσικών μοντέλων. Αυτά τα μοντέλα τελευταίας τεχνολογίας αντιμετωπίζουν μια πληθώρα εργασιών Επεξεργασίας Φυσικής Γλώσσας αυξάνοντας την απόδοση. Η Αναγνωστική Κατανόηση είναι μια από τις βασικές εργασίες που επωφελήθηκαν από αυτές τις πρόσφατες εξελίξεις. Αυτή η δύσκολη εργασία απαιτεί από τις μηχανές να διαβάζουν ένα απόσπασμα κειμένου και να απαντούν σε ερωτήσεις με βάση το περιεχόμενο. Εκτός από τη δομή αυτών των μοντέλων, τα σύνολα δεδομένων κατανόησης ανάγνωσης έχουν διαδραματίσει αποφασιστικό ρόλο στην επίτευξη επιτυχημένων αποτελεσμάτων. Έχοντας ως κίνητρο αυτή την τάση στην Αναγνωστική Κατανόηση, ολοένα και περισσότερα συνόλα δεδομένων ερωτήσεων και απαντήσεων έχουν εμφανιστεί στα αγγλικά και ένα συγκεκριμένο σύνολο άλλων γλωσσών. Όσον αφορά την ελληνική γλώσσα, δεν έχει σημειωθεί κάποια πρόοδος σε εγγενή σύνολα δεδομένων ερωτήσεων και απαντήσεων, πέρα από αυτόματα μεταφρασμένα σύνολα από άλλες γλώσσες.

Υπό το φως των παραπάνω, παρουσιάζουμε το Ελληνικό Σύνολο Δεδομένων Ερωτήσεων και Απαντήσεων GreekQA, ένα σύνολο δεδομένων Αναγνωστικής Κατανόησης στα ελληνικά το οποίο βασίζεται σε άρθρα της Wikipedia. Το σύνολο δεδομένων GreekQA1.0 αποτελείται από 1.000+ ερωτήσεις που τέθηκαν από εθελοντές σε επιμελημένα αποσπάσματα από ένα σύνολο άρθρων της Wikipedia στα ελληνικά. Για την ανάπτυξη του συνόλου δεδομένων GreekQA, εισάγουμε επίσης την ομώνυμη Πλατφόρμα Πληθοπορισμού και Επισημείωσης Δεδομένων GreekQA, μια διαδικτυακή εφαρμογή ειδικά σχεδιασμένη και υλοποιημένη για τον πληθοπορισμό της συλλογής ζευγών ερωτήσεων και απαντήσεων για αυτό το σύνολο δεδομένων. Αναλύουμε τις απαιτήσεις και τις επιλεγμένες τεχνολογίες της πλατφόρμας, περιγράφουμε την δομή της υλοποίησης και παρουσιάζουμε την πλαφόρμα. Έπειτα, περιγράφουμε τη διαδικασία συλλογής και επιμέλειας αποσπασμάτων κειμένου και τις καθορισμένες κατευθυντήριες γραμμές για τη συλλογή ζευγών ερωτήσεων και απαντήσεων. Προκειμένου να κατανοήσουμε τις ιδιότητες του συνόλου δεδομένων GreekQA1.0, αναλύουμε την ποικιλομορφία στις ερωτήσεις και τις απαντήσεις καθώς και το σκεπτικό που απαιτείται για να απαντηθούν οι ερωτήσεις με βάση το αντίστοιχο απόσπασμα. Τέλος, αξιολογούμε την Ανθρώπινη Απόδοση ως βάση για μελλοντική πειραματική αξιολόγηση γλωσσικών μοντέλων που χρησιμοποιούν αυτό το σύνολο δεδομένων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Υπολογιστική Γλωσσολογία, Επεξεργασία Φυσικής Γλώσσας, Μηχανική Μάθηση, Ανάπτυξη Ιστού

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μηχανική Αναγνωστική Κατανόηση, Απάντηση σε Ερωτήσεις, Συλλογή Συνόλου Δεδομένων, Επισημείωση Δεδομένων, Πλατφόρμα Πληθοπορισμού

To my family.

ACKNOWLEDGEMENTS

Throughout the writing of this thesis, I have received a great deal of support and assistance. I would primarily like to thank my supervisors, Professor Manolis Koubarakis and PhD Candidate Despina-Athanasia Pantazi for their guidance and support throughout the completion of the project. In addition, I would like to acknowledge all the volunteers who contributed and assisted with the primary collection of the GreekQA1.0 dataset.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This thesis was documented to meet the graduation requirements of the undergraduate program at the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens. The research, development, and writing of this thesis were conducted from January 2022 to October 2022 under the coordination of the AI Team, part of the Management of Data, Information, and Knowledge Group (MaDgIK). During the development of this assignment, it was essential to be involved with a diverse set of fields, such as Web Development, Data Collection, Handling, and Management, as well as Natural Language Processing, focused on Data Analysis, Language Modeling, and the Machine Reading Comprehension task.

1. INTRODUCTION

1.1 Motivation

Recently, sophisticated language models based on Deep Learning (DL) architectures have tackled many Natural Language Processing (NLP) tasks [1] bringing significant performance gains. Reading Comprehension (RC) [2], the task of reading a passage of text and answering relevant questions, has been one of the cornerstone tasks that benefited from these recent advances. However, this progress in reading comprehension was based on more than just the structure of these models. Large-scale Question Answering (QA) datasets of high quality have played a decisive role in achieving these results [3]. On the other hand, creating such large QA datasets is a costly process posing significant difficulties, considering the fact that these datasets contain annotated passages of human-generated question and answer pairs.

The majority of reading comprehension datasets are in English, while a lack of native QA datasets in other languages is also noticed. Therefore, many researchers have utilized annotated datasets in English with automatic translation to their language of interest. This method is particularly useful in developing and evaluating language models in the scarcity of native-language datasets. However, the increasing demand for properly annotated language datasets in other languages has led to the research trend of creating new native-language QA datasets in other languages besides English. Some of these remarkable efforts involve languages such as French [4] and Korean [5], part of the related work we will discuss later. These datasets have brought significant advances in language-specific QA tasks. Consequently, the absence of a Greek Question Answering dataset has been our primary motivation in order to assist the further development and evaluation of QA models handling the Greek language.

1.2 Objective

The main objective of this thesis is the creation of a Greek QA dataset following the proposed methodologies and standards of SQuAD [6] and other related works with a similar objective. Therefore, this dataset is aimed to be based on a set of articles from Greek Wikipedia. For the development of the GreekQA dataset, we also aim for the development of a Crowdsourcing Annotation Platform, a web application specifically designed and implemented for crowdsourcing the collection of question and answer pairs. After creating the first version of this dataset, an essential aim is to understand the properties of this dataset, analyzing the questions and answers as well as the reasoning required to answer the questions based on the corresponding passage. Finally, our last goal is to evaluate the Human Performance as a baseline for future experimental evaluation of monolingual and multilingual language models using this dataset.

1.3 Overview

The chapters of this thesis are concerned as follows. This current chapter was an introduction to the motivation, the objective, and the aims of this thesis. Chapter 2 discusses background and various concepts of the related major fields, followed by chapter 3, which

focuses on related works and published matter in the field of Reading Comprehension datasets. Then, in chapter 4 we present the GreekQA Crowdsourcing Annotation Platform, a web application which was developed and extensively used for the collection of questions and answers. In chapter 5 we discuss the dataset collection procedure of collecting and curating passages based on Greek Wikipedia articles along with the prementioned process of collecting questions and answer pairs on the collected passages. Next, in chapter 6 we present the structure of the GreekQA1.0 dataset, analyze the development set thoroughly and evaluate the Human Performance. Finally, in chapter 7 we present our conclusions and discuss future work regarding the experimental evaluation of language models and the extension of the GreekQA dataset.

2. BACKGROUND

The chapter below briefly discusses historical background and concepts from the two major fields relevant to this thesis, computational linguistics and web application development. Computational linguistics [7] is a field concerned with the scientific modeling of natural language for computational systems. The primary objective of computational linguistics is to comprehend, process, and produce human language using computers. This field can be diverged into the two major subfields of theoretical and applied computational linguistics. For the purpose of this thesis, we focus on the latter area, applied computational linguistics. Later, we will briefly discuss the combinational use of categorized technologies to build an end-to-end modern web application.

2.1 Natural Language Processing

Natural Language Processing (NLP) [8, 9] is the field that applies computational linguistics algorithms and methodologies to read, understand, interpret, and generate various forms of natural language. In a broad definition, any computational technique which utilizes human language may be part of this research area.

Historically, the techniques used and the problems faced within the field of NLP varied dramatically. From the early days of the 1950s till the 1980s, the vast approaches in NLP had been symbolic-based, using explicitly defined rules, schemes, and algorithms. By the end of that period, symbolic NLP [10] gradually managed to deal with various tasks of great significance, such as parsing and knowledge extraction. Concurrently, the importance of computational resources led researchers to attempt to generalize NLP problems and reconsider other approaches [10]. Hence, a previously complementary methodology, statistical NLP [11], flourished in the 1990s, assisting in the transition from hand-written rules to the automatic manipulation of natural language. Over time, statistical NLP addressed crucial problems like part-of-speech tagging, lexical acquisition, and word sense disambiguation. As a result, the combinational use of symbolic and statistical NLP lead to significant accomplishments, such as the development of the first chatbots [10].

In the last decade, the remarkable development of neural network techniques in NLP has offered the opportunity to tackle a plethora of problems and achieve state-of-the-art results. What makes neural network-based approaches remarkable, such as Deep Learning (DL), is their efficiency conjoined with their exceptional performance on higher-level tasks, making them the new state-of-the-art models. [1]. These challenging problems include but are not limited to conversational, summarization, text classification, text translation, information extraction, and question answering tasks [12].

2.1.1 Machine Reading Comprehension

Machine Reading Comprehension (MRC) [13], simply known as Reading Comprehension (RC) [2], is a task that requires a machine to read a passage of text and answer questions according to the context, performing the task of Question Answering (QA). The Reading Comprehension task can be either extractive, where the computational resource attempts to extract the answer from the corresponding span of text, or generative, where the computer tries to generate a well-formed answer utilizing information from the text.

2.1.2 Language Model

A language model [9] is defined as a distribution that makes probabilistic predictions on the sequence of words that belong to a language. The modern technique of pre-training language models allows the model to comprehend and represent general language offering the ability to adjust on various NLP tasks. Consequently, the process of training additional training features or fine-tuning all features on these generalized language models has brought notable performance gains in tasks such as Text Classification [14], Sequence Learning [15], and also Reading Comprehension [2].

2.2 Web Application Development

A web application [16] is a software system that executes domain logic on a website. Nowadays, web applications can be accessed through the World Wide Web (WWW). A web application consists of the major components of the client and the server side. This demarcation is performed in order to model, implement and deploy individual components which serve different purposes as well as categorize utilized technologies.

2.2.1 Client Side

The client side, also called the front end, is the combination of technologies required to provide an interface to the client and allow the end user to interact with the server. In the case of a web application, the client is the web browser that displays the web application to the end user. Hence, some components ensure the functionality of the front end, such as front-end frameworks. In contrast, other components deal with the visual appearance of the web application, like styling frameworks.

2.2.2 Server Side

The server side, otherwise known as the back end, is the set of utilized technologies responsible for handling incoming requests and responding based on processed business logic. The server side typically consists of the database and the back-end application. The database persistently stores data in an organized manner. The back-end application listens for client requests, accesses the database, applies business logic at will, and responds to the client accordingly.

3. RELATED WORK

3.1 Reading Comprehension

Reading comprehension task has gradually evolved into a task heavily dependent on high-quality and large-scale data. Therefore, several attempts have been made to create such RC datasets. As previously noted by Rajpurkar et al.'s [6], there have been QA datasets with various task formats. Some of the RC task variants which we will not delve into along this section are generative QA (generating an answer based on context), open-domain QA (answering questions based on a set of documents), knowledge base QA (answering questions based on a knowledge base or graph), multiple choice answering and gap filling answering. In this section, we focus on extractive QA datasets, which are datasets that have extracted the annotated answers directly from the corresponding context.

3.1.1 Monolingual Extractive Question Answering datasets

A monolingual dataset is a set of data in one specific language. Most monolingual extractive QA datasets have been built in English, followed by a trend of native datasets being built in other languages. More specifically, SQuAD1.1 (2016) [6] and SQuAD2.0 (2018) [17] have become the archetypal datasets for their proposed methodologies and standards. Moreover, they have been utilized as reference datasets becoming points of comparison among other monolingual datasets which follow similar approaches. For instance, KorQuAD1.0 (2019) [5], FQuAD1.1 (2020) [4], and JaQuAD (2022) [18] applied similar methodologies to SQuAD1.1. Later, SQuAD2.0 added unanswerable questions, followed by FQuAD2.0 (2022) [19] respectively. All these datasets are based on Wikipedia articles of their language, crowdsourced the collection of question and answer pairs. Regarding GreekQA1.0, we also follow the practices of SQuAD1.1.

Some other researchers use different sources to build their QA datasets, showcasing the variety of possible sources of passages. For instance, TrivialQA (2017) [20] was based on trivia websites. In the same year, NewsQA (2017) [21] was published, a dataset that used news articles from CNN. It should be mentioned that our research is limited to only related work of extractive QA datasets with the answer appearing as a span within the context, as this is the practice that we will follow on GreekQA1.0. Finally, all the datasets mentioned above run a challenge with a competitive leaderboard. Researchers follow this practice to encourage further development of language models and utilization of their datasets.

3.1.2 Multilingual Extractive Question Answering datasets

The majority of multilingual QA datasets have been built with some semi-automatic methodology. The two primary multilingual extractive QA datasets are MLQA [22] and XQuAD [23], published around the same period (July 2020). Firstly, MLQA dataset is built by aligning Wikipedia paragraphs on multiple ways. Then, question and answer pairs are collected on these aligned passages. On the other hand, XQuAD was built by picking a portion of SQuAD v1.1 and translating it into ten other languages by professional translators. These datasets are valuable in evaluating cross-lingual QA performance. However, significant issues may be posed, such as unbalanced language distribution and overfitting, mostly due to the translating and aligning methodologies being followed.

Table 3.1: Monolingual extractive Question Answering datasets

Dataset	Language	Passage source	Size
SQuAD 1.1	English	English Wikipedia	100k+
SQuAD 2.0	English	English Wikipedia	150k
FQuAD1.1	French	French Wikipedia	60k+
FQuAD2.0	French	French Wikipedia	79k+
KorQuAD1.0	Korean	Korean Wikipedia	70k+
KorQuAD2.1	Korean	Korean Wikipedia	102k+
JaQuAD	Japanese	Japanese Wikipedia	39k+
SearchQA	English	Conversations	127k
NewsQA	English	CNN News Articles	100k+
TriviaQA	English	Trivia and Quiz-league Websites	95k

Table 3.2: Multilingual extractive Question Answering datasets

Dataset	Languages	Dataset source	Total size (Breakdown)
MLQA	English, Arabic, German, Spanish, Hindi, Vietnamese and Simplified Chinese	Automatic align of Wikipedia paragraphs across multiple languages	42,000+ (12k+ in English, 5k in each other language)
XQuAD	English, Spanish, German, Greek, Russian, Turkish, Arabic, Vietnamese, Thai, Chinese, and Hindi	Professional translation of subset of SQuAD1.1	13,090 (1,190 original from SQuAD1.1)

3.2 Crowdsourcing Platforms for Question Answering Annotation

Crowdsourcing platforms are widely used for the collection of large-scale human-generated questions-answer pairs. First, Rajpurkar et al. [6, 17] used Daemo crowdsourcing platform [24] for SQuAD1.1 and SQuAD2.0. Similarly, it is mentioned by d'Hoffschmidt et al. [4] that a specifically designed platform was developed for annotating passages and collecting the answers and questions for FQuAD. However, most researchers of the monolingual datasets mentioned above avoid to mention the platform used for the crowdsourced collection of question and answer pairs, as it is not the objective of their research.

4. CROWDSOURCING ANNOTATION PLATFORM

A crowdsourcing annotation platform is developed in order to crowdsource the procedure of collecting question and answer pairs. Firstly, the requirements of this web application are analyzed to predefine and document what needs to be built. Then, the set of technologies and the implementation structure are discussed. Finally, the platform implementation is presented, introducing the GreekQA Crowdsourcing Annotation Platform, designed and implemented to create the GreekQA dataset.

4.1 Requirement Analysis

Requirement analysis is concerned with determining and documenting the needs of a software project bound to be developed. This process is critical for any software development project in order to have a typical agreement on the expected implementation and succeed. The requirements can be divided into two major categories of functional and non-functional requirements, as explained below.

4.1.1 Functional Requirements

Functional requirements [25] is a term used in software engineering in order to define the set of functions and the intended behavioral aspects of a system. Regarding the GreekQA Crowdsourcing Platform, the functional requirements are described as follows:

- The platform should have a landing page that functions as the initiation of crowd interest. Said otherwise, this page should inform potential volunteers about the motivation, objective, and importance of the conducted initiative within this platform.
- The platform should have an authentication and identification system in order to control access to the dashboard. It should allow users to sign up, log in and reset their password. The registration process should only be permitted to visitors affiliated with the National and Kapodistrian University of Athens providing a related email.
- The dashboard should enable authorized crowdworkers to read the guidelines in the form of Frequently Asked Questions (FAQ), view their personal information and change them at will. Moreover, statistics of the user should be displayed regarding their contribution. Finally, the storage of personal information should comply with the General Data Protection Regulation (GDPR). Therefore, all personal information is stored in a way allowing their erasure and the anonymization of their contribution.
- The dashboard should also enable access to the page of contribution. This page displays an unannotated paragraph, allowing users to write a question and select a span of the context as the answer. As proposed by the standards of the dataset, only three to five question and answer pairs should be collected per paragraph. Moreover, a question-answer pair deletion is allowed in case of a mistake during the annotation process. Finally, a submission button enables users to submit their annotations and continue to the next unannotated paragraph.

- A database management system should be utilized for the needs of authorization of the crowdsourcing process. Authorized admins should have access to view general and user-specific statistics regarding the number of annotated paragraphs and collected question-answer pairs. Moreover, this system should allow admins to review any annotated passage.

In the Unified Modeling Language (UML) [26], functional requirements can be expressed through use case diagrams, summarizing the interactions of actors with the system [25]. In a use case diagram, an actor is associated with several possible use cases which may interact. Additionally, the use cases may include other use cases as part of a sub-process or extend other use cases, enhancing them.

The use case diagram shown in Figure 4.1. summarizes the interactions of a visitor who has not been authorized to access the platform. Similarly, the interactions of an authorized crowdworker in the platform and an authorized admin in the database management system are demonstrated in Figure 4.2 with a use case diagram.

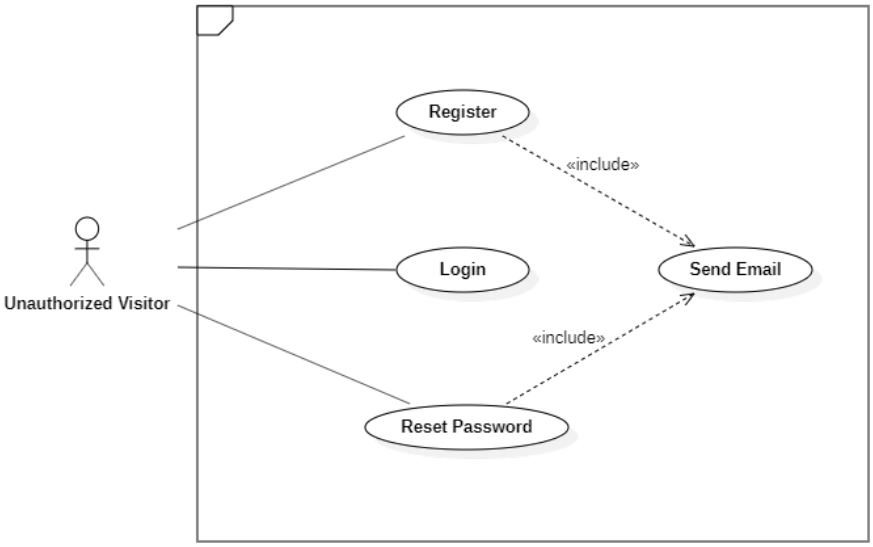


Figure 4.1: Use case diagram of an unauthorized visitor

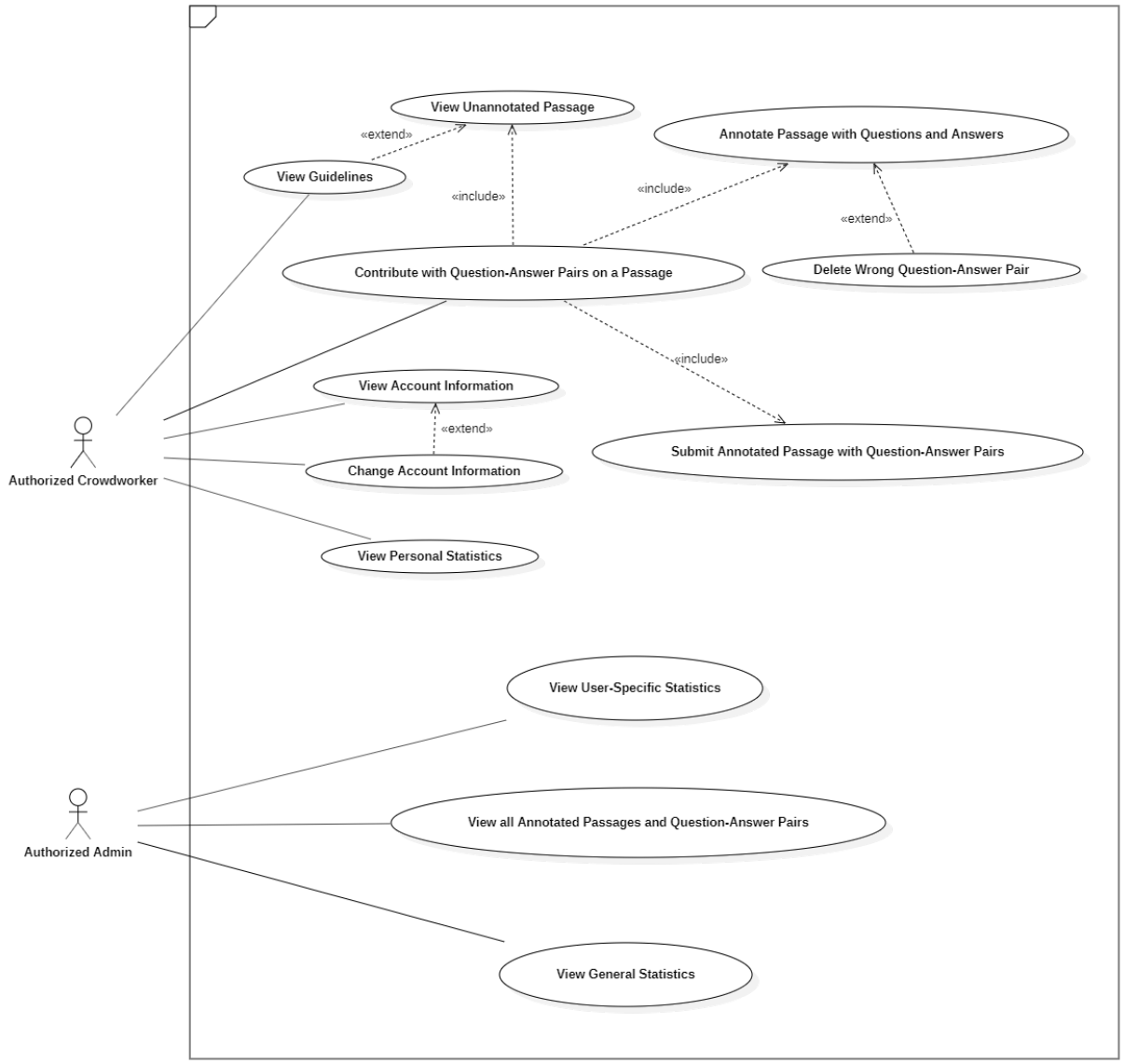


Figure 4.2: Use case diagram of an authorized crowdworker and an authorized admin

4.1.2 Non-Functional Requirements

Non-functional requirements [27] are generally described as the non-behavioral attributes and constraints of a system. However, the definition of this term has been a debatable topic due to the lack of consensus, as stated by Glinz [27]. The non-functional requirements of the GreekQA Crowdsourcing Annotation Platform are listed as follows:

- A pleasant and portable user experience should be offered to user. Therefore, the User Interface (UI) of the platform should have some specific characteristics. Firstly, intuitive behaviour is desired by the user. Hence, the design of the platform should be consistent and minimalistic, following common practices regarding the design of specific pages for reasons of familiarity. For instance, authentication pages are usually designed in a specific format over various web pages and applications, which should be followed for this platform as well. Moreover, the design as a whole needs to responsive for all devices and screen sizes, allowing the user
- A reliable and secure set of software technologies must be used for the implementation. Each part of the system should be implemented on regularly maintained technologies with the minimum amount of critical failures. Moreover, the communication between the different parts of the system needs to be flawless, secure, and successful at all times.
- The platform needs to be deployed on reliable technology infrastructure. The servers hosting the individual parts of this platform must have the minimum down time and be properly set up in order to serve the platform faultlessly.

4.2 Application Implementation

The implementation of an application is composed of a set of components, also known as a technology stack. As explained before in Section 2.2, a web application implementation is split into the major components of the client and the server side. Therefore, in this section, the implementation of each side is described along with their utilized technologies.

4.2.1 Client-Side Implementation

The client side of the application is based on JavaScript, incorporated with a set of frameworks, libraries, and packages. The choice of a limited number of modern and well-maintained technologies provides stability to the implementation, making it future-proof. The main language of the front end, JavaScript[1] (JS), is the most popular programming language for web development. This language may be utilized for both client and server sides combined with suitable libraries and frameworks. Below, the choice of the two primary technologies is justified, and their use is explained.

[1] `https://javascript.com`

React.js

React.js[2] (React) is the most frequently used front-end JavaScript library maintained by Meta and an open-source community. The main purpose of this library is to facilitate the process of creating interactive UIs. This library offers key benefits which motivated us to make this choice over the second most popular framework, Angular.js (Angular).

In terms of performance, React makes use of a virtual Document Object Model, prevailing the performance of frameworks that use an actual one, as Angular uses. An actual Document Object Model (DOM) is a programming interface responsible for the representation of a page to enable the interaction of programming languages with the page. Therefore, a set of Application Programming Interfaces (APIs) allows individual technologies to alter the document accordingly. React achieves higher performance with a virtual DOM, a virtual representation of a DOM, as during a page refresh, only the changed parts are updated, in contrast with the real DOM that updates the whole page.

Regarding productivity, the learning curve of React is shallower than Angular. Moreover, React is believed to be ideal for implementing a small application, like this platform, compared to Angular, which is superior in larger applications. Most importantly, React assists with the reusability of components simply and efficiently, reducing implementation time and code effectively.

Delving into the importance of code reusability, code structure based on components is one of the critical aspects of utilizing React effectively and assuring a future-proof code. We make use of the commonly used concepts *component, layout, and page* for our file and code structure. Firstly, *components* are reusable blocks of code, which we categorize based on the involved feature, *authentication* and *dashboard*. This way, primary related components are put together, while standard and small components, such as *button*, are put in a *shared* directory. Then, moving to *layouts*, we create two high-level layouts, *authentication and dashboard layouts*, designed for these two major types of pages. This practice aims to avoid repeating code over pages that share the same design. Finally, pages are categorized in the same way. However, the code of a whole page is now limited to a few lines of code, requiring only the use of a specific layout and a set of components. To sum up, this directory and code structure offers the ability to create new pages with significant convenience, implementation code, and time. The source code of Login Page is shown in Listing 4.1. in order to demonstrate the code structure, while Figure 4.3 is an overview of the directory structure of the client-side implementation.

Listing 4.1: Source code of Login Page

```
import { ReactComponent as LoginSvg } from 'assets/loginsvg.svg';
import { AuthLayout } from 'layouts/AuthLayout';
import { LoginForm } from 'components/Auth/LoginForm';

export function LoginPage() {
    return (
        <AuthLayout
            form={ <LoginForm /> }
            illustration={ <LoginSvg /> }
        />
    );
}
```

[2] https://reactjs.org

Tailwind CSS

Tailwind CSS[3] is a styling framework for customizing the visual design of an application. It is unopinionated with no pre-applied styles, while it is highly customizable. With the use of this framework, Cascading Style Sheets (CSS) become mostly redundant, avoiding the accompanying complexity and difficulty of this styling method. Even though Tailwind involves low-level coding, the styling process within the components makes it easy to use. Therefore, Tailwind pairs ideally with component-based libraries like the library of our choice, React.

Compared to other viable visual styling options, such as Bootstrap and Bulma, they are both heavily opinionated in design, making the application distinctive regarding the framework used. In contrast, Tailwind is highly customizable, allowing the application to create a unique design and visual appearance. Lastly, Tailwind is well-documented, open-source, supported by a large community, and currently the most popular styling framework with abundant examples. Consequently, Tailwind meets all requirements of this platform regarding design responsiveness, delivering a pleasant experience to the user.

4.2.2 Server-Side Implementation

The server side is solely based on Firebase, which offers all the necessary back-end capabilities for the GreekQA platform. Firebase[4] is a Backend-as-a-Service (Baas) brought by Google, providing a set of services to assist the development of an application without the implementation, use, and deployment of a custom back-end server. The two services required for the GreekQA platform are an authentication and identification system and a database. Firebase Authentication is used with an email and password-based authentication system, which is integrated into the front-end with the firebase module[5]. Similarly, Cloud Firestore, a NoSQL database, stores the general and user-specific contribution statistics as well as the under construction dataset. The structure chosen to store the dataset within the database is proportionate to the standard JSON structure of the exported dataset, as demonstrated in Section 6.1. Finally, it should be noted that the database is secured with specific rules for authorized access.

4.2.3 Deployment

The client side was deployed on the servers of the National and Kapodistrian University of Athens under a subdomain of the Department of Informatics and Telecommunications, where the website can currently be accessed. Regarding the back-end deployment, the no-cost Spark Plan of Firebase was adequate for the daily traffic of the platform. Hence, no additional setup or cost was required for using the Firebase Authentication and the Cloud Firestore.

[3] `https://tailwindcss.com`

[4] `https://firebase.google.com`

[5] `https://www.npmjs.com/package/firebase`

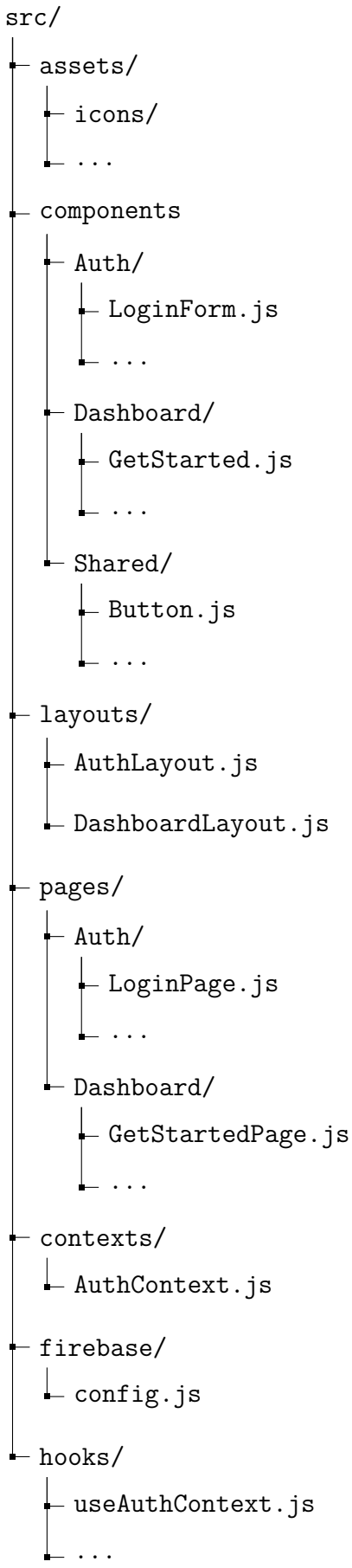


Figure 4.3: Directory structure of client-side source code

4.3 Application Demonstration

In this section, we briefly demonstrate the implementation of the GreekQA Crowdsourcing Annotation Platform. Firstly, the authentication pages are shown and discussed for their use. Then, the dashboard pages are explored and examined. Lastly, the mobile-friendliness of the application is showcased.

4.3.1 Auhentication Pages

Landing Page

The Landing Page (Figure 4.4) is a single web page that functions as the initiation of crowd interest. Therefore, this page informs potential volunteers about the motivation, objective, and other aims of this initiative. In order to attract more people to contribute, the great importance of creating a Greek QA dataset in the scarcity of other similar native datasets is emphasized.

Sign Up Page

The Sign Up Page (Figure 4.5) allows users to independently create an account and contribute to the GreekQA dataset. The registration form requires the first and last name, an email address, and a secure password. Due to security reasons, the password must consist of at least eight characters long and contain at least one symbol and one number without Greek characters. Moreover, only emails affiliated with the National and Kapodistrian University of Athens are allowed to sign up.

Verify Email Page

The Verify Email Page (Figure 4.6) appears while awaiting the validation of the registered email address. Users may verify their email by clicking a link within an email sent to their email address. Until their email is verified, their account is locked to this page. If the email hasn't been received, a request for resending the email is available on this page.

Login Page

The Login Page (Figure 4.7) enables users to access the platform by entering the credentials with which they signed up. The login form requires the email address and the password of the user for the process of user authentication.

Reset Password Page

The Reset Password Page (Figure 4.8) assists users who have forgotten their passwords in regaining access to their accounts. After entering their email and requesting the password reset, an email is sent to the user's address with a link to set a new password.

Figure 4.4: Landing Page of GreekQA Crowdsourcing Annotation Platform

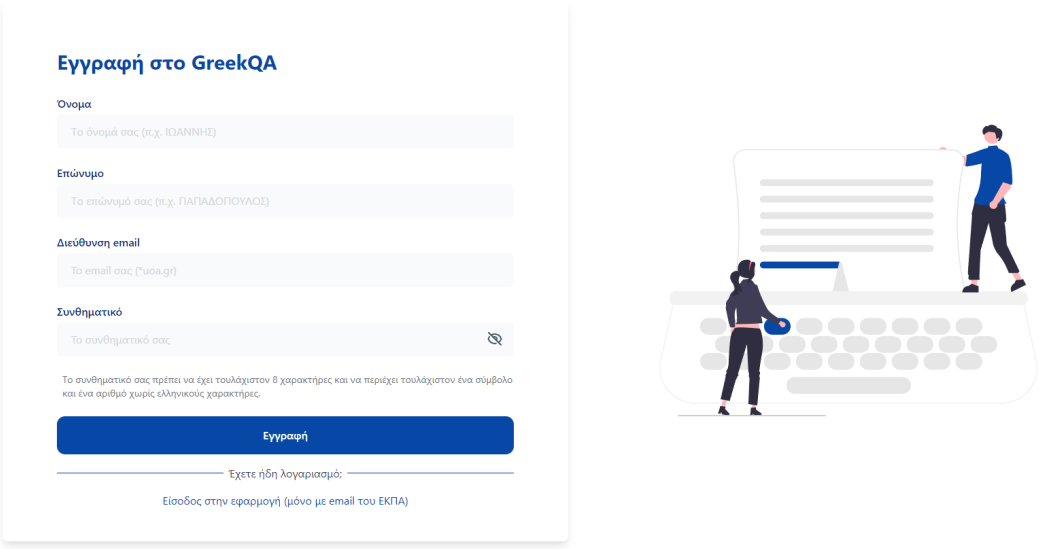


Figure 4.5: Sign Up Page of GreekQA Crowdsourcing Annotation Platform

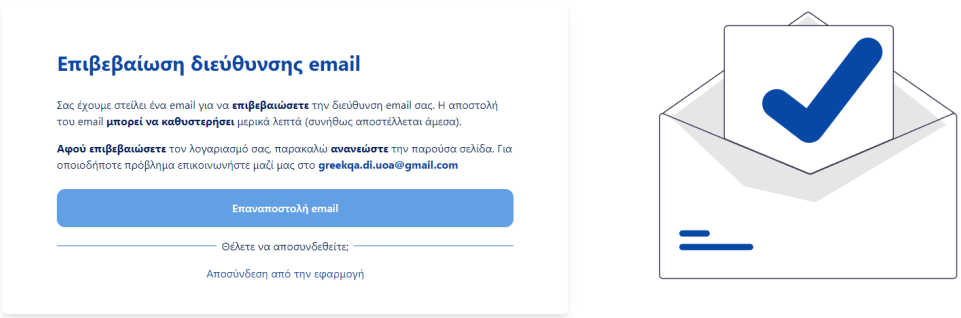


Figure 4.6: Verify Email Page of GreekQA Crowdsourcing Annotation Platform

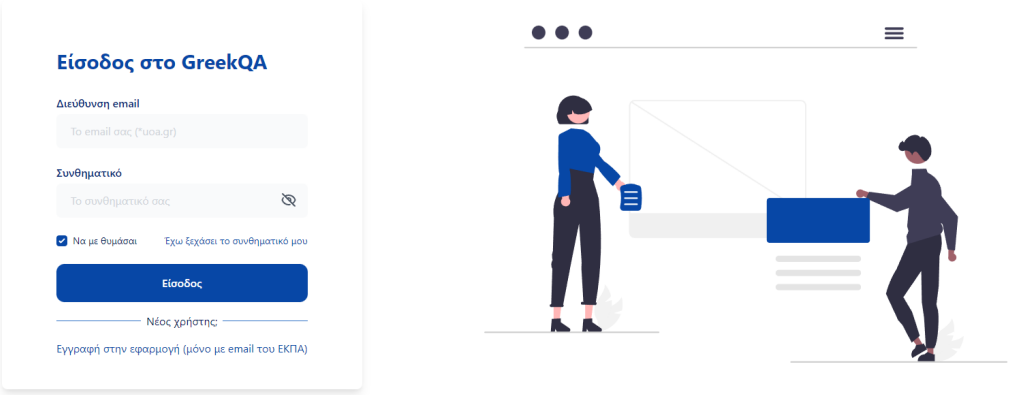


Figure 4.7: Login Page of GreekQA Crowdsourcing Annotation Platform

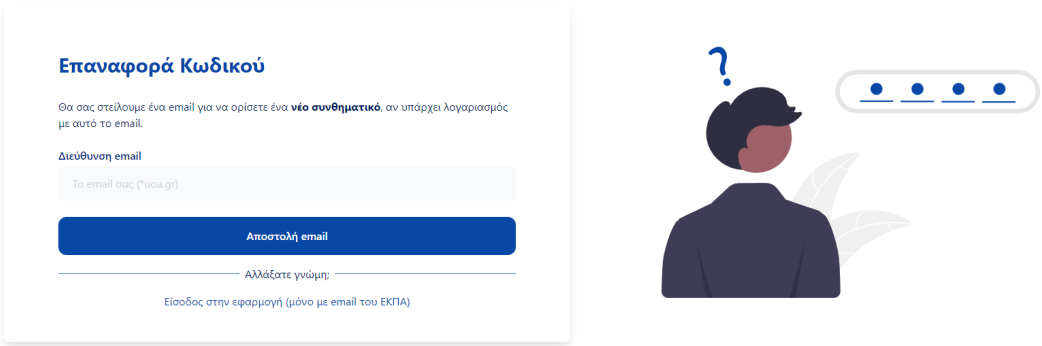


Figure 4.8: Reset Password Page of GreekQA Crowdsourcing Annotation Platform

4.3.2 Dashboard Pages

Get Started Page

The Get Started Page (Figure 4.9) is the landing page of the dashboard. This page welcomes users to the platform, urging them to read the guidelines and begin contributing with their questions and answers.

Guidelines Page

The Guidelines Page (Figure 4.10) provides instructions for use in the form of Frequently Asked Questions (FAQ). These instructions begin with a brief discussion of the GreekQA dataset objectives and motives. Then, various examples from all categories of reasoning [6], and possible types of answers were provided, noting the importance of high diversity and difficulty in questions. Additionally, volunteers are urged to refrain from using exact words and phrases from the passage in their questions. Lastly, it is noted that all posed questions must be answerable.

Profile Page

The Profile Page (Figure 4.11) displays the personal information of the user, containing their name and email address. Moreover, statistics regarding their contributions are provided. Besides the display of information, users are allowed to change their password.

Contribution Page

The Contribution Page (Figure 4.12) enables users to annotate passages for the GreekQA dataset, making it the most significant page of this platform. Firstly, an unannotated para-

graph of an article appears on the page. The user can write a question, mark the answer span in the passage and add the question to a temporary list of questions. This way, the user can continue to add three to five question and answer pairs. In case of a mistake, a question may be deleted from the temporary list shown beneath the paragraph. Finally, after the user has completed at least three and not more than five question-answer pairs, they may submit the annotated paragraph and proceed to the next passage. The users are urged to follow the guidelines and submit their annotations before leaving the page.

Figure 4.9: Get Started Page of GreekQA Crowdsourcing Annotation Platform

Figure 4.10: Guidelines Page of GreekQA Crowdsourcing Annotation Platform

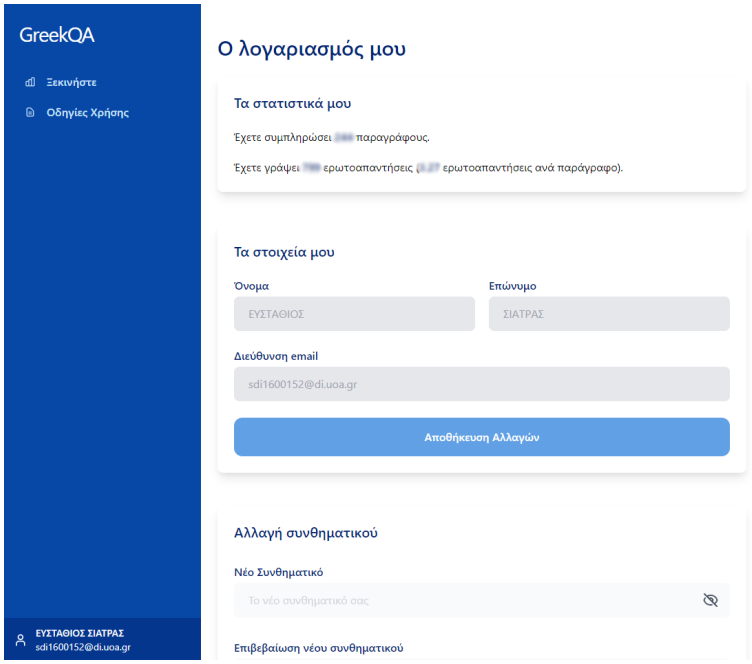


Figure 4.11: Profile Page of GreekQA Crowdsourcing Annotation Platform

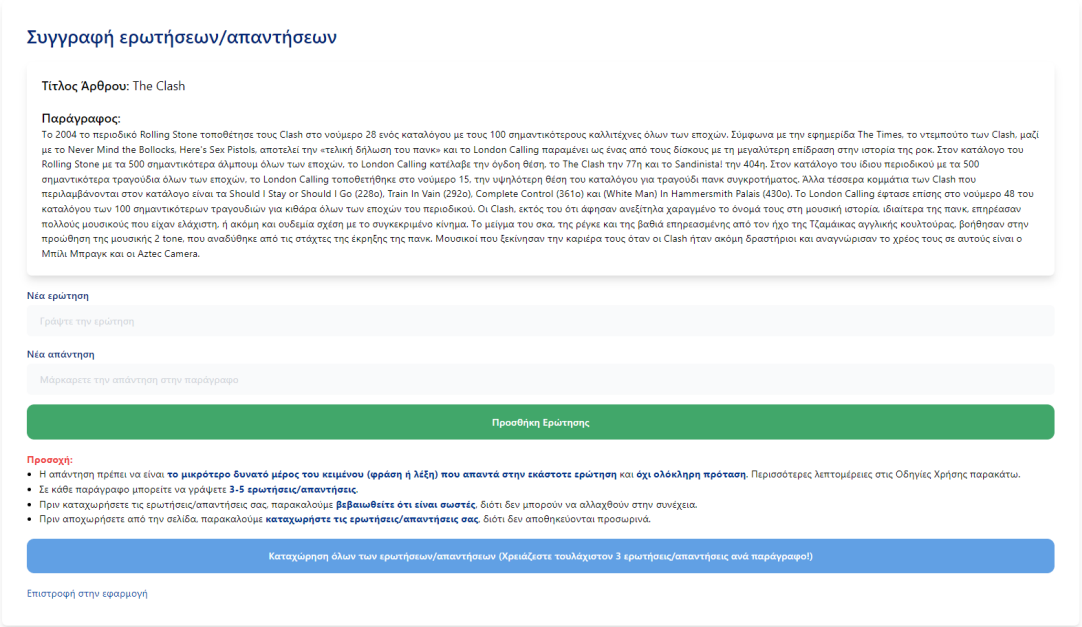
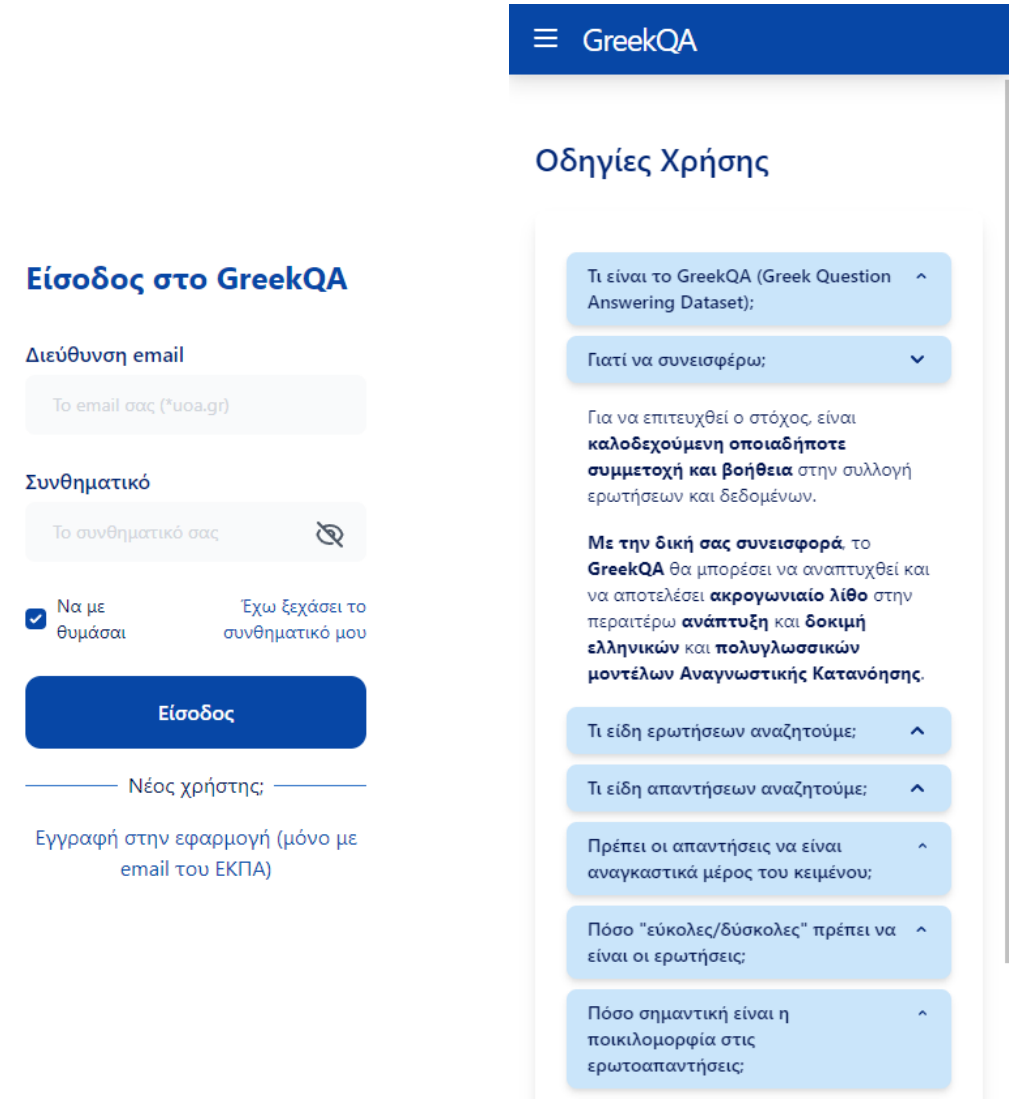


Figure 4.12: Contribution Page of GreekQA Crowdsourcing Annotation Platform

4.3.3 Responsive Design

All pages mentioned above are implemented with responsive design. Thus, the application is also mobile-friendly. For demonstration reasons, the mobile versions of two sample pages are shown in Figure 4.13.



(a) Mobile version of Login Page (b) Mobile version of Guidelines Page

Figure 4.13: Mobile versions of two sample pages

5. DATASET COLLECTION

This chapter describes the dataset collection process, which follows Rajpurkar et al.'s [6] proposed standards in creating the SQuAD1.0 and SQuAD1.1 datasets, as well as the succeeding initiative of d'Hoffschmidt et al. [4] in building the FQuAD1.0 and FQuAD1.1 datasets. This approach consists of two phases, presented as follows. In the first phase, paragraphs based on Greek Wikipedia articles are selected and retrieved. Later, the task of annotating curated passages to collect question and answer pairs is crowdsourced using the GreekQA Crowdsourcing Annotation Platform. Moreover, for the posed questions in the development and test sets, we obtain additional answers. A more detailed account of this process is given in the following section.

5.1 Passages Collection

We began with listing candidate articles from Greek Wikipedia and retrieving them based on the methodologies and criteria discussed below. Then, we sampled a portion of these articles and retrieved them. Next, we broke down the sampled articles into paragraphs which were eventually used as passages for the development of this dataset.

5.1.1 Featured Articles of Greek Wikipedia

The first set of selected articles are the featured articles of Greek Wikipedia. D'Hoffschmidt et al. [4] suggested that articles which Wikipedia characterizes as quality articles may be used to create a QA dataset. Likewise, we consider the set of 124 featured articles out of 214,581 articles of Greek Wikipedia for building the GreekQA dataset. Greek Wikipedia's featured articles[1] are accurate, neutral, comprehensive, complete, properly formatted, and conform to Wikipedia's criteria discussed below. Moreover, these articles are showcased on the main page of Greek Wikipedia regarding their subject area.

Greek Wikipedia sets five primary conditions referencing a quality article[2]:

- **It contains an encyclopedic topic** discussing a subject not covered by other relevant articles.
- **It has a proper structure** with an appropriate title, introduction, and definition of the objective.
- **It is well-written** with clear, comprehensive, accurate, and detailed content using proper grammar.
- **It is well documented** with verifiable sources and follows the present scientific knowledge, resulting from original writing referring to others' research.
- **It follows proper browser navigation procedures**, which suggests that the article can be found in the related Wikipedia categories. Moreover, readers may browse through the article using internal and external links.

[1] Greek Wikipedia: Featured articles
[2] Greek Wikipedia: The perfect article

5.1.2 Top Articles of Greek Wikipedia's Internal PageRanks

The second methodology of selecting articles is by computing the Greek Wikipedia's internal PageRanks ranking score and using a portion of the top articles. This procedure was presented by Rajpurkar et al. [6], who computed the English Wikipedia's internal PageRanks scores to obtain first-rate articles for SQuAD.

The term PageRank was proposed by Page et al. [28] as a computational algorithm for ranking the relative importance of web pages using a web graph. PageRank has become one of the cornerstone algorithms for ranking the relevance of web pages by computing a score on a scale of 0 to 10. The basic concept of PageRank is that a web page is cumulatively important based on the significance of other web pages pointing to this page. Delving into the basis of this algorithm intuitively, the behavior of an ideal *random surfer* [29] is simulated. Otherwise said, a random surfer starts their web exploration from a web page chosen randomly and uniformly, followed by clicking on one link of the current page and continuing this process repeatedly. This sequence may come to an end, as the ideal surfer may stop clicking on links set by a specific probability and restart their web walk from a new random page. Consequently, these web walks and the distribution of visits on each web page are the basis for the computation of this algorithm, which is described extensively by [28, 29].

The PageRank algorithm can be applied to rank web pages offline using a web graph [29]. The structure of a web graph is a directed graph, viewing pages as nodes and hyperlinks as arcs. A link can be perceived as a forward link pointing with a link to another site. Vice versa, it is also a backlink for the other page, as the other page is pointed by the same link. Intuitively, the computation of PageRank is based on the number of backlinks. Therefore, it may be used on any web graph like a crawled version of the World Wide Web (WWW). In our case, PageRank will be utilized for ranking the internal pages of Greek Wikipedia, which is just a tiny web graph compared to the WWW.

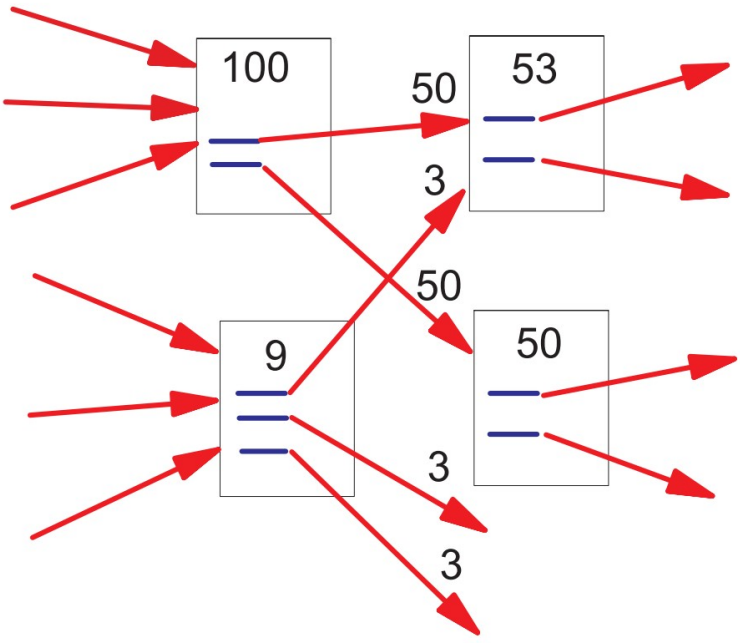


Figure 5.1: Simplified PageRank Calculation [28]

We compute the Greek Wikipedia's internal PageRanks using Project Nayuki's implementation[3] and adjust this implementation for use on the Greek Wikipedia, as shown by Rajpurkar et al. [6]. Hence, we download the latest Greek Wikipedia database dumps of pages[4] and page links[5]. Then, we use these two database dumps on Project Nayuki's Wikipedia's internal PageRanks program and get the top 10000 pages of the highest internal PageRank score.

For instance, 10 randomly sampled pages of Greek Wikipedia articles with their calculated PageRank scores are provided in Table 5.1 below.

Table 5.1: PageRank scores of 10 sampled pages of Greek Wikipedia articles

Greek Wikipedia article	PageRank score
Δικαστήριο της Ευρωπαϊκής Ένωσης	4.872
Πελοποννησιακός πόλεμος	4.774
Στάδιο ποδοσφαίρου	4.683
Μοντερνισμός	4.650
Οσπιτάλιοι Ιππότες	4.546
Ηνωμένα Έθνη	4.460
Αφροδίτη (πλανήτης)	4.330
Σωκράτης	4.267
Σεισμός	4.125
Ινδικός Ωκεανός	4.046

5.1.3 Articles Collection

For the purposes of the GreekQA1.0 dataset, the set of the 124 featured articles of Greek Wikipedia were retrieved, from which 21 articles were randomly sampled. Regarding the candidate articles from the computation of the Greek Wikipedia's internal PageRanks, they will be considered for the extended versions of this dataset. This process resulted in 15 articles for the training set, 3 for the development set, and 3 for the test set of GreekQA1.0.

In order to parse and collect the content of Greek Wikipedia articles into a directory of text files, we developed a relevant script based on the Wikipedia Python library[6]. Provided a list of requested article titles, the script finds the requested pages and extracts the content of the articles in separate text files.

The steps of the implemented script are described in Algorithm 1:

[3] https://www.nayuki.io/page/computing-wikipedias-internal-pageranks
[4] https://dumps.wikimedia.org/elwiki/latest/elwiki-latest-page.sql.gz
[5] https://dumps.wikimedia.org/elwiki/latest/elwiki-latest-pagelinks.sql.gz
[6] https://pypi.org/project/wikipedia

Algorithm 1 Wikipedia Content Extractor Script Algorithm

Input: $Wikipedia$, $titles\_list$
Output: $output$ directory containing the content of articles

```
make_dir(output)
for title in titles_list do
    page_ref ← Wikipedia.page(title)
    if page_ref ≠ NULL then
        file_ref ← open_file(output + title)
        write_file(file_ref, page_ref.content)
        close_file(file_ref)
    end if
end for
```

5.1.4 Passages Curation

We processed the collected articles and removed irrelevant content, such as subtitles, tables and URLs in order to extract a set of separate paragraphs for each article. Following the methodology of [6, 4], only the paragraphs containing at least 500 characters were kept. Additionally, a portion of paragraphs primarily concerned with mathematical functions were discarded from the dataset, following the KorQuAD1.0 [5] proposal. Then, we split the total 21 articles into 15 in the training set (71.4%), 3 in the development test (14.3%), and 3 in the test set (14.3%). It should be noted that the primary concern of this article level split was to maintain the range of subject matters in each set and avoid bias. Concurrently, we approached to the 80/10/10 split ratio of sets regarding the passages, resulting in 305 passages split into 243 in the training set (79.6%), 31 in the development set (10.2%), and 31 in the test set (10.2%).

5.2 Question and Answers Pairs Collection

The collection of the questions and answers on the curated passages was crowdsourced using the GreekQA crowdsourcing annotation platform. All the crowdworkers contributed voluntarily; the majority were students associated with the AI Team of the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens.

5.2.1 Guidelines

The approach of writing questions and answers follows the presented methodology of SQuAD1.0 [6]. The contribution to the GreekQA dataset begins with a passage given to the volunteer. Then, the volunteer is assigned to write down three to five questions regarding the context. For each question, the volunteer selects the shortest span in the passage which answers the question.

In order to inform the volunteers about the purposes and the usage of the GreekQA crowdsourcing annotation platform, instructions for use were provided in the form of Frequently Asked Questions (FAQ). These instructions begin with a brief discussion of the GreekQA dataset objectives and motives. Then, we set the guidelines as suggested by SQuAD1.0. More specifically, various examples from all categories of reasoning from [6] and pos-

sible types of answers were provided, noting the importance of high diversity and difficulty in questions. Additionally, volunteers were urged to refrain from using exact words and phrases from the passage in their questions. Lastly, all posed questions must be answerable, as unanswerable questions are planned to be added to the extended versions of the GreekQA dataset.

5.2.2 Additional Answers Collection

In order to evaluate the bias in answering, one additional answer was collected for every question in the development and test sets [4]. The process of collecting additional answers starts with presenting each question and the corresponding passage and hiding the given answer. Then, the volunteer is asked to answer the question by selecting the smallest span within the passage, as the proposed guidelines suggest. As explained further below in Subsection 6.2.2, these additional answers will be utilized in order to measure the human performace on GreekQA.

5.2.3 Manual Review

After the collection of question and answer pairs, 50% of the total questions were randomly sampled and manually reviewed, as similarly shown by d'Hoffschmidt [4]. The purpose of this manual review is to ensure a high level of quality of the questions and answers. Thus, if any reviewed pair did not follow the designated standards, it was removed from the GreekQA dataset.

6. DATASET ANALYSIS AND EVALUATION

6.1 Dataset Structure

In this section, we present the structure of our dataset. The conducted dataset collection resulted in the GreekQA1.0 dataset of 1008 questions on 305 passages and 21 articles in total. As mentioned earlier, we split the total 21 articles into 15 in the training set, 3 in the development test, and 3 in the test set. Our primary objectives of this article level split was to maintain the range of subject matters in each set, avoid bias, and approach to the 80/10/10 split ratio of sets regarding the passages. The statistics of the GreekQA1.0 dataset can be summarized in Table 6.1.

Table 6.1: Statistics of GreekQA1.0 dataset

Dataset	Articles	Passages	Questions
Training Set	15 (71.4%)	243 (79.6%)	796 (79.3%)
Development Set	3 (14.3%)	31 (10.2%)	102 (10.2%)
Test Set	3 (14.3%)	31 (10.2%)	106 (10.5%)
Total Dataset	21 (100.0%)	305 (100.0%)	1004 (100.0%)

Regarding the structure of GreekQA1.0, the same JSON structure of SQuAD1.1 dataset [6] is followed. In Listing 6.1, an overview of this structure of each set is briefly presented, while in Listing 6.2 the structure of an example paragraph and a question on this paragraph is provided.

Listing 6.1: Overview of JSON structure of GreekQA1.0 sets

```
{
    "version": "v1.0",
    "data": [
        ...,
        {
            "title": "Ανταρκτική",
            "paragraphs" : [
                ...
            ]
        },
        ...
    ]
}
```

Listing 6.2: JSON structure of a sample paragraph in GreekQA1.0 development set

```
{
    "pid": "s3589CgY7rPtVE"
    "context": "Η πέμπτη ταινία της σειράς Χάρι Πότερ...",
    "qas": [
        ...,
        {
            "id": "dcf86a81ce994b4a833d57e515145bf9",
            "question": "Πότε βγήκε η Γουάτσον...;",
            "answers": [
                {
                    "answer_start": 991,
                    "text":  "Φεβρουάριο του 2010"
                },
                {
                    "answer_start": 1006,
                    "text": "2010"
                }
            ],
        },
        ...
    ]
}
```

6.2 Answer Analysis

6.2.1 Answer Length Analysis

The length of answers can extract significant information for the dataset, compared to a standard dataset, such as SQuAD1.1 [6], as shown by d'Hoffschmidt et al. [4]. Therefore, we calculate the tokens per answer in GreekQA1.0 and SQuAD1.1 to compare these two datasets. Firstly, we use the Natural Language ToolKit (NLTK) to split the answers into tokens. Then, we remove the Greek articles *ο, η, το, τα, στο, στη, ένας, μια, ένα* and their variants from the answers of GreekQA1.0, as well as the articles *a, an, the* from the answers of SQuAD1.1. Finally, we count the number of tokens per answer. Interestingly, the average length of answers in GreekQA1.0 is 2.69, approaching SQuAD1.1's, which is 2.72. This finding designates that GreekQA1.0 may pose similar difficulty in finding exact answers. The number of tokens per answer distribution for GreekQA1.0 and SQuAD1.1 is provided in Figure 6.1.

6.2.2 Answer Diversity Analysis

The diversity in answers is demonstrated by labeling the types of answers using various set of tools, as suggested by d'Hoffschmidt et al. [4]. We first find *locations* and *people* in answers with Entity Recognition using spaCy [30]. Then, we use regular expression rules to extract *dates* and *other numeric* entities in answers. Lastly, the rest are manually categorized as *adjectives*, *common nouns*, *verbal nouns*, and *other entities*. The outcome of this categorization in Table 6.2 showcases the heterogeneity of answers in the dataset.

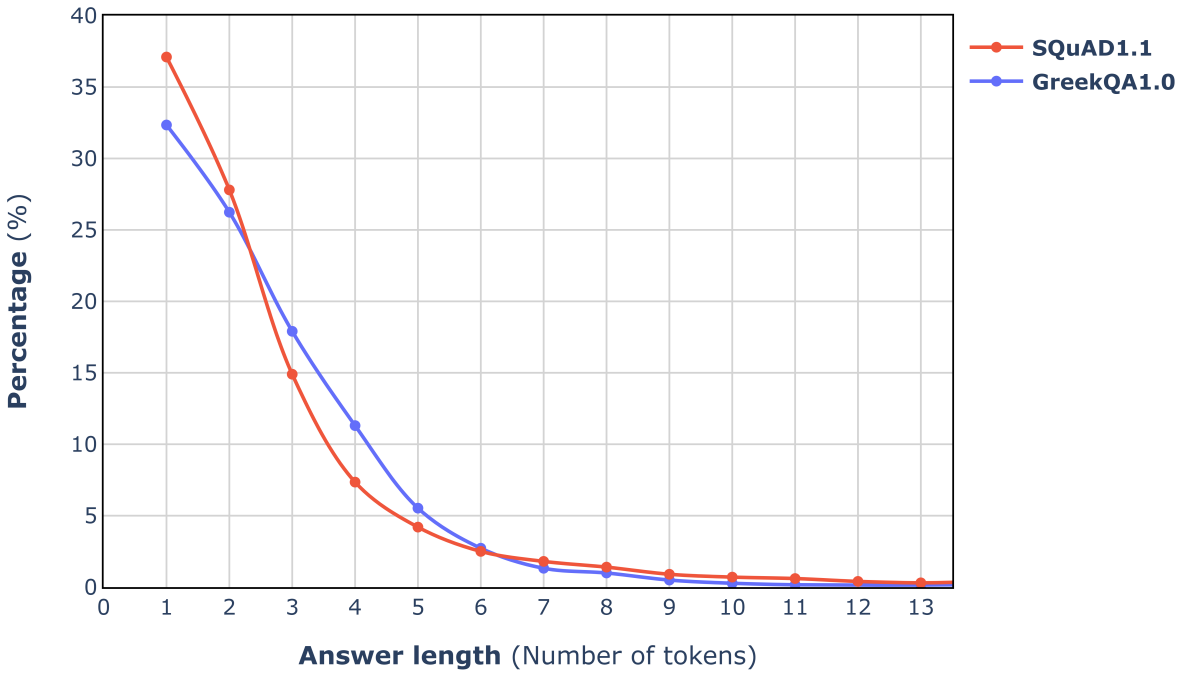


Figure 6.1: Number of tokens per answer distribution for GreekQA1.0 and SQuAD1.1

Table 6.2: Types of answers in GreekQA1.0 development set

Answer type	Example	Percentage (%)
Date	Μάρτιο του 2014	29.4
Other Entity	Acrocanthosaurus atokensis	16.7
Other Numeric	11,5 μέτρα	14.7
Person	Στίβεν Τσμπόσκι	13.7
Location	Νέα Υόρκη	7.8
Adjective	εμπρόσθια	7.8
Common Nouns	δικηγόροι	6.9
Verbal Nouns	σχεδίαση γραφείων	3.0

6.3 Question Analysis

Similarly, we examine the diversity of the questions in this dataset. Firstly, we categorize the types of questions based on the interrogative words being used in the development set of GreekQA1.0. The most frequent question word is *Which* (27.5%), which illustrates a diverse set of common, entity and other nouns. The next two most represented interrogative words are *When* (24.5%) and *How many* (12.7%) questioning about dates and other numeric answers respectively. The results of this categorization are summarized in Table 6.3 and demonstrate the wide variety of questions of the dataset.

Table 6.3: Types of questions based on interrogative words in GreekQA1.0 development set

Question type	Example	Percentage (%)
Which	Ποιο ήταν το πιο αξιοσημείωτο χαρακτηριστικό…	27.5
When	Πότε σταμάτησε η λειτουργία της σχολής…	24.5
How many	Πόσους μήνες διαρκούσε το αρχικό στάδιο…	12.7
What	Τι σχήμα είχε ο εγκέφαλος του…	9.8
Who	Ποιος είχε την εποπτεία του…	9.8
Where	Που τοποθετείται χρονικά η αρχή της ιστορίας…	8.8
How	Πώς ονομάζεται η πέμπτη ταινία της…	2.0
Other	…ήταν μικρός ή μεγάλος θηρευτής;	4.9

6.4 Reasoning Analysis

Finally, we analyze the types of reasoning required to answer questions based on the passage, as proposed by Rajpurkar et al. [6]. The categories of reasoning are briefly explained as follows:

- **Synonymy:** Require use of synonyms in the question and the correspondence in the passage.
- **World knowledge:** Require use of world knowledge in the question and the correspondence in the passage.
- **Syntactic variation** Require change of syntactic structure of context for the question.
- **Multiple sentence reasoning** Require combined use of multiple sentences in the passage for the question.
- **Ambiguous** No significant reasoning noticed or disagreement with the question and answer pair.

The percentages of reasoning types [6] is presented in Table 6.4. The answer is underlined and the words relevant to the corresponding reasoning category are bolded. It should be noted that the categorization is exclusive. The results of this labeling process demonstrate that the majority (90.2%) of the questions in the development set pose significant difficulty in order to be answered due to notable lexical or syntactic variation of the questions and the corresponding passages.

Table 6.4: Types of reasoning [6] in GreekQA1.0 development set

Reasoning	**Example**	**Percentage (%)**
Synonymy	*Question:* Ποια είναι τα **κύρια στοιχεία** του Μπαουχάους; *Sentence:* **Βασικά χαρακτηριστικά** του Μπαουχάους ήταν η απλότητα, η λειτουργικότητα και η χρηστικότητα…	54.9
World knowledge	*Question:* Τι **δουλειά** έκαναν οι Βρετανοί γονείς της Εμμα Γουάτσον; *Sentence:* Οι γονείς της, Ζακλίν Λούεζμπι και Κρις Γουάτσον, είναι Βρετανοί **δικηγόροι**.	2.9
Syntactic variation	*Question:* Ποιος έγραψε το **μυθιστόρημα Ballet Shoes**; *Sentence:* …στην **ταινία Ballet Shoes** του καναλιού BBC το 2007, που αποτέλεσε προσαρμογή του **ομώνυμου μυθιστορήματος** της Νόελ Στρίτφαϊλντ.	20.6
Multiple sentence reasoning	*Question:* Τι εισπράξεις είχε **η πέμπτη ταινία της σειράς Χάρι Πότερ**; *Sentence:* **Η πέμπτη ταινία της σειράς Χάρι Πότερ**, Ο Χάρι Πότερ και το Τάγμα του Φοίνικα… κυκλοφόρησε το 2007. **Η ταινία** αποτέλεσε μεγάλη εισπρακτική επιτυχία και έκανε ρεκόρ καλύτερου ανοίγματος τριημέρου παγκοσμίως με εισπράξεις 332,7 εκατομμυρίων δολαρίων.	11.8
Ambiguous	*Question:* Ο Ακροκανθόσαυρος ήταν μικρός ή **μεγάλος θηρευτής**; *Sentence:* Καθώς ο Ακροκανθόσαυρος ήταν **μεγάλος θηρευτής**, είναι αναμενόμενο …	9.8

6.5 Human Performance Evaluation

For this section, a brief reference to the two widely used metrics for evaluating QA models' accuracy is required. The first metric is Exact Match (EM), which measures the percentage of predictions that exactly match the ground truth answer. The second is F1-Score, which measures the average overlap between the prediction and ground truth answer. As similarly followed by SQuAD [6], the Greek articles *ο, η, το, τα, στο, στη, ένας, μια, ένα* and their variants, as well as punctuation, are ignored during the evaluation process.

Based on the above, the human performance on development and test sets is evaluated utilizing the collected additional answers, similar to SQuAD [6]. In order to measure human performance, one answer is taken as a prediction and the other as the ground truth using the two prementioned metrics. This evaluation process results in the human performance of 79.4% EM, 91.9% F1 scores on the development set and 81.1% EM, 93.3%F1 scores on the test set, as summarized in Table 6.5. The human performance scores are noticed to be slightly higher than other standard datasets, such as SQuAD1.1 [6] and FQuAD1.1 [4]. This observation could indicate higher bias, possibly due to the use of one additional answer instead of proposed two.

Table 6.5: Human Performance on GreekQA1.0 dataset

Set	EM (%)	F1 (%)
Development Set	79.4	91.9
Test Set	81.1	93.3

7. CONCLUSIONS AND FUTURE WORK

In this thesis, we created a Greek Question Answering dataset following the proposed methodologies and standards of SQuAD [6] and other related works with a similar objective. Therefore, we presented the Greek Question Answering (GreekQA) dataset, a Greek reading comprehension dataset based on Wikipedia articles. GreekQA1.0 dataset consists of 1,000+ questions posed by crowdworkers on curated passages from a set of Wikipedia articles in Greek. For the development of the GreekQA dataset, we also introduced the namesake GreekQA Crowdsourcing Annotation Platform, a web application designed and implemented for crowdsourcing the collection of question and answer pairs for this dataset. We analyzed the requirements and the selected technologies of the GreekQA crowdsourcing platform, described the structure of the implementation, and demonstrated the platform. Then, we described the procedure of curating passages and the defined guidelines for collecting question-answer pairs. In order to comprehend the properties of the GreekQA1.0, we analyzed the questions and answers as well as the reasoning required to answer the questions based on the corresponding passage. Through this analysis, the variety of answers, questions, and reasoning was demonstrated, and similarities with SQuAD1.1 were noticed. Finally, we evaluated the human performance of GreekQA1.0, reaching an Exact Match of 79.4%, F1-score of 91.9% on the development set, and an Exact Match of 81.1%, F1-score of 93.3% on the test set.

The higher human performance scores than standard datasets indicate the need for two additional answers to reduce possible bias instead of the current one in the dataset. Therefore, all questions in the development and test sets need to be enhanced with one more additional answer. Moreover, the total dataset size should be expanded from 20 to 100 times its current size, following the standard sizes of related datasets, in order to be utilized properly by language models. With the possible arrival of larger GreekQA datasets, further dataset analysis and specific comparison with other standard datasets would assist in comprehending the altered and improved properties of these versions. Hence, the Human Performance should also be remeasured along with the conducting experiments on monolingual Greek (GreekBERT) and multilingual (XLM-RoBERTa, multilingual-BERT) language models. More specifically, these models can be fine-tuned, tested on GreekQA, and evaluated compared to the Human Performance Baseline of each set. Another interesting set of experiments could be cross-lingual reading comprehension with a zero-shot learning approach. During these experiments, multilingual models are fine-tuned using the training set of a language-specific dataset and evaluated on the development test of another language dataset. For example, a multilingual model can be fine-tuned with English SQuAD and evaluated on GreekQA or vice versa. Lastly, the dataset could be extended further with the addition of unanswerable questions, as demonstrated by SQuAD2.0 [17] and FQuAD2.0 [4].

To sum up, the GreekQA1.0 dataset is our first attempt to fill the gap of a Greek native language annotated dataset focused on QA. It meets all the procedural standards; however, it lacks result-based standards due to the small size of the dataset. Therefore, it is a promising initiative with significant potential for the future. In other words, this first version of the dataset sets the basis for extended versions, which will be capable of fostering and be pertinent to impressive progress on Greek and multilingual QA language models, as done similarly by other datasets of languages other than English. Therefore, there is a plethora of future work to be done outside the time constraints of this undergraduate thesis which has fully met our expectations.

ABBREVIATIONS - ACRONYMS

DL	Deep Learning
NLP	Natural Language Processing
RC	Reading Comprehension
MRC	Machine Reading Comprehension
QA	Question Answering
SQuAD	Stanford Question Answering Dataset
FQuAD	French Question Answering Dataset
KorQuAD	Korean Question Answering Dataset
JaQuAD	Japanese Question Answering Dataset
EM	Exact Match
UI	User Interface
JS	JavaScript
DOM	Document Object Model
WWW	World Wide Web

BIBLIOGRAPHY

[1] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Comput. Intell. Mag.*, 13(3):55–75, 2018.

[2] Zheqian Chen, Rongqin Yang, Bin Cao, Zhou Zhao, Deng Cai, and Xiaofei He. Smarnet: Teaching machines to read and comprehend like human. *CoRR*, abs/1710.02772, 2017.

[3] Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha C. Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. Overview and importance of data quality for machine learning tasks. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3561–3562. ACM, 2020.

[4] Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. Fquad: French question answering dataset. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1193–1208. Association for Computational Linguistics, 2020.

[5] Seungyoung Lim, Myungji Kim, and Jooyoul Lee. Korquad1.0: Korean QA dataset for machine reading comprehension. *CoRR*, abs/1909.07005, 2019.

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[7] Ralph Grishman. *Computational linguistics: an introduction*. Cambridge University Press, 1986.

[8] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

[9] Yoav Goldberg. *Neural Network Methods for Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2017.

[10] Elizabeth D Liddy. Natural language processing. 2001. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.

[11] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 2001.

[12] Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Networks Learn. Syst.*, 32(2):604–624, 2021.

[13] Shanshan Liu, Xin Zhang, Sheng Zhang, Hui Wang, and Weiming Zhang. Neural machine reading comprehension: Methods and trends. *CoRR*, abs/1907.01118, 2019.

[14] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. Text classification algorithms: A survey. *Inf.*, 10(4):150, 2019.

[15] Andrew M. Dai and Quoc V. Le. Semi-supervised sequence learning. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3079–3087, 2015.

[16] Jim Conallen. Modeling web application architectures with UML. *Commun. ACM*, 42(10):63–70, 1999.

[17] Elior Sulem, Jamaal Hay, and Dan Roth. Do we know what we don't know? studying unanswerable questions beyond squad 2.0. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 4543–4548. Association for Computational Linguistics, 2021.

[18] ByungHoon So, Kyuhong Byun, Kyungwon Kang, and Seongjin Cho. Jaquad: Japanese question answering dataset for machine reading comprehension. *CoRR*, abs/2202.01764, 2022.

[19] Quentin Heinrich, Gautier Viaud, and Wacim Belblidia. Fquad2.0: French question answering and learning when you don't know. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 2205–2214. European Language Resources Association, 2022.

[20] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics, 2017.

[21] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In Phil Blunsom, Antoine Bordes, Kyunghyun Cho, Shay B. Cohen, Chris Dyer, Edward Grefenstette, Karl Moritz Hermann, Laura Rimell, Jason Weston, and Scott Yih, editors, *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 191–200. Association for Computational Linguistics, 2017.

[22] Patrick S. H. Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: evaluating cross-lingual extractive question answering. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7315–7330. Association for Computational Linguistics, 2020.

[23] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online, July 2020. Association for Computational Linguistics.

[24] Snehal (Neil) Gaikwad, Durim Morina, Rohit Nistala, Megha Agarwal, Alison Cossette, Radhika Bhanu, Saiph Savage, Vishwajeet Narwal, Karan Rajpal, Jeff Regino, Aditi Mithal, Adam Ginzberg, Aditi Nath, Karolina R. Ziulkoski, Trygve Cossette, Dilrukshi Gamage, Angela Richmond-Fuller, Ryo Suzuki, Jeerel Herrejón, Kevin Le, Claudia Flores-Saviaga, Haritha Thilakarathne, Kajal Gupta, William Dai, Ankita Sastry, Shirish Goyal, Thejan Rajapakshe, Niki Abolhassani, Angela Xie, Abigail Reyes, Surabhi Ingle, Verónica Jaramillo, Martin Godínez, Walter Ángel, Carlos Toxtli, Juan Flores, Asmita Gupta, Vineet Sethia, Diana Padilla, Kristy Milland, Kristiono Setyadi, Nuwan Wajirasena, Muthitha Batagoda, Rolando Cruz, James Damon, Divya Nekkanti, Tejas Sarma, Mohamed Saleh, Gabriela Gongora-Svartzman, Soroosh Bateni, Gema Toledo Barrera, Alex Peña, Ryan Compton, Deen Aariff, Luis Palacios, Manuela Paula Ritter, Nisha K. K., Alan C. Kay, Jana Uhrmeister, Srivalli Nistala, Milad Esfahani, Elsa Bakiu, Christopher Diemert, Luca Matsumoto, Manik Singh, Krupa Patel, Ranjay Krishna, Geza Kovacs, Rajan Vaish, and Michael S. Bernstein. Daemo: A self-governed crowdsourcing marketplace. In Celine Latulipe, Bjoern Hartmann, and Tovi Grossman, editors, *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST 2015 Adjunct Volume, Charlotte, NC, USA, November 8-11, 2015*, pages 101–102. ACM, 2015.

[25] Ruth Malan, Dana Bredemeyer, et al. Functional requirements and use cases. *Bredemeyer Consulting*, 2001.

[26] Grady Booch, Ivar Jacobson, James Rumbaugh, et al. The unified modeling language. *Unix Review*, 14(13):5, 1996.

[27] Martin Glinz. On non-functional requirements. In *15th IEEE International Requirements Engineering Conference, RE 2007, October 15-19th, 2007, New Delhi, India*, pages 21–26. IEEE Computer Society, 2007.

[28] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[29] Paolo Boldi, Massimo Santini, and Sebastiano Vigna. Pagerank as a function of the damping factor. In Allan Ellis and Tatsuya Hagino, editors, *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, pages 557–566. ACM, 2005.

[30] Eleni Partalidou, Eleftherios Spyromitros Xioufis, Stavros Doropoulos, Stavros Vologiannidis, and Konstantinos I. Diamantaras. Design and implementation of an open source greek POS tagger and entity recognizer using spacy. In Payam M. Barnaghi, Georg Gottlob, Yannis Manolopoulos, Theodoros

Tzouramanis, and Athena Vakali, editors, *2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019, Thessaloniki, Greece, October 14-17, 2019*, pages 337–341. ACM, 2019.