Journal of Information Systems and Telecommunication
Vol.12, No.2, April-June 2024, 138-151
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Designing a Semi-Intelligent Crawler for Creating a Persian Question Answering Corpus Called Popfa

Hadi Sharifian[1] , Nasim Tohidi[2] , Chitra Dadkhah[2*]

[1].K. N. Toosi University of Technology, e-learning center, Tehran, Iran
[2].Artificial Intelligence Department, Computer Engineering Faculty, K. N. Toosi University of Technology, Tehran, Iran

Received: 28 Jan 2023/ Revised: 04 Oct 2023/ Accepted: 12 Nov 2023

Abstract

Question answering in natural language processing is an interesting field for researchers to examine their ability in solving the tough Alan Turing test. Everyday computer scientists are trying hard to develop and promote question answering systems in various natural languages, especially English. However, in Persian, it is not easy to advance these systems. The main problem is related to low resources and not enough corpora in this language. Thus, in this paper, a Persian question answering text corpus is created, which covers a wide range of religious, midwifery, and issues related to youth marriage topics and question types commonly encountered in Persian language usage. In this regard, the most important challenge was introducing a method for data gathering in Persian as well as facilitating and expanding the data gathering process. Though, SIC (Semi-Intelligent Crawler) is proposed as a solution that can overcome the challenge and find a way to crawl the Persian websites, gather text and finally import it to a database. The outcome of this research is a corpus called Popfa, which stands for Porsesh Pasokh (question answering) in Farsi. This corpus contains more than 53,000 standard questions and answers. Besides, it has been evaluated with standard approaches. All the questions in Popfa are answered by specialists in two general topics: religious and medical questions. Therefore, researchers can now use this corpus for doing research on Persian question answering.

Keywords: Question Answering, Persian Corpus, Religious Questions, Medical Questions, Natural Language Processing.

1- Introduction

It has been many years since Alan Turing introduced his famous experiment, and despite all the advances that have taken place in the world of computer science, no computer or even supercomputer has yet successfully passed the Turing test completely. This simple experiment is an artificial intelligence that communicates with a human through a computer user interface and convinces the human that he is communicating with a human. [1]

Designing and implementing an efficient question answering system, which can provide the accurate answer to the user input question in natural language in the shortest time, is one of the most attractive and practical problems in the field of artificial intelligence for computer scientists and researchers as well as managers of companies providing computer technology services such as the production of programs, websites, speech bots, etc. In fact, this question answering system can be a model of the same machine that is supposed to pass the Turing test successfully. There have been astonishing advances in English in this area with scientists achieving successful results in terms of the Turing experiment, but in Persian, despite several tools which have been proposed in recent years [2-6], research on question answering datasets has not progressed significantly [7]. One of the reasons for the abandonment of the ancient and rich Persian language in this category is the inadequacy of a comprehensive and powerful corpus of valid questions and answers. Currently, to the best of our knowledge, the largest Persian question answering corpus was far much smaller than the similar one for other languages like English.

Persian (Farsi) language has many attributes that make it distinct from other well-studied languages. In terms of script, Persian is similar to Semitic languages, like Arabic and Amharic. Linguistically, however, Persian is an Indo-European language [8,9] and thus distantly related to most of the languages of Europe as well as the northern part of

✉ **Corresponding Author**
dadkhah@kntu.ac.ir

the Indian subcontinent. Therefore, the Persian language has features that distinguish it from the English language and make its processing more complex. For example, in Persian, some letters stick to each other, and in addition to the space between words, the half-space is also used in writing Persian text. In addition, the structure of sentences and the way in which words with different roles are placed in Persian sentences is not the same as in English sentences. Therefore, the methods introduced for English cannot be used for Persian. Another important point is that many of the texts and data available are not written in the formal language, for example, the half-space is not observed all the time. These are some of the reasons why the number of research done on the Persian language is very small compared to the English language.

Today, Google is known as an intelligent search engine. However, this powerful search engine returns many links for each incoming question that are not necessarily the intended result. Therefore, users must open links and check the content of each one to finally find the answer among a large number of returned links. This is where having a question answering system in Persian that receives a question and provides an accurate answer seems necessary. In this regard, the purpose of this paper is to create a question answering corpus in the Persian language.

The structure of the paper is as follows: In Section 2, some previous works in this field are introduced. Then, Section 3 explains the proposed method. Section 4 contains the steps for implementing our proposed strategy. Sections 5 and 6 describe evaluations and experimental results, respectively. More, section 7 presents the prepared user interface and section 8 gives a discussion about the unique feature of the corpus. Finally, in Section 9 conclusion and future works are summarized.

2- Related Works

The number of available systems for Persian language processing is very small compared to the English language, which has led to a decrease in the research on Persian language in the field of natural language processing [5,3]. There is a lack of standard systems in the field of Persian language question answering systems, as one of the applications of language processing.

Since 1999, TREC (Text Retrieval Conference) has had a question answering track [10] resulting in high accuracy systems for English, like methods in [11] and [12].

In some related papers, researchers have decided to either apply community-sourced datasets or develop restricted-domain question answering systems. For instance, in [5] the Rasekhoon[1] question answering dataset was used to evaluate a question matching model in Persian. Plus, in [13], TriviaQA was presented, which was a reading comprehension dataset including question-answer-evidence triples. It contained question-answer pairs written by trivia enthusiasts and gathered evidence documents (on average 6 per question) which provided distant supervision for answering the questions.

The main activities in the field of Persian question answering systems, like [7], [14], and [15], have focused on approaches based on the feature that the question raised in Persian can be analyzed from a syntactic and semantic perspective, and the most appropriate answer can be selected based on the available database.

In [10], authors introduced a standard Persian text collection, named Hamshahri, which was built from a large number of newspaper articles according to TREC specifications in which statistical information about documents, queries, and their relevance judgment were presented. This collection can be downloaded as a package from its website. The package contains all relevant judgments for the 65 standard topics, some descriptions of previous research conducted based on the collection, and some source codes for indexing and retrieval of the collection [16].

In [17], sentences were classified into two levels of coarse and fine classes based on the type of answer to each question. After extracting features and setting a sliding window on the Conditional Random Fields (CRF) model, CRF Question Classifier (QC) was trained to predict labels for every token in question. Then, a majority voting on the question classification output was used to extract a unique label for each question, and the effects of features on the ultimate accuracy of the system were evaluated.

Also, in [18], they proposed an approach that was used in an online automatic question answering system. They combined rule-based and machine learning question classification approaches for highly inflectional languages such as Persian. They got satisfactory results according to the high number of question classes.

In [19], a cross-lingual approach using a unified semantic space among languages was introduced. In this study, after keyword extraction, entity linking, and answer type detection, cross lingual semantic similarity was used to extract the answer from the knowledge base via relation selection and type matching.

In [20], a corpus for the Persian language was presented. This corpus consists of 2,118 non-factoid and 2,051 factoid questions and for each question, question text, question type, question difficulty from the questioner and responder perspective, expected answer type in coarse-grained and fine-grained level, the exact answer, and page and paragraph number of answer are annotated. This corpus can be applied to learn components of a question answering system, including question classification, information retrieval, and answer extraction. This corpus is freely available for academic purposes.

[1] www.rasekhoon.net

In [15], a medical question answering system for the Persian language was proposed. In their research, a dataset of diseases and drugs was collected and structured. The system included three main modules: question processing, document retrieval, and answer extraction. For the question processing module, a sequential architecture was designed which retrieved the main concept of a question by using different components. In these components, rule-based methods, natural language processing, and dictionary-based techniques were used. In the document retrieval module, the documents were indexed and searched using the Lucene library. The retrieved documents were ranked using similarity detection algorithms and the highest-ranked document was selected to be used by the answer extraction module. This module was responsible for extracting the most relevant section of the text in the retrieved document.

In [21], PeCoQ was defined which was a Persian question answering dataset. It included 10K questions and answers extracted from the Persian knowledge graph, FarsBase. Additionally, for each question, the SPARQL query and 2 paraphrases authored by linguists were provided. There were various complexity types in this dataset, like multi-relation, multi-entity, comparative, superlative, aggregation, ordinal, and temporal constraints.

In [22], a Persian Question Answering Dataset (ParSQuAD) was generated based on translating the SQuAD 2.0 dataset by machine. Through it, some errors have been detected within the process of translation; resulting in two different versions of it, depending on whether these errors have been corrected automatically or manually. The most important weakness of this dataset is that it does not have the quality of a native Persian reading comprehension dataset containing native question and answer samples annotated by multiple human annotators.

In [23], PersianQuAD was introduced which was a native question answering dataset for the Persian language. The authors built this dataset in 4 phases: 1) Wikipedia article selection, 2) question-answer collection, 3) three-candidate test set preparation, and 4) Data Quality Monitoring. The output dataset contained about 20K questions and answers made by native annotators on a set of Persian Wikipedia articles. The answer to each question was a segment of the corresponding article text. According to their report, PersianQuAD consisted of questions of different types and complexities. Plus, they proposed 3 versions of a deep learning-based question answering system trained using MBERT, ALBERT-FA, and ParsBERT on PersianQuAD, and for MBERT they achieved the best result.

Finally, in [24], authors proposed PQuAD, a crowdsourced reading comprehension dataset for Persian on Wikipedia articles which included various subjects. Its data collection process had 3 phases: 1) passage curation, 2) question-answer pair annotation, and 3) additional answer collection. The output dataset consisted of 80K questions and their answers. They evaluated different properties of the dataset to depict its diversity and complexity as a machine reading comprehension benchmark.

Considering all the mentioned efforts and information, proposing a feasible and accurate method for gathering questions and answers in Persian in a corpus to be used for training question answering systems in the future is crucial. The created corpus in this paper serves as a dataset that can be utilized to train and improve the performance of Persian question answering systems. These systems can leverage machine learning techniques, such as deep learning algorithms, to learn from the provided data and enhance their question answering abilities. Hence, in the following the proposed method for creating the corpus is explained.

3- The Proposed Method

Research conducted in the field of question answering systems shows the shortage of a standard Persian question answering corpus [7]. Naturally, scientists face various limitations and challenges in this area. Considering the incoherence of information related to Persian question answering, there could be two solutions:

- Solution 1: Generating basic questions and answers, followed by a Persian question answering system.
- Solution 2: Collecting questions and answers available on global websites in Persian and editing them to produce a Persian question answering corpus.

3-1- Challenges

As a creative and new way, a cost-benefit table was formed to select the logically desired solution. The most important selection features were:

1) Time taken
2) Research and executive costs
3) Required human resources
4) Technical and structural limitations
5) Verification capability
6) The size of the database
7) Domain comprehensiveness

Since these features were selected by a group without any previous background and are among the innovations of this paper, there was no reported quantitative data. Therefore, we decided to compare the features using two strategies of collection and production. In other words, what is the relationship between them regardless of the quantity of each one. The value comparison has been summarized as Low, Medium, and High cost.

The evaluation results of the above-mentioned features are shown in Table 1. Production strategy as well as fact-checking was preferred for technical and structural features, but it was significantly different from the collection strategy for other features. Of course, due to the pristine nature of this area, the execution time for either

solution was not clear, and we could only compare the time between the two solutions.
The scientific method of cost-benefit analysis in our proposed approach resulted in -10 for the collection method and -17 for the production method. Therefore, the selected method for producing the Persian question answering corpus was based on the collection strategy.

Table 1: Results of the cost-benefit method

Features	***Collection solution***	***Production solution***
Doing time	-3	-1
Research and executive costs	-3	-1
Required human resources	-3	-1
Technical and structural limitations	-1	-3
Verification capability	-1	-2
The size of the database	-3	-1
Domain comprehensiveness	-3	-1
Result	**-17**	**-10**

3-1-1- Identifying Reliable Websites

The next step after choosing solution 2 was to identify reliable websites that include a significant number of Persian questions and answers. A number of these websites were found by searching and the extracted sites were selected through two refinement stages and entered the final phase of corpus production. The first step was to refer to search engines to find websites having question and answer banks, and the second step was to look at the criteria for choosing the right website to crawl. The most important of these criteria are:

1) Intellectual property rights
2) Acceptable quantity
3) The possibility of crawling on the website
4) Random fact-checking of answers in domains
5) The comprehensiveness of the domain
6) Website ranking in Alexa[1]

The websites that entered the final phase, based on the above criteria, have been listed in Table 2.

Table 2: Persian websites suitable for crawling

Name	***#Q&As***	***Domain***	***Rank***
Hawzah	2371	Hawzah.net	460
Mamai	35609	Mamasite.ir	1119
Rasekhoon	83364	Rasekhoon.net	198
Shahab Moradi	7262	Shahab-moradi.ir	27747

3-2- Crawling Strategies

After selecting acceptable websites, the building process analysis was started with the aim of selecting the best way to extract questions and provide answers on each website. As observed in Fig. 1, the selection in this phase consists of four steps:

1) Manual extraction
   Considering the goal of producing a Persian question answering corpus with at least 50,000 records, this method was practically erosive and non-optimal and was removed from the selected strategies at the very beginning of the work.
2) Using ready-made tools
   Since the ready-made tools have limitations to crawl in all available websites, their use did not lead to the desired result. The main problem of these tools is data redundancy. Hence, this method was also rejected.
3) Production of a fully intelligent crawler robot
   At first glance, it seems like an attractive solution, however, the production of this robot may be a much more difficult project than the production of a Persian question answering corpus. Therefore, considering technical challenges, it is not possible to use this method.
4) Using an intermediate method (semi-intelligent)
   The only remaining option was this method which was selected and applied in this paper.

3-2-1- Semi-Intelligence Crawler (SIC)

As a new and unprecedented method, a crawler that is not a fully intelligent robot but can extract the materials required from the website using the human primary guide is designed. SIC is a revision crawler, whose primary setups for each website are easily carried out by a program, and performs the rest of the crawling steps, extracting and inserting into the database, in a completely intelligent way. Since the number of selected websites for producing Persian question answering systems was very small, performing the primary setting of the crawler was not a serious challenge, because it was important to configure it from any website.
To have a crawler that extracts exactly the desired information, it was necessary to examine each website separately with common methods, mostly innovative and sometimes combined methods. What is performed at this step is as follows:

1) View and check the page text
   Each page of the website that is loaded in the browser contains invisible information that can only be seen in the source text of that page. Information such as Document Type, Alphabetical coding, Metadata, Scripts used, Layout settings, File events, Local functions called, etc.
2) Parse Tree
   One of the most valuable ways to get the settings for each website is to rearrange the parse tree of different files on a website and discover the legal relationship between them.

Fig. 2 shows the metadata of the file. The useful part of designing SIC is the knowledge of the files that this page uses. These files are often used for either graphical settings of the page and have the suffix CSS (Cascading Style Sheets) or user-side functions that, if necessary,

[1] www.alexa.com/siteinfo

communicate with the server-side functions and are observed with the suffix JS (Java Script).

In Web 2.0 and later programming generations, an attempt is made to send fewer requests to the server, and with the minimum need to reload the entire page, the user requests are preprocessed on the user side by AJAX (Asynchronous JavaScript and XML) technology-based functions and then, sent to the server. By observing these functions and examining their structure, we were able to design a more powerful SIC website. In Fig. 3, these local functions are outlined.

In structured websites, fetching information from the order in their results' structure is a more efficient SIC design. Here, we sought to maximize the use of the order in the file structure. Fig. 4 shows a part of the parse tree.

3) Use of the web browser developer environments

The actions and reactions between the user system and the server on which the website is located are managed through browsers and are usually hidden from the ordinary user. To be aware of these interactions and behind-the-scenes events, you need to enter the web developer environment in the relevant browser.

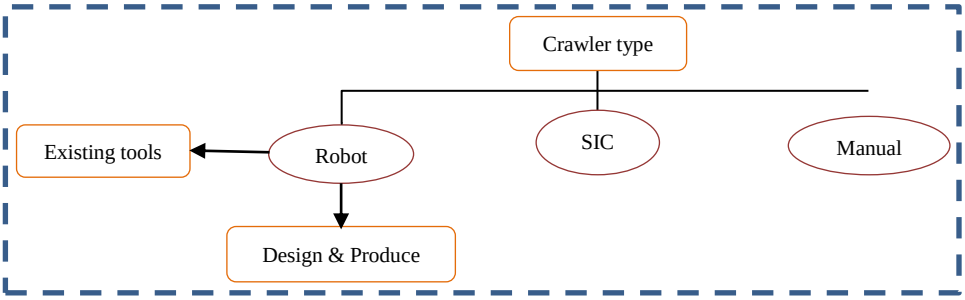

Fig. 1 Selection steps

```
<!DOCTYPE html>
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="Author" content="Hadi Sharifian(09121869975)" />
  <meta name="Copyright" content="Copyright © 2018 Shimanaa, All Rights Reserved" />
  <meta name="robots" content="index, follow" />
  <meta name="keywords" content="شیمانا، آموزش، تکنولوژی، پیپاد سمپاش، موتلی کوپتر، هوش مصنوعی"/>
  <title>پیپاد سپرتگ</title>
  <!-- Tell the browser to be responsive to screen width -->
  <meta content="width=device-width, initial-scale=1, maximum-scale=1, maximum-scale=3.0, minimum-scale=0.86" name="viewport">
  <!-- Bootstrap 3.3.7 -->
  <link rel="stylesheet" href="dist/css/bootstrap-theme.css">
  <!-- Bootstrap rtl -->
  <link rel="stylesheet" href="dist/css/rtl.css">
  <link rel="stylesheet" href="dist/css/quiz.css">
  <!-- Font Awesome -->
  <link rel="stylesheet" href="bower_components/font-awesome/css/font-awesome.min.css">
  <!-- Ionicons -->
  <link rel="stylesheet" href="bower_components/Ionicons/css/ionicons.min.css">
  <!-- Theme style -->
  <link rel="stylesheet" href="dist/css/AdminLTE.css">
  <!-- AdminLTE Skins. Choose a skin from the css/skins
       folder instead of downloading all of them to reduce the load. -->
  <link rel="stylesheet" href="dist/css/skins/_all-skins.min.css">

  <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
  <!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
  <![endif]-->
  <!-- jQuery 3 -->
  <script src="bower_components/jquery/dist/jquery.min.js"></script>
  <!-- Bootstrap 3.3.7 -->
  <script src="bower_components/bootstrap/dist/js/bootstrap.min.js"></script>
  <!-- SlimScroll -->
```

Fig. 2 Source text of a web page

```
<script>
    $('#addedtopics').hide();
    $('select').each(function(){
        ajaxlink="ajax/form_load_select.php?table="+$(this).attr("id");
        $(this).load(ajaxlink,function(){if($('#sections option').length>0&&$('#courses option').length>0&&$('#majors option').length>0&&$('#lessons
option').length>0)$('.modal-primary').hide();});
        $(this).css("width","95%");
    });
    $('select').bind('change',function(event){
        if($(this).attr("data-zir")!=""){
            $('.modal-primary').addClass('in');
            $('.modal-primary').css("display","block");
            tar=$(this).attr("data-zir");
            ajaxlink="ajax/form_load_zir.php?table="+$(this).attr("id")+"&zir="+$(this).attr("data-zir")+"&value="+$(this).val();
            $('#'+tar).load(ajaxlink,function(){$('.modal-primary').hide();});
            //$('.modal-primary').hide();
            //window.location='squiz#'+tar;
        }
    });
    $(document).on('change','#topics',function(){
        pm=$('#sections option:selected').text()+'-'+$('#courses option:selected').text()+'-'+$('#lessons option:selected').text()+'-'+$('#topics opt
        pm+='<br><br>'+'این مبحث به آزمونک شما اضافه شود؟'+'<br><br>'+'<button type="button" class="btn btn-block btn-success">بله! اضافه شود</button>'+'<butto
block btn-danger">نه، منصرف شدم</button>';
        $('.modal-info .modal-body p').html(pm);
        $('.modal-info').addClass('in');
        $('.modal-info').css("display","block");

    });
    $(document).on('click','.btn-success',function(){
        txt='<li>'+$("#sections option:selected').text()+'-'+$('#courses option:selected').text()+'-'+$('#lessons option:selected').text()+'-'+$('#to
class="fa fa-fw fa-trash-o" id="'+$('#topics option:selected').val()+'"></i>'+'</li>';
        $('ol').append(txt);
        $('#topics option:selected').remove();
        $('.modal-info').hide();
        $('#addedtopics').show();

    });
    $('.modal-info').on('click','.btn-danger',function(){
        $('.modal-info').hide();
    });
    $('ol').on('click','.fa-trash-o',function(){
```

Fig. 3 Local functions called

```
<div class="attachment-block clearfix">
  <img class="attachment-img" src="images/arts/1589383615sirang-logo.jpg" alt="تماس با سیرنگ">
  <div class="attachment-pushed">
      <a href="9&تماس با سیرنگ" href="_blank"  title="تماس با سیرنگ">
      <h4 class="attachment-heading">تماس با سیرنگ</h4>
      </a>
      <div class="attachment-text">
          سیرنگ راه های ارتباطی متنوعی در اختیار شما قرار می دهد. فراخور ضرورت و نیاز خود می توانید یکی از آنها را انتخاب کنید.
  </div>
</div>
```

Fig. 4 A part of the parse tree

provides useful information to developers, such as:

- Details of each file
- The loading time of each element
- The protocol used to send the request to the server
- Return the data type of each request
- Variables exchanged between browser and server
- Displaying possible user side errors
- How to call the file updater function

Fig. 5 shows a part of the web developer environment in the Chrome browser (Google).

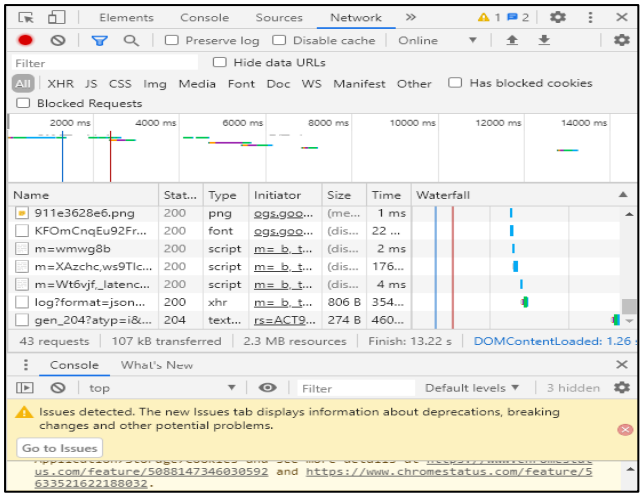

Fig. 5 A part of the web developer environment

4- SIC Strategy Implementation

In terms of implementation, to crawl the websites that we could extract the target of the project, the server-side programming language of PHP as well as the user-side programming languages of JS, jquery, and CSS were exploited. Plus, for inserting the extracted information into the crawls, MySQL was selected.

In the first crawl on the *.net* domain website, we faced an obstacle called Same-Origin Policy (SOP)[1]. SOP is a web security policy that does not allow web pages to load content from other domains, especially user-side scripts such as JS and jquery.

In fact, if a page in the *example1.com* domain tries to fetch content from the *example2.com* domain through one of the loads, post, get methods, or other common methods in various programming languages, browsers will be blocked according to agreed standards. The SIC crawler produced in the proposed approach was implemented in the hosting space of *shimanaa.com* and according to its mission, it was tasked to fetch and load the content of the question answering pages of the websites listed in Table 2 in this domain.

The next serious challenge which was tackled in this paper was facing this security policy.

Today, all websites have local scripts that are written primarily for the specific needs of the website. These scripts are stored in the web hosting space and addressing has been relatively defined in the context of their programs. For example, in the following code:

```
<script src=”bower_components/fastclick/lib/fastclick.js”></script>
```

As observed, the calling address of the *fastclick.js* file has been locally defined in the path of *bower_components/fastclick/lib/fastclick.js*.

Supposing there is one/more of these files on each website, they are necessary for the programs to run properly. However, SIC did not access any of these files on its host. Accordingly, there were two solutions to this problem:

1) Downloading all the required files of the websites and placing them in the same URLs.
2) Making changes to the fetched content so that the need for those files is eliminated.

Both solutions had some difficulties. The first one seemed to be a tedious task that required lots of time and operator work. The second one required an approach to pass the functions and procedures required by the websites.

Since our vision for the future was to carry out our approach on a larger scale, as well as to use the corpus produced in an operational plan, we selected the more difficult path and overcame this obstacle safely by inventing new methods.

According to W3C (World Wide Web Consortium) [2] standardization, every element in a web file must follow a set of rules. If this is not achieved, either the file upload process will be disrupted, or the functions and procedures will be called.

For example, the following code has a structural problem:

```
<li><a href=”profile_2_admin” target=_self ><i class=”fa fa-circle-o”></i>مشخصات کاربری.</li>
```

Because the tag of <a> had to be closed, which was not. The correct form of the above code is as follows:

```
<li><a href=”profile_2_admin” target=_self ><i class=”fa fa-circle-o”></i>مشخصات کاربری</a></li>
```

There were so many such cases that sometimes they seriously disrupted the SIC crawling process. To solve this problem, we made changes in the SIC to fetch as few elements as possible to make it easier to diagnose and fix structural defects.

Simultaneously with the successful crawling and fetching steps, the database entry step also should be performed. Hence, these three steps were implemented in two phases.

4-1- Phase 1: Crawling

In this step, a table was also needed to store information related to the question and answers rich pages. For this purpose, in the MySQL project database to store the information collected by SIC, 2 tables were created with the names *crl_links* and *crl_links_cats*. The *crl_links_cats* table stores classifications of questions and answers. The columns and some parts of the crawled records in this table are shown in Table 3.

Table 3: A part of the crl_links_cats table

cat_id	***subcat***	***Title***
1	2473	قرآن و تفسیر
2	2513	عقاید
3	2525	احکام
4	6330	مهدویت و انتظار
5	6699	تاریخ
6	2623	فرق، ادیان و مذاهب

The *crl_links* table stores the URLs of web pages containing questions and answers. The columns and some of the crawled records in this table are shown in Table 4. In Tables 3 and 4, *subcat* refers to the thematic classification code of each question and its reference website which has a unique code, shown in Table 4.

Table 4: A part of the crl_links table

link_id	***qid***	***subcat***	***Title***
1	1079865	2473	راه رهایی از دنیا دوستی چیست
2	723806	2473	مراد از فراز شریف الله مولی الذین آمنو…

[1] http://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy
[2] https://www.w3.org/standards/

3	503363	2473	آیا نمونه‌هایی وجود دارد که نشان دهنده تاثیر فصاحت...
4	1079864	2473	چرا نماز باید به زبان عربی خوانده شود
9	1079847	2473	اگر اعمال دنیوی ما مربوط به تصمیمان در عالم ذر....
24	1079733	2473	معنای اسلام چیست
25	1079732	2473	در زندگی‌ام همیشه با مشکلات و گرفتاری روبه‌رو هستم...

4-2- Phase 2: Insert Fetch

At this point, the SIC crawler, based on the information in the *crl_links* table, scanned the page of each address in this table and entered the information for each question and answer in the *crl_questions* table. The columns and some of the crawled records in this table are shown in Table 5.

5- Evaluation

After the production of the corpus, the only thing left to do was to standardize it as shown in Fig. 6.
The indicators considered are:

- The text of the questions and answers should be free of any HTML protocol tags and symbols.
- Having no nature other than questions and answers.
- The data should not be duplicated.
- The addresses must be valid for fact-checking.
- Classifications should not be too general or partial.

The pre-standardization corpus contained 83, 364 questions and answers. At this step, duplicate questions were removed first. Then, in a separate table, we saved a copy of the corpus after deleting the stop word. We received the list of stop words from this link.
After processing all the questions and answers that were in the draft corpus, we proceeded to extract all the unique words. 56,925 of these words were stored in a separate table, in which, 4 parameters were calculated for each word and the table was updated. In this regard, we used the four metrics from Eq. (1) to Eq. (4), [25].

$$\text{TF}(t,d) = \frac{\text{Number of times term t appears in a document}}{\text{Total number of terms in the document}} \quad (1)$$

$$\text{DF}(t) = \frac{\text{Number of documents that the term appears in them}}{\text{Total number of documents}} \quad (2)$$

$$\text{IDF}(i) = \frac{\ln(N+1)}{\text{DF}(i)} + 1 \quad (3)$$

Where N is the total number of documents in the collection.

$$\text{TF-IDF}(i,d) = \text{TF}(i,d) \times \text{IDF}(i) \quad (4)$$

The corpus includes rather huge amounts of information in question-and-answer form; therefore, suitable metrics for evaluation, are those used in information retrieval. There are several of them, but order-aware metrics are chosen in this paper as follows (Eq. (5) to Eq. (10)) [10, 26, 27, 11]:

a. Mean Reciprocal Rank (MRR)

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{r_i} \quad (5)$$

Where |Q| denotes the total number of queries and r_i shows the rank of the *i*th relevant result.

b. Average Precision (AP)

$$AP = \frac{\sum_{i=1}^{n}\left(P(i)\times rel(i)\right)}{number\ of\ relevant\ items} \quad (6)$$

Where *P(i)* is the Precision@k metric and *rel(i)* is 1 if the *i*th item is relevant otherwise is 0.

c. Mean Average Precision(MAP)

$$MAP = \frac{1}{|Q|}\sum_{i=1}^{|Q|}AP(i) \quad (7)$$

Where |Q| denotes the total number of queries and *AP(i)* is the average precision for the *i*th query.

d. Cumulative Gain (CG@k)

$$CG@k = \sum_{i=1}^{k} rel_i \quad (8)$$

e. Discounted Cumulative Gain (DCG@k)

$$DCG@k = \sum_{i=1}^{k}\frac{rel_i}{\log_2(i+1)} \quad (9)$$

f. Normalized Discounted Cumulative Gain (NDCG@k)

$$NDCG@k = \frac{DCG@k}{IDCG@k} \quad (10)$$

Where *IDCG@k* is the ideal *DCG@k* (more relevant item comes first).
Metrics *CG@k*, *DCG@k*, and *NDCG@k* consider the grade of relevancy while the first three metrics do not mention it.

Table 5: Part of the crl_questions table

id	*question*	*answer*	*keyword*
1	راه رهایی از دنیا دوستی چیست	اجمالی: دنیا مونث....	رهایی از دنیادوستی | دنیادوستی | قرآن مجید | زندگی آخر ...
2	مراد از فراز شریف الله مولی الذین آمنو...	آیه شریف "ذلک بان....	مولا | مومنان | کافران
3	آیا نمونه‌هایی وجود دارد که نشان دهنده تاثیر فصاحت...	آری، نمونه‌های فراوانی در این...	
4	چرا نماز باید به زبان عربی خوانده شود	برای روشن شدن پاسخ، ابتدا...	نماز | حقوق و احکام | زبان عربی | نماز با زبان عربی
9	اگر اعمال دنیوی ما مربوط به تصمیمان در عالم ذر....	شرح در مورد مسئله...	عالم ذر | دنیا | اعمال دنیوی | تفسیر | جبر یا اختیار | عدالت...
24	معنای اسلام چیست	پاسخ اسلام در لغت...	معنای اسلام | اسلام | دین اسلام | قرآن کریم | توحید کامل
25	در زندگی‌ام همیشه با مشکلات و گرفتاری روبه‌رو هستم...	شرح در کاری که از....	ضرر و بیماری | امتحان | عدالت| پروردگار | ابتلا و امتحان...

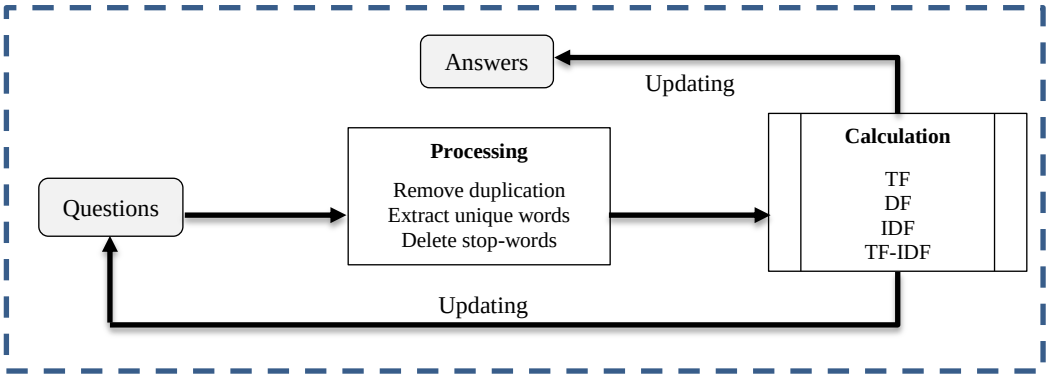

Fig. 6 Schematic of corpus standardization steps

To evaluate the corpus, we ran a list of questions written by a user interface in PHP in the MySQL database. Some of these questions include:

1) ازدواج با اهل کتاب چه حکمی دارد؟
2) چرا هنگام مسواک زدن لثه ام خونی می‌شود؟
3) دختر از پدر چه مقدار ارث می‌برد؟
4) نماز خواندن با لباس خونی چه حکمی دارد؟
5) چگونه درد زایمان را تحمل کنیم؟

These queries were edited before being entered as queries and their stop words were removed. The output of questions 3 and 4 are shown in Tables 6 and 7.

Table 6: Results of the evaluations made on question 4

نماز خواندن با لباس خونی چه حکمی دارد؟					
حکمی	خونی	لباس	خواندن	نماز	
2031	128	768	807	4213	TF
1969	109	616	747	3009	DF
3,42461	6,31854	4,58664	4,39382	3,0005	IDF
3,53244	7,41993	5,71841	4,74674	4,20113	TF-IDF

Table 7: Results of the evaluations made on question 3

دختر از پدر چه مقدار ارث می برد؟				
می برد	ارث	پدر	دختر	
295	515	1368	1459	TF
279	438	1073	1017	DF
5,37867	4,92767	4,03167	4,08527	IDF
5,68713	5,79395	5,1401	5,86078	TF-IDF

The evaluation of the two mentioned questions had acceptable results, according to the predefined metrics.

In brief, in the proposed corpus, questions that are entered directly from the questions in the system as input by the user interface are retrieved with 100% accuracy. Questions that use a part of the vocabulary contained in the system questions lead to 100% accurate retrieval. Questions that are randomly generated by a human agent have a wide range of retrieval accuracy from bad to very good, depending on the type of vocabulary used and the quantity of the corresponding 4 parameters. More precisely, if the user input keywords are available in the database, the results will be very good, otherwise, the system output may not be good.

6- Experimental Results

As described in the previous section, after eliminating the stop words from 83,364 questions, 56,925 unique words remained. The first step of evaluating the corpus is designing a robust model, which is useful for the whole corpus rather than part of it. Therefore, the stored words in the MySQL database were analyzed with a focus on Tf-Idf and Df parameters. Table 8 displays the range of changes in these parameters.

Table 8: Range of changes in Tf-Idf and Df

Item	*Minimum value*	*Maximum value*
Tf-Idf	2.9	66.1
Df	1	3522

The scattered values of these parameters led to the design of two evaluation models that will be discussed in the following.

6-1- Model No.1

Designing, implementing, and evaluating this model have been done within 10 steps:

1. Dividing unique words into five clusters according to their Tf-Idf.
2. Calculating the average Tf-Idf for each cluster.
3. Selecting the sample word from the database randomly based on the average Tf-Idf in each cluster.
4. Creating an arbitrary question by a human agent using the selected word.
5. The created question is run as a query in the database by means of the designed UI
6. The results are fetched based on the model relevancy prediction.
7. Duplicate results are removed, and the first 5 results are retained. (k=5)
8. Based on a ground-truth annotation each result is assigned a score between 1 (least relevant) and 5 (most relevant).
9. The metrics are calculated for each query.
10. The quality of evaluation is determined by examining the calculated metrics.

Table 9 shows the unique words division in 5 clusters and their sample words.

In Table 10, cells with green color have a grade upper than 2 and are considered as a True result. Besides, cells with red color have a grade lower than 3 and are considered as a False result. Then, in Table 11, the standard metrics for Model No.1 are calculated.

6-2- Model No.2

This model is like Model No.1, just it uses df parameters instead of Tf-Idf.

1. Dividing unique words into five clusters according to their df values.
2. Calculating the average df for each cluster.
3. Selecting the sample word from the database randomly based on the average df in each cluster.
4. Creating an arbitrary question by a human agent using the selected word.
5. The created question is run as a query in the database by means of the designed UI.
6. The results are fetched based on the model relevancy prediction.
7. Duplicate results are removed, and the first 5 results are retained (k=5).
8. Based on a ground-truth annotation each result is assigned a score between 1 and 5 (most relevance).
9. The metrics are calculated for each query.
10. The quality of evaluation is determined by examining the calculated metrics.

Table 12 shows the unique words division in 10 clusters and their sample words. In Table 13, cells in green have a grade upper than 2 and are considered as a True result. Besides, cells in red have a grade lower than 3 and are considered a False result. Then, in Table 13, the standard metrics for Model No.1 are calculated.

Table 9: Dividing the unique words into 5 clusters and choosing sample words

cluster	*Tf-Idf range*	*Tf-Idf average*	*Selected Word*	*Pronunciation*	*Meaning*	*Tf-Idf*
1	2.9 – 8.3	7.239	یقین	/yagheen/	Certainty	6.1
2	8.4 – 13.7	10.53	ازدواج	/ezdevaaj/	Marriage	12.3
3	13.8 – 19.2	15.524	طلسم	/telesm/	talisman	18.8
4	19.3 – 24.9	21.81	تبعید	/tab'eed/	Exile	21.9
5	25 – 66.1	35.98	روح	/ruh/	ghost	25.9

Table 10: Ground-Truth Annotation given grade to the results

Selected Word	*Arbitrary Question*	*Grades 1 (least relevant) to 5 (most relevant)*				
		Result1	Result2	Result3	Result4	Result5
یقین	چگونه به یقیین برسیم؟	5	2	1	1	1
ازدواج	ازدواج با اهل کتاب چه حکمی دارد؟	5	4	5	3	4
طلسم	چگونه می توان طلسم را باطل کرد؟	5	3	2	4	5
تبعید	در چه صورت فرد تبعید می شود؟	4	4	4	5	5
روح	مشکلات روحی چگونه درمان می شود؟	4	1	5	2	4

Table 11: Calculated Metrics for Model No.1

Selected Word	*Reciprocal Rank*	*Average Precision*	*CG@5*	*DCG@5*	*NDCG@5*
یقین	1	1	10	7.581	1
ازدواج	1	1	21	12.867	0.987
طلسم	1	0.888	19	11.555	0.945
تبعید	1	1	22	12.617	0.940
روح	1	0.756	16	9.543	0.886

Table 12: Dividing the unique words into 10 clusters and choosing sample words

Cluster	*Df range*	*Df average*	*Selected Word*	*Pronunciation*	*English Equivalent*
1	11 - 20	15	خودسازی	/khodsaazi/	Self-construction
2	21 - 30	25	اهانت	/ehaanat/	contempt
3	31 - 40	35	مسئولیت	/mas’uliat/	responsibility
4	41 - 50	45	جوراب	/juraab/	socks
5	51 - 60	55	خرما	/khorma/	Date palm
6	61 - 70	65	پوسیدگی	/puseedegi/	decay
7	71 – 80	76	لاغر	/laaghar/	thin
8	81 – 90	86	امانت	/amaanat/	trusteeship
9	91 – 100	95	صدقه	/sadagheh/	alms
10	100 - 3522	332	زکات	/zakat/	zakat

Table 13: Ground-Truth Annotation given grade to the results

Selected Word	*Arbitrary Question*	*Grades 1(least relevant) to 5(most relevant)*				
		Result1	Result2	Result3	Result4	Result5
خودسازی	خودسازی چگونه انجام می شود؟	3	5	5	4	5
اهانت	اهانت به اهل سنت چه حکمی دارد؟	1	5	5	2	1
مسئولیت	مسئولیت والدین درباره فرزندان چیست؟	5	4	5	5	1
جوراب	نظر اسلام درباره جوراب پوشیدن خانم ها چیست؟	5	2	5	5	5
خرما	خرما خوردن چه فایده ای دارد؟	5	5	3	4	4
پوسیدگی	نشانه پوسیدگی دندان چیست؟	5	5	4	5	5
لاغر	چطور لاغر شوم؟	1	5	2	5	4
امانت	احکام مربوط به خیانت در امانت چیست؟	5	4	5	5	5
صدقه	چه کسی مستحق دریافت صدقه است؟	5	5	3	2	2
زکات	به چه چیزهایی زکات تعلق می گیرد؟	3	5	5	5	5

Table 14: Calculated Metrics for Model No.2

Selected Word	*Reciprocal Rank*	*Average Precision*	*CG@5*	*DCG@5*	*NDCG@5*
خودسازی	1	1	22	12.317	0.910
اهانت	0.5	0.583	14	7.904	0.793
مسئولیت	1	1	20	12.567	0.984
جوراب	1	0.804	22	12.855	0.946
خرما	1	1	21	12.929	0.991
پوسیدگی	1	1	24	14.248	0.992
لاغر	0.5	0.533	17	8.860	0.777
امانت	1	1	24	14.117	0.983
صدقه	1	1	17	11.292	1.000
زکات	1	1	23	12.748	0.912

In both models, while calculating the Reciprocal Rank (RR) and Average Precision (AP), results with grades lower than 3, are considered as a False result and True otherwise. Table 15 represents the comparison of the evaluation results of the two models.

Table 15: Comparing evaluation results of two models

Model	***MRR***	***MAP***	***NDCG@5***
No.1	1	0.929	0.951
No.2	0.9	0.892	0.929

Comparison of the evaluation results of the 2 models is as expected, as the parameter used in Model No.1 is more effective than the parameter used in Model No.2. Figs. 7 and 8 reveal more details from clusters in Model No.1.

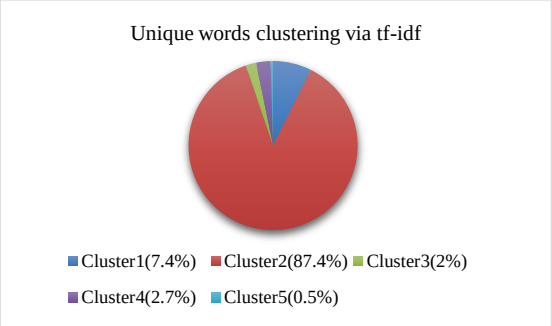

Fig. 7 Unique words clustering via Tf-Idf

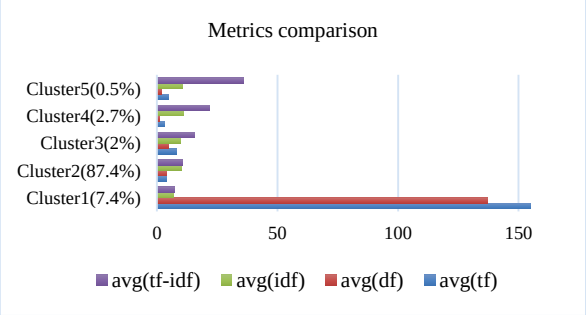

Fig. 8 Metrics comparison between five clusters of Model No.1

7- User Interface

Now, this corpus with 53,844 Persian questions and answers and the name "Popfa", has been provided to the interested parties and researchers. A part of the User Interface (UI) designed for it is shown in Fig. 9. In the user interface, questions that are entered as queries are returned based on two types of searches:

1) (Search by questions) جستجو بر اساس پرسش‌ها
2) (Search by answers) جستجو بر اساس پاسخ‌ها

The output of the UI is shown in Fig. 9, also this interface is available to the public through this link. For each question, the output is presented to the user as two separate lists based on the similarity to the questions and answers available in the created corpus.

Fig. 9 The designed UI

8- Discussion

Considering the number of questions in the corpus and the various classifications of it, it has a significant advantage over the few existing systems in the Persian language, both in terms of quantity and quality. It has 2 general classifications, Religious and Medical. Each class is divided into other sub-classes. Table 16 shows some of the thematic classifications of the questions and answers of the corpus. 20 different classifications in the designed system are presented in Table 16.

Table 16: Different classifications in the corpus

cat_id	***title***	***cat_id***	***title***
1	قرآن و تفسیر	11	اندیشه اسلامی
2	عقائد	12	حدیث شناسی
3	احکام	13	منطق
4	مهدویت و انتظار	14	فلسفه
5	تاریخ	15	کلام
6	فرق، ادیان و مذاهب	16	عرفان و تصوف
7	تربیت و مشاوره	17	حقوق
8	دین پژوهی	18	مسائل زنان
9	اخلاق	19	پزشکی
10	سیاست	20	علمی

The importance of addressing different simple and challenging question types and scenarios in the field of question answering systems is undeniable. It should be noted that Popfa corpus contains various kinds of questions including descriptive, factoid, confirmation, comparative, relationship-based, and list questions. However, the generated corpus does not include

hypothetical and complex questions. Hypothetical questions ask for information associated with any hypothetical event and no specific answers to these questions are necessarily available. For example:
What will happen if a big earthquake occurs in Tehran?
Complex questions are more challenging to answer, and their answers generally consist of a list of different kinds of answers. For instance:
What are the reasons for heavy traffic in megacities?
These questions can have various answers according to the idea of each person.
Here, to ensure ease of access for readers, we have made the corpus available on GitHub, where it can be downloaded and utilized for research and development purposes. The link to access the Popfa Persian Question Answering corpus can be found in this link.

9- Conclusion and Future Works

In this paper, some weaknesses and challenges of the Persian language question answering systems have been highlighted. Then, Popfa question answering corpus has been generated due to a standard procedure to fulfill the mentioned need for Persian NLP tasks. In this regard, different methods for generating a question answering corpus are investigated. Eventually, the innovative method of designing a semi-intelligent crawler in this paper has led to the production of a corpus containing 53,844 questions in Persian. Furthermore, by performing the corpus standardization processes and designing a user interface for better communication between the audience and the system, we achieved a set of standard and reliable questions and answers. In the last stage, we compared the Popfa corpus with the corpora used in the related question answering systems in the Persian language, in which many questions of the Popfa corpus and its sub-thematic diversity were evaluated as a competitive advantage.
Undoubtedly, the development and improvement of Persian question answering systems to handle all ranges of questions would require further research and implementation efforts. As the Popfa corpus contains various kinds of questions, future research can work on more complicated types of questions, such as hypothetical and complex questions. In this regard, exploring and explaining the potential approaches, techniques, and models that could be employed to tackle more complicated questions would indeed be a valuable area for future research. Additionally, the domain of questions can be diversified and expanded. Moreover, the user interface designed for Popfa can be developed into an intelligent robot. In the future, this user interface can be turned into a powerful text assistant using various machine learning methods. On top of that, the link to the created corpus is provided, so researchers can apply it in Persian question answering systems in the future.

Acknowledgments

The authors acknowledge that this study is edited by Dr. Belmond Yoberd, Divisional Director, Mott MacDonald Limited, United Kingdom.

References

[1] R. French, "The Turing Test: The first 50 years," Trends in Cognitive Sciences, vol. 4, no. 3, pp. 115-122, 2000.
[2] Z. Khalifeh Zadeh, and M. A. Zare Chahooki, "An Effective Method of Feature Selection in Persian Text for Improving the Accuracy of Detecting Request in Persian Messages on Telegram," Journal of Information Systems and Telecommunication (JIST), vol. 8, no. 32, pp. 249-262, 2021.
[3] N. Tohidi, and S. M. H. Hasheminejad, "A Practice of Human-Machine Collaboration for Persian Text Summarization", in The 27th International Computer Conference, Tehran, 2022.
[4] A. Hoseinmardy, and S. Momtazi, "Recognizing Transliterated English Words in Persian Texts", Journal of Information Systems and Telecommunication (JIST), vol. 8, no. 30, pp. 84-92, 2020.
[5] N. Tohidi, C. Dadkhah, and R. B. Rustamov, "Optimizing Persian multi-objective question answering system," International Journal on Technical and Physical Problems of Engineering (IJTPE), vol. 13, no. 46, 2021.
[6] M. Breja, "A Customized Web Spider for Why-QA Pairs Corpus Preparation," Journal of Information Systems and Telecommunication (JIST), vol. 11, no. 41, pp. 41-47, 2023.
[7] Tohidi, Nasim., Dadkhah; Chitra. Rustamov, and Rustam B., "Optimizing the Performance of Persian Multi-objective question answering system", in The 16th International Conference on Technical and Physical Problems of Engineering, Istanbul, Turkey, 2020.
[8] C. P. Masica, The Indo-Aryan Languages, New York, Cambridge University Press, 1993.
[9] D. Khashabi, A. Cohan, S. Shakeri, P. Hosseini, P. Pezeshkpour, M. Alikhani, M. Aminnaseri, M. Bitaab, F. Brahman,S. Ghazarian, M. Gheini, A. Kabiri, R. Karimi Mahabagdi, O. Memarrast, et al., "ParsiNLU: A Suite of Language Understanding Challenges for Persian," Transactions of the Association for Computational Linguistics, vol. 9, p. 1147–1162, 2021.
[10] E. M. Voorhees, "The TREC-8 Question Answering Track Report (1999)," in In Proceedings of TREC-8, 1999.
[11] N. Tohidi, and S. M. H. Hasheminejad, "MOQAS: Multi-objective question answering system", Journal of Intelligent & Fuzzy Systems, vol. 36, no. 4, pp. 3495-3512, 2019.
[12] I. Khodadi, and M. Saniee Abadeh, "Genetic programming-based feature learning for question answering," Elsevier, Information Processing and Management, vol. 40, 2015.
[13] M. Joshi, E. Choi, D. Weld, L. Zettlemoyer, "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, 2017.

[14] A. Muttaleb Hasan, and L. Q. Zakaria, "Question classification using support vector machine and pattern matching," Journal of Theoretical and Applied Information Technology , vol. 87, no. 2, pp. 259-265, 2005.
[15] H. Veisi, and H. Fakour Shandi, "A Persian Medical Question Answering System," International Journal on Artificial Intelligence Tools, vol. 29, no. 6, 2020.
[16] A. Aleahmad, H. Amiri, E. Darrudi, and F. Oroumchian, "Hamshahri: A standard Persian text collection", Knowledge-Based Systems, vol. 22, no. 5, pp. 382-387, 2009.
[17] A. Mollaei, S. Rahati Quchani, and A. Estaji, "Question classification in Persian language based on conditional random fields", in 2nd International eConference on Computer and Knowledge Engineering (ICCKE), 2012.
[18] E. Sherkat, and M. Farhoodi, "A Hybrid Approach for Question Classification in Persian Automatic Question Answering Systems", in 4th International eConference on Computer and Knowledge Engineering (ICCKE), Mashahd, Iran, 2014.
[19] A. P. Ben Veyseh, "Cross-Lingual Question Answering Using Common Semantic Space", in Proceedings of the 2016 Workshop on Graph-based Methods for Natural Language Processing, San Diego, California, 2016.
[20] Y. Boreshban, H. Yousefinasa, and S.A. Mirroshandel, "Providing a Religious Corpus of Question Answering System in Persian", Signal and Data Processing, vol. 15, no. 1, pp. 87-102, 2018.
[21] R. Etezadi, and M. Shamsfard, "PeCoQ: A Dataset for Persian Complex Question Answering over Knowledge Graph", in 11th International Conference on Information and Knowledge Technology (IKT), Tehran, Iran, 2020.
[22] N. Abadani, J. Mozafari, A. Fatemi, M. A. Nematbakhsh, and A. Kazemi, "ParSQuAD: Persian Question Answering Dataset based on Machine Translation of SQuAD 2.0", International Journal of Web Research, vol. 4, no. 1, pp. 34-46, 2021.
[23] A. Kazemi, J. Mozafari, and M. A. Nematbakhsh, "PersianQuAD: The Native Question Answering Dataset for the Persian Language", IEEE Access, vol. 10, pp. 26045-26057, 2022.
[24] K. Darvishi, N. Shahbodagh, Z. Abbasiantaeb, and S. Momtazi, "PQuAD: A Persian Question Answering Dataset", arXiv:2202.06219, 2022.
[25] D. Jurafsky, and H. Martin James, Speech and Language Processing, Upper Saddle River, NJUnited States: Prentice Hall, 2019.
[26] R. Dragomir; H. Qi, H. Wu, and W. Fan, "Evaluating Web-based Question Answering Systems", in The Third International Conference on Language Resources and Evaluation (LREC'02), Las Palmas, Canary Islands - Spain, 2002.
[27] K. Järvelin, and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," ACM Transactions on Information Systems, vol. 20, no. 4, pp. 422-446, 2002.