Proceedings of the First workshop on Language Technologies for Government and Public Admnistration (LT4Gov), pages 7–14
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

FRAQUE: a FRAme-based QUEstion-answering system for the Public Administration domain

Martina Miliani[*,†], **Lucia C. Passaro**[†], **Alessandro Lenci**[†]
[*]Università per Stranieri di Siena, [†]CoLing Lab (Dipartimento di Filologia, Letteratura e Linguistica, Università di Pisa)
m.miliani@unistrasi.studenti.it, lucia.passaro@fileli.unipi.it, alessandro.lenci@unipi.it

Abstract

In this paper, we propose FRAQUE, a question answering system for factoid questions in the Public Administration domain. The system is based on semantic frames, here intended as collections of slots typed with their possible values. FRAQUE is a pattern-base system that queries unstructured data, such as documents, web pages, and social media posts. Our system can exploit the potential of different approaches: it extracts pattern elements from texts which are linguistically analysed by means of statistical methods. FRAQUE allows Italian users to query vast document repositories related to the domain of Public Administration. Given the statistical nature of most of its components such as word embeddings, the system allows for a flexible domain and language adaptation process. FRAQUE's goal is to associate questions with frames stored into a Knowledge Graph along with relevant document passages, which are returned as the answer. In order to guarantee the system usability, the implementation of FRAQUE is based on a user-centered design process, which allowed us to monitor the linguistic structures employed by users, as well as to find which terms were the most common in users' questions.

Keywords: Question Answering, Semantic Frames, Knowledge Graph

1. Introduction

Although late, Italy is slowly advancing in the digitization process of Public Administration data and services (Carloni, 2019). Now, more and more institutions in Italy manage data and delivery services on the web. Several municipalities started to adopt Question Answering Systems (QASs), chatbots, and digital assistants to ease citizens' access to public data. A wide range of citizens can use these systems since they permit to query vast repositories in natural language (Hovy et al., 2000; Ojokoh, 2018).

In this paper, we propose FRAQUE (FRAme-based QUEstion-answering), a domain-specific question answering system for factoid questions. Our system exploits semantic frames, here intended as templates consisting of a set of slots typed with their possible values (Minsky, 1974; Jurafsky and Martin, 2019). Thanks to frames, our QAS can query unstructured data, such as documents, web pages, and social media posts. We applied FRAQUE to the administrative domain in the Italian language. Nonetheless, the system is potentially adaptable to different domains and different languages. It relies on the statistical components of CoreNLP-it (Bondielli et al., 2018) for morphosyntactic analysis, which exploits the Universal Dependencies (UD) annotation scheme (Nivre, 2015). Statistical components are also employed for the semantic analysis of questions for Named Entity Recognition (NER) and term extraction. Finally, our system performs query expansion following an unsupervised approach based on word embeddings (Mikolov et al., 2013).

A first implementation of FRAQUE has been developed on the administrative domain. Our target users are municipality officers and common citizens who need to access the rich amount of information hidden in public documents. In particular, we decided to focus on citizens, who are supposed to use a QAS to get notice about municipality regulations and to receive other kind of information related to a certain administrative area. In order to guarantee the effectiveness and the usability of FRAQUE, we followed user-center design principles introduced by Gould and Lewis (1985).

We collected questions written by Italian native speakers to assess FRAQUE's outcomes. We tested FRAQUE on the administrative domain by employing the information extracted from a set of Italian documents including administrative acts, social media posts, and official municipality web pages. In particular, FRAQUE has been embedded into a dialogue management system and has been tested as a module of a larger project involving several instruments developed for the Public Administration (PA) domain.

The paper is structured as follows: An overview on QASs is given in Section 2., the definition of FRAQUE methodology is outlined in Section 3. The evaluation of the system in a real-case scenario is described in Section 4.

2. Related Work

Existing QASs have been categorized in different ways, e.g. depending on the addressed question type (e.g., confirmation questions, factoid questions, list questions), on the features of consulted data bases (e.g., full relational databases, RDF databases), on the adopted approaches and techniques (Ojokoh, 2018).

According to Dwivedi and Singh (2013) and Pundge et al. (2016) QASs can be distinguished into three different categories on the basis of the adopted approach: *linguistic approach* (Green et al., 1961; Clark et al., 1999; Fader and Etzioni, 2013; Berant et al., 2013), *statistical approach* (Moschitti, 2003; Ferrucci, 2010; Chen et al., 2017; Devlin et al., 2019) and *pattern matching approach* (Ravichandran and Hovy, 2002; Paşca, 2003).

QASs based on a linguistic approach exploit Natural Language Processing (NLP) and language resources such as knowledge-based or corpora. The knowledge architecture of these systems relies on production rules, logic, frames, templates, ontologies, and semantic networks (Dwivedi and Singh, 2013). On the one hand, the linguistic approach is very effective in specific domains. On the other hand, it shows limitations in portability through different domains, since building an appropriate knowledge base has usually heavy time costs. On the contrary, statistical approaches are easily adapted to various domains since they are independent of any language form. This kind of QASs are often based on Support Vector Machine (SVM) classifiers, Bayesian classifiers, Maximum Entropy models and Neural Networks (NN). Such question classifiers analyze the user's question to make predictions about the expected answer type, thanks to statistical measures. Statistical QASs require an adequate amount of data to train the models, therefore in this case the development cost moves from the manual production of linguistic rules to the preparation of annotated resources to feed the classifiers. Pattern matching approaches exploit text patterns to analyze the question to select and return the right answer. For example, the question "Where was Cricket World Cup 2012 held?" corresponds to the pattern "Where was `<Event Name>` held?" and is associated with the answer pattern "`<Event Name>` was held at `<Location>`" (Dwivedi and Singh, 2013). These systems are less complex than those exploiting linguistic features, which require time and specific human skills, and most of them automatically learn patterns from texts (Dwivedi and Singh, 2013; Hovy et al., 2000).

Furthermore, as reported by Jurafsky and Martin (2019), there are two different major paradigms of QASs: *information-retrieval based* and *knowledge-based*. In the former case, systems leverage on a vast quantity of textual information, which is retrieved and returned thanks to text analysis methods (Brill et al., 2002; Paşca, 2003; Lin, 2007; Fader and Etzioni, 2013; Chen et al., 2017; Devlin et al., 2019). In the latter case, semantic data are already structured into knowledge bases (Green et al., 1961; Clark et al., 1999; Ravichandran and Hovy, 2002; Fader and Etzioni, 2013; Berant et al., 2013). Finally, *hybrid systems*, like IBM Watson DeepQA (Ferrucci, 2010), rely both on text datasets and structured knowledge bases to answer questions.

Following such a classification, FRAQUE can be seen as an hybrid approach system. Firstly, it is based on linguistic analysis through statistical methods, which serves as prerequisite to maximize the performance of pattern matching techniques application. Secondly, it draws its data from a thesaurus and a Knowledge Graph (KG) both structured into semantic frames. In the thesaurus, simple terms, complex terms, and named entities related to the same frame are clustered and arranged into patterns exploited for the question analysis. In the KG, each slot frame contains a text passage (i.e., a single sentence *snippet*), selected through a ranking process measuring its relevance for that frame slot. Differently from relational databases, a pre-defined set of relations is not required by a KG, so that a more flexible object-oriented data storage is guaranteed (Miliani et al., 2019). Moreover, FRAQUE applies statistical techniques to identify and cluster data, such as word embeddings and classifiers.

3. The FRAQUE Methodology

In this section we present an overview of the user-centered design process employed to create FRAQUE. Moreover, we report on its components through the three main stages described in Dwivedi and Singh (2013), namely *document analysis*, *question analysis* and *answer analysis*.

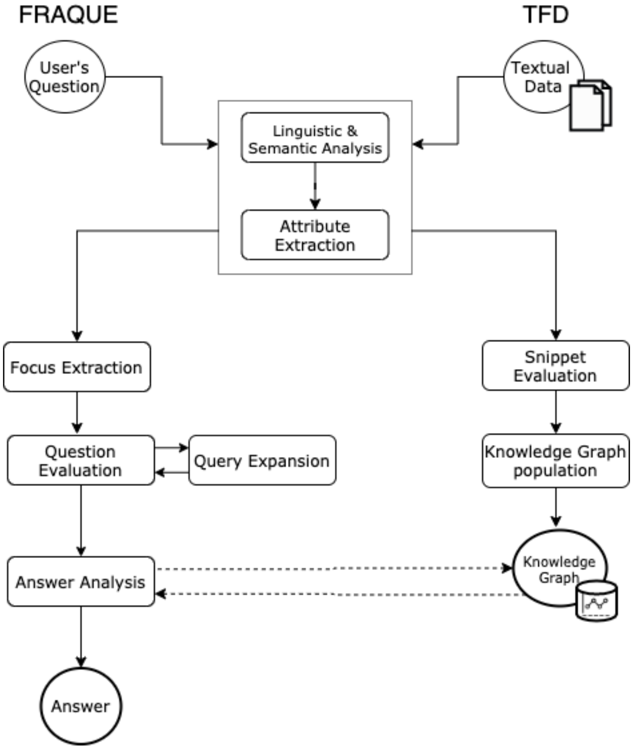

Figure 1: The diagram shows the FRAQUE analysis pipeline, which shares some modules with the Text Frame Detector (TFD) system (Miliani et al., 2019). Components in the central box belong to both FRAQUE and TFD systems. Except for the *answer analysis* component, all the other FRAQUE modules are employed in the *question analysis* described in Section 3.3.

3.1. User-Centered Design Process

We decided to adopt a user-centered design process (Gould and Lewis, 1985) to consider users' needs as a fundamental requirement for FRAQUE implementation. We distributed a questionnaire to 30 users divided into four age groups: $18 - 25$ (15%); $26 - 35$ (33.3%); $36 - 50$ (20%); $51 - 65$ (30%). We asked the users to write a small number of questions, pretending to interact with a QAS. The questionnaire allowed us to monitor the linguistic structures employed by users, as well as to find which terms were the most common in users' questions so that it was easier to identify frame triggers and attribute triggers. (see Section 3.2.). Further linguistic features detected by analyzing users' questions were: (i) lack of punctuation; (ii) variable length of questions: from 1 to 15 tokens (the shorter ones contained only keywords, as if the users were querying a search engine); (iii) typos. Considering (i) and (ii), we opted for avoiding fixed pattern for question analysis: we decided to look for patterns of unordered elements on the question text, without sticking to fixed term sequences.

3.2. Document Analysis

Document analysis consists of identifying candidate documents and detecting possible answers among document snippets (Dwivedi and Singh, 2013). The knowledge base employed by our system is a KG populated by the *Text Frame Detector* (TFD), an Information Extraction (IE) system described by Miliani et al. (2019) (see Figure 1), containing semantic frames selected through the design process described in Section 3.1. (see Figure 2).

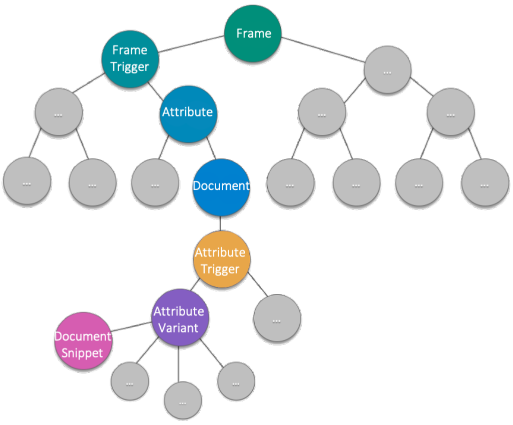

Figure 2: The Knowledge Graph structure employed by the TFD (Miliani et al., 2019).

3.2.1. Linguistic Analysis and preparatory IE process

As anticipated, FRAQUE and TFD have been embedded into a dialogue management system as the QAS of a chatbot. The systems are part of a bigger project that involves several instruments aimed at analyzing and indexing documents belonging to the PA domain. In particular, FRAQUE and TFD work downstream of a complex indexing process composed of both general purpose and domain specific components. First of all, TFD exploits two different linguistic pipelines: T2K^2 (Dell'Orletta et al., 2014) and CoreNLP-it (Bondielli et al., 2018). The former has been adapted for administrative acts analysis, the latter for the annotation of questions and texts like social media posts, since it includes statistical models for tokenization, sentence splitting, Part-of-Speech (PoS) tagging, and parsing. For event detection, our QAS exploits a model embedded in the broader system where it has been integrated. To extract NEs, the Stanford NER (Manning et al., 2014) is employed. In particular, it exploits the INFORMed PA (Passaro et al., 2017) model to extract entities related to the administrative domain. Furthermore, it employs EXTra (Passaro and Lenci, 2016) for in-domain complex terms extraction.

Table 1 shows the performances of the components used for the morphosyntactic and semantic analysis of texts. As anticipated, T2K^2 has been employed to analyze administrative acts, but to our knowledge its performances have not been assessed on the PA domain yet. We report an evaluation performed over general-purpose documents (Dell'Orletta, 2009).

Nevertheless, it is worth mentioning that morphosyntactic annotation underlying INFORMed PA, and EXTra was carried out with the adapted version of T2k^2 to the PA domain.

Component	PA	Measure	Score
T2K^2: PoS tagging	no	Accuracy	96.34%
CoreNLP-it: PoS tagging	no	F1	0.97
INFORMed PA	yes	F1$_{\text{MacroAVG}}$	0.77
EXTra	yes	Precision	93.50%

Table 1: Performances of each component exploited for the morphosyntactic and semantic analysis of texts in FRAQUE. The *PA* column indicates whether each module has been tested on the administrative domain.

3.2.2. Detecting Frames

In FRAQUE, each frame F encodes semantic categories relevant for a specific domain, such as the Tax frame for the administrative domain. "Municipality Tax" or "Garbage Tax" are linguistic cues called *frame triggers* (F_t) and enable the detection of frame instances on texts. **Deadline** and **methods of payment** are considered *attributes* (A). Attributes encode the relevant features of the semantic category represented by each frame. *Attribute triggers* (T) ease the attribute extraction from texts. T and Ft are both expressed by simple and complex terms, Named Entities (NEs), and Temporal Expressions (TEs). For instance, the **deadline** attribute is detected by the triggers "disbursement", "installment", and usually by date (see Figure 3). For ease of reading, the examples provided along the paper have been translated in English.

> The [Municipality Tax]$_{tax}$ [disbursement]$_{payment}$ must be made through [wire transfer]$_{payment-form}$ or [postal order]$_{payment-form}$ in two [installments]$_{sum}$: [down payment]$_{sum}$ by [June 18$^{\text{th}}$]$_{date}$ and [balance]$_{sum}$ by [December 17$^{\text{th}}$]$_{date}$.

Figure 3: Example of a snippet expressing an instance of the Tax frame. It contains relevant information for both the **deadline** and the **methods of payment** attributes.

Triggers are stored in an thesaurus and linked to the related frames and attributes. They are registered with their standard form s and a small number of orthographic and morphosyntactic variants v selected by domain experts. Trigger variants are expanded with their semantic neighbors to improve frame and attribute recall. In Figure 3, the attribute triggers "wire transfer" and "postal order" are tagged with their standard form "payment-form".

After the linguistic analysis, we applied TFD to search frame and attribute triggers on the text, in the same or adjacent sentences. The snippet in Figure 3 shows the trigger for the Tax frame "Municipality Tax" along with several attribute triggers: simple terms, such as "disbursement" and "installment"; complex terms, like "wire transfer" and "postal order"; and TEs, i.e. "June 18th" and "December 17th". The extracted sentences are ranked according to different scores, taking into account metrics like the number of retrieved triggers related to a given attribute, the average distance (in tokens) between the frame and the attribute triggers, the sentence length. Consider the snippet in Figure 3 concerning the attribute **methods of payment**: there are three retrieved triggers ("disbursement", "wire transfer" and "postal order"); the average token distance between the frame trigger "Municipality Tax" and these triggers is $(0 + 5 + 7)/3 = 4$ (e.g., "wire transfer" is five tokens distant from "Municipality Tax"); finally, the sentence length is 22 tokens.

The sentence with the highest rank is linked to the related attribute. More specifically, each candidate snippet receives a double score, a *Sentence Score* (SS), which ranks it within the set of snippets extracted from the same document, and a *Document Score* (DS) ranking it within the set of snippets extracted from the entire collection of documents (Miliani et al., 2019).

Frame instances are stored in a Neo4j[1] KG. As shown in Figure 2, each frame corresponds to a root node, which is represented by the TAX frame in the proposed example in Figure 4. Each *frame node* is connected with all the frame triggers found on the collection of documents. If we consider the snippet in Figure 3, the instance of the frame is given by the trigger "Municipality Tax", which labels the *frame trigger node* connected to "Tax" in Figure 4.

Frame trigger nodes are linked to attribute nodes. For instance, the snippet in Figure 3 contains information about the attribute **deadline**. This *attribute node* is connected to at least a *document node*, representing the document where the attribute has been extracted from: we took as example a "Rome Municipality Act". A snippet with the higher SS for the connected attribute is stored together with the document node. The snippet is also registered with its DS. One of the triggers extracted from the snippet in Figure 3 is "June 18th", which labels the *trigger variant node*: this node is connected on one side to a *trigger node* marked by its standard form, i.e. "date", and on the other side to the *snippet node* representing the snippet containing the trigger.

3.3. Question Analysis

Question analysis includes parsing, question classification, and query reformulation (Dwivedi and Singh, 2013). The main goal of the question analysis module is to find a match between a question and at least a frame attribute indexed into the KG. The analysis is carried out exploiting some components shared with the TFD for the linguistic annotation and the frame extraction (See Fig. 1), a *focus detection* (Cooper and Ruger, 2000) and a *question evaluation* process, aiming at associating each question to the right frame and attribute and formulate the query to the KB. With the same goal, a *query expansion* module exploits word embeddings to find triggers among the semantic neighbours of questions ngrams (see Figure 1).

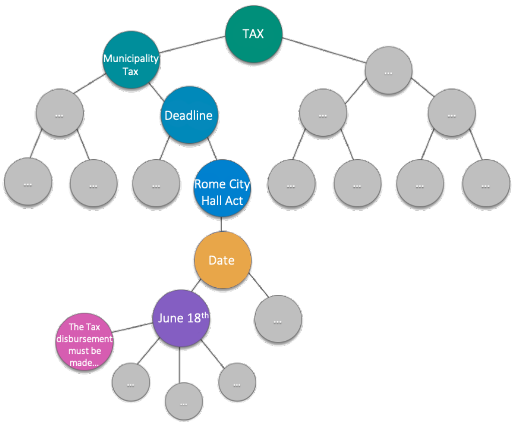

Figure 4: The Knowledge Graph populated by the TFD with an instance of the attribute **deadline**, belonging to the TAX frame.

The morphosyntactic analysis of questions is carried out by the CoreNLP-it pipeline, whereas rule-based components are exploited for NER. GATE[2] and the Stanford TokensRegex (Chang and Manning, 2014) are used to extract from questions the entities annotated with statistical components during the document analysis phase (See 3.2.1.).

Given a set of frame attributes A, an attribute $a \in A$ is identified in a question by the co-occurrence of a frame trigger F_t and a subset of the attribute triggers set T associated with it, such as $A = \{F_t, T\}$, where $T = \{t_1, ..., t_n\}$. Triggers are grouped by several standard forms $\{s_1, ..., s_n\}$, such as $S = \{s_1, ..., s_n\}$ (see Section 3.2.). Moreover, a subset Q of T is implicitly expressed on text by means of question foci. Thus, $Q \subset T$ and $S \subset T$.

The TFD module employed by FRAQUE for *attribute extraction* looks for a frame trigger F_t to possibly associate the question with a frame F. For instance, in this phase the frame trigger for the TAX frame "Municipality Tax" is extracted from the question in Figure 5. Then, the TFD searches for attribute triggers related to the TAX frame attributes. Different degrees of flexibility can be set for the attribute retrieving. A binary feature assigned to each trigger t_i suggests if the trigger is compulsory for associating an attribute with the examined question (Miliani et al., 2019). In the example in Figure 5, the *attribute extraction* module detects only the generic trigger "payment", which led to associate the question with both the attributes **deadline** and **methods of payment**.

When the [payment]$_{payment}$ of the [Municipality Tax]$_{tax}$ is due?

Figure 5: Example of a user question containing the question focus ("when"). The tagged tokens are attribute triggers and tags correspond to their standard forms.

If no attribute is activated, a *query expansion* module checks if simple and complex terms extracted from questions are at least semantic neighbors of the triggers con-

[1] http://neo4j.com/

[2] https://gate.ac.uk/

tained in the thesaurus. Semantic neighbors are computed within a distributional space trained with word2vec (Mikolov et al., 2013) on *La Repubblica* corpus (Baroni and Mazzoleni, 2004) and PaWaC (Passaro and Lenci, 2017) for administrative domain-specific knowledge. FRAQUE searches for the terms extracted from the question among the distributional space targets. Target words are lemmatized and combined for complex terms. Following the compositional property of word embeddings, each complex term vector consists of an element-wise sum of its word embedding elements (Hazem and Daille, 2018). Semantic neighbors are then detected among the terms with the highest cosine similarity measure. Among these neighbors, FRAQUE searches for triggers.

To solve the potential ambiguity resulting from the *attribute extraction* process and to facilitate a connection between questions and attributes, we implemented a *focus detection* module. The question focus is expressed by interrogative adverbs, like "how", and by equivalent linguistic expressions composed by more than one token, such as "in which way".

Each focus is associated with an attribute trigger. For instance "how" is linked to the trigger "methods", whereas the focus "where" is related to a trigger represented by a location named entity.

The extracted focus is then involved in the *question evaluation* process. In Figure 5, the question focus is "when", which is associated with TEs. Thus, the snippet containing the answer of the cited question must include a TE. The attribute including a date among its trigger is the attribute **deadline**, which is therefore associated with the question.

Can I [pay]$_{payment}$ the [Municipality Tax]$_{tax}$ with [postal order]$_{payment-form}$?

Figure 6: Example of a user question. The tagged tokens are attribute triggers and tags correspond to their standard forms.

If the focus extracted from the question is not connected to any frame attribute, or if no focus has been extracted from the question (as showed in Figure 6), a different procedure is followed. In this case, the attribute selected is the one with the highest *Attribute Score* (AS). The AS is computed for each candidate attribute selected by the *attribute extraction* module, and it is defined as:

$$AS = \frac{|S_Q|}{|S_T|} \times \frac{\sum_{i=1}^{n} cos}{|T_Q|} \qquad (1)$$

where S_Q is the set of the standard forms of all the triggers T_Q extracted from the question and related to a certain attribute, such as $S_Q \subset S$ and $T_Q \subset T$. AS is directly proportional to the average of the cosine similarity between the triggers in T_Q and the triggers stored in the thesaurus. In this way, AS favors terms semantically closer to the triggers contained in the thesaurus, so that the noise resulting from query expansion process is reduced. Furthermore, AS does not consider only T_Q, the set of all triggers found on the question. AS takes into consideration the ratio between trigger standard forms in S_Q and T_Q, because it better expresses the variety of triggers by which an attribute is described on the text.

3.4. Answer Analysis

Finally, the extraction and ranking of candidate answers are carried out in the *answer analysis* (Dwivedi and Singh, 2013) (see Figure 1). The answer returned by FRAQUE is a snippet that is detected walking through the KG nodes, following a path indicated by the information extracted from the question during the *question analysis* phase. Once the question is analysed we identify three different scenarios:

- The *attribute* scenario: the question is associated with an attribute;
- The *frame* scenario: the question is linked to a frame, can be specified;
- The *residual* scenario: the question cannot be related to any attribute or frame.

In the first scenario, FRAQUE uses the question analysis results to query the KG and retrieve a snippet. Consider the question in Figure 5, which is related to the attribute **deadline** of the TAX frame, and which contains the frame trigger "Municipality Tax". FRAQUE looks at the root nodes inside the graph and selects the one labelled by "Tax". Then, it looks for "Municipality Tax" among the frame instances and checks for the presence of an *attribute node* tagged with "deadline" afterwards. At this point, if the requested information has to be extracted from the whole corpus, FRAQUE considers the snippets stored with each *document node* and returns the one with the highest DS. Otherwise, if the information has to be searched in a specific document (e.g., "Rome Municipality Act"), FRAQUE searches that document among those connected to the considered attribute, and returns the snippet associated with it.

In the *frame* scenario, only a frame trigger has been extracted from the question, but no focus or attribute trigger can disambiguate the user's information request. In this case, FRAQUE returns the document or the set of documents connected to the highest number of attribute nodes for the detected frame. Such documents are in fact supposed to contain a more complete knowledge about the frame itself.

In the *residual* scenario, triggers can not be detected neither among the question terms, nor among their semantic neighbours. In that case, FRAQUE extracts all metadata from the question, such as complex terms and entities, and uses them to query a document base indexed on Lucene[3]. In this database, the documents are indexed with terms, entities and topics related to the administrative domain. Terms and topics are structured in an ontology built by domain experts and employed for the platform SemplicePA (Miliani et al., 2017). FRAQUE returns those documents where the extracted terms and entities co-occur, by exploiting AND queries based on a list of pre-defined groups of metadata organized by type (i.e., terms, entities, and topic).

[3] https://lucene.apache.org/

4. Evaluation And Results

We evaluated FRAQUE on the administrative domain. In particular, we detected two frames: (i) the domain-specific TAX frame, and (ii) the EVENT frame, concerning the events taking place in a given city area, which we considered as a more general purpose frame (see Table 2). FRAQUE's outcomes are assessed on

To test our QAS, we selected 50 questions among those gathered through the questionnaire employed in the design process (see Section 3.1.), and among the FAQ reported on several Italian Municipality web sites. More precisely, we focused on a subset of questions referring to the target frames attribute (i.e., those asking information about events and taxes) and on another subset of questions not related to them. This way, we were able to evaluate the performances of the system for the three scenarios outlined in Section 3.4. Table 2 reports the frame attributes on which the performances of FRAQUE have been assessed.

FRAME	EVENT	TAX
ATTRIBUTES	Where When Cost	Deadline Methods of payment

Table 2: Attributes of the EVENT and TAX frames.

We evaluated FRAQUE on its ability to return (i) The right answer type; (ii) The right answer content. For what concern the first point, the goal is to assess whether the system is able to return the expected output type based on the scenarios described in Section 3.4. (i.e., *attribute*, *frame*, and *residual*). Traditional test accuracy metrics were employed, like F_1 *score*, which takes into consideration the overlap between the system outcomes and the correct answer type for each question.

	PRECISION	RECALL	F1-SCORE
macroAVG	0.69	0.57	0.61
microAVG	0.72	0.72	0.72

Table 3: Performances of FRAQUE for what concern the right answer type returned according to the detected scenario associated with the question.

Table 3 shows relatively low results for recall. Such a score is affected by the cases in which FRAQUE could not provide an answer to the question due to several reasons including (i) the absence of the information requested by the user and (ii) its ability to find the proper match within the question and the documents frames.

With regard to the second point (i.e., the answer content) a different evaluation was performed. A domain expert was asked to decide whether the returned snippets or documents (according to the detected scenario) contain the right answer to the questions. The metrics we used differ from one scenario to another (see Section 3.4.). Table 4 reports the FRAQUE's performances according to each scenario.

SCENARIO	MEASURE	PERFORMANCE
Attribute	MRR	0.58
Frame	MRR	0.75
Residual	Precision	0.59

Table 4: Evaluation of the content of the answers returned by FRAQUE according to the detected scenario. In the *frame* scenario, the system detected a frame, but no attribute related to it. In the *attribute* scenario, FRAQUE extracted at least a frame and an attribute from the text of the question. In the *residual* scenario, no frame could be extracted from the question text.

When the question can be associated with an attribute, as in the first scenario, we employed the *Mean Reciprocal Rank* (MRR). MRR is a metric introduced in the TREC Q/A track in 1999 for factoid question answering system evaluation (Jurafsky and Martin, 2019). For a set of questions N, it was computed on a short list of snippets containing possible answers, ranked by SS or DS (see Section 3.2.). Each question is then scored according to the reciprocal of the rank of the first correct answer. Given a set of questions Q:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{2}$$

where $rank_i$ refers to the rank position of the first relevant document for the i^{th} query. As for the *attribute scenario*, in table 5 we report a deeper evaluation over the various attributes.

FRAME	ATTRIBUTE	MRR
EVENT	Where	1
	When	0.75
	Cost	0
TAX	Deadline	0.60
	Methods of payment	0.50

Table 5: Evaluation of the answers to the questions related to EVENT and TAX frame attributes, according to the *attribute* scenario. The score is computed on the returned snippets.

It is important to notice that such results are highly affected by the **cost** attribute, for which the system was not able to find correct answers. Such errors are mainly due to a wrong indexed snippet for the corresponding attribute. Because of the high number of municipality acts stored in our database, most of the events have been extracted from this kind of documents. In most cases, these acts report how much the municipality spent to fund the events, instead of the ticket cost of the event. It is clear that we expect completely different results by evaluating the system on a knowledge base where information related to events is mainly extracted from social media posts, where the price of the ticket to participate in a certain event is usually specified.

In the *frame* scenario, the given question could not be associated with any attribute, so the documents containing relevant snippets for the detected frame are returned. Here, the MRR is calculated on the list of documents ranked by the number of the relevant snippets extracted from them and associated with the frame attributes. Table 6 shows the results for each frame concerning this scenario.

FRAME	MRR
EVENT	0.50
TAX	1
macroAVG	0.75

Table 6: Evaluation of the answers to questions related to EVENT and TAX frames, according to the *frame* scenario. The score is computed on the returned documents.

The low performance of the system in retrieving the information related to the EVENT frame is mainly caused by some features of the indexing process. TFD indexes a document only if it contains information relevant for at least one attribute. For this reason, even though the TFD stored an event in the graph, no document may be associated with it and thus returned.

In the *residual* scenario, no frame is associated to the question and the system queries a Lucene database with in-domain terms, entities, and topics extracted from the question text. In this case, FRAQUE returns up to 5 documents. Since the results are not ranked, the system performance was evaluated considering if at least one of the returned documents was actually relevant for the question. The employed evaluation metric is a variant of the precision: we considered as *true positive* only those cases where FRAQUE returned at least a relevant document for each query (seeTable 4). We decided to consider this metric also taking into consideration the QAS usage context, where the real goal is to guarantee that the information the user needs is among the returned documents.

The results showed that, in some cases, the queries returned no answers. On the one hand, this happens because we decided to maximize the quality of the returned results by employing AND queries in querying the Lucene database. Specifically, output documents were required to contain all (or pre-defined groups) of the relevant metadata identified in the text of the question. However, this way, the system never retrieves documents containing different combinations of terms, entities or topics extracted from the question. On the other hand, the errors are caused by the absence of documents related to the question topic. By evaluating FRAQUE without considering questions for which the Lucene database does not contain the needed information, the precision increases by 29%, reaching overall a performance of $0,76$.

5. Conclusions

In this paper we introduced FRAQUE, a question answering system based on semantic frames. FRAQUE structures textual data into frames so that they can be queried by means of natural language. This solution is based on an IE module for *document analysis*, namely the TFD (Miliani et al., 2019), allowing for the indexing of documents by text frames. Given this kind of metadata, FRAQUE is able to detect correct answers contained into document snippets and to associate them to frame attributes stored in a KG. FRAQUE has been integrated into a Dialogue Management System (DMS) as the question answering component of a chatbot, designed to give information about Italian Public Administrations.

However, in-domain linguistic analysis and resources in FRAQUE are easily portable to other domains, thanks to its statistical components, such as word embeddings, adopted in the query expansion module.

We evaluated FRAQUE in several real case scenarios obtaining encouraging results. The results calculated over the frames annotated with the TFD module reach an average MRR of $0,667$, whereas FRAQUE reaches a $0,59$ precision score in those questions not answered exploiting frames. Of course, there is still room for improvement, but if we consider only the cases where TFD performs well, FRAQUE reaches even higher results. By looking at these outcomes, we are led to believe that improving the TFD performances, the FRAQUE's ones can be drastically improved as well.

In the near future we plan to compare the obtained results with those of available related systems, at least on the first of the scenarios detected, where document snippets are returned as answer. Moreover, further development of our work will focus on the conversion of FRAQUE thesaurus to open standards, such as the *Resource Description Framework* (RDF), with the consequent adaptation of FRAQUE modules to this data model. This could ease the application of FRAQUE on existing resources, as well as facilitate other frameworks to exploit FRAQUE in-domain thesaurus.

6. Acknowledgements

This research has been supported by the Project "SEM il Chattadino" (SEM), funded by Regione Toscana (POR CreO Fesr 2014-2020). Besides the CoLing Lab, project partners include the companies ETI3 s.r.l. (coordinator), BNova s.r.l. and Rigel Engineering s.r.l.

7. Bibliographical References

Berant, J., Chou, Andrew Frostig, R., and Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical methods in natural language processing (EMLNP)*.

Bondielli, Passaro, and Lenci. (2018). CoreNLP-it: A UD pipeline for Italian based on Stanford CoreNLP. In *CliC-it 2018*.

Brill, E., Dumais, S., and Banko, M. (2002). An analysis of the askmsr question-answering system. In *Proceedings of conference on Empirical methods in natural language processing (EMLNP)*. Association for Computational Linguistics.

Carloni, E. (2019). Algoritmi su carta. politiche di digitalizzazione e trasformazione digitale delle amministrazioni. *Diritto pubblico*, 25(2):363–392.

Chang, A. X. and Manning, C. D. (2014). Tokensregex: Defining cascaded regular expressions over tokens. *Stanford University Computer Science Technical Reports. CSTR*, 2:2014.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open-domain questions. In *ACL*.

Clark, P., Thompson, J., and Porter, B. (1999). A knowledge-based approach to question-answering. In *Proceedings of AAAI*, pages 43–51.

Cooper, R. J. and Ruger, S. M. (2000). A simple question answering system. In *Text REtrieval Conference (TREC)*.

Dell'Orletta, F., Venturi, G., Cimino, A., and Montemagni, S. (2014). T2kˆ 2: a system for automatically extracting and organizing knowledge from texts. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 2062–2070.

Dell'Orletta, F. (2009). Ensemble system for part-of-speech tagging. *Proceedings of EVALITA*, 9:1–8.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT*, pages 4171–4186.

Dwivedi, S. K. and Singh, V. (2013). Research and reviews in question answering system. *Procedia Technology*, 1(10):417–424.

Fader, Anthony, Z. L. and Etzioni, O. (2013). Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Ferrucci, D. (2010). Build watson: an overview of deepqa for the jeopardy! challenge. In *Proceedings of 19th International Conference on Parallel Architectures and Compilation Techniques (PACT)*. IEEE.

Gould, J. D. and Lewis, C. (1985). Designing for usability: key principles and what designers think. *Communications of the ACM*, 25(3):300–311.

Green, B. F. J., Wolf, A. K., Chomsky, C., and Laughery, K. (1961). Baseball: an automatic question-answerer. In *Awestern joint IRE-AIEE-ACM computer conference*. ACM.

Hazem, A. and Daille, B. (2018). Word embedding approach for synonym extraction of multi-word terms. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., , and Lin, C.-Y. (2000). Question answering in webclopedia. In *Text REtrieval Conference (TREC)*.

Jurafsky, D. and Martin, J. H. (2019). Speech and language processing. Third edition draft on webpage: `https://web.stanford.edu/~jurafsky/slp3/`. Accessed: 3 July 2019.

Lin, J. (2007). An exploration of the principles underlying redundancy-based factoid question answering. *ACM Transactions on Information Systems (TOIS)*, 25(2):6.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS 2013, 26th Conference on Advances in Neural Information Processing Systems*, pages 171–178, Lake Tahoe, Nevada, USA.

Miliani, M., Passaro, L., Gabbolini, A., Passaro, L., Leci, A., and Battistelli, R. (2017). Semplicepa: Semantic instruments for public administrators and citizen. In *GARR*.

Miliani, M., Passaro, L. C., and Lenci, A. (2019). Text frame detector: Slot filling based on domain knowledge bases. In *Proceedings CliC-it 2019, 6th Italian Conference of Computational Linguistics*, Bari.

Minsky, M. (1974). *A framework for representing knowledge*. Massachusetts Institute of Technology, Cambridge, MA.

Moschitti, A. (2003). Answer filtering via text categorization in question answering systems. In *Proceedings of 15th IEEE International Conference on Tools with Artificial Intelligence*. IEEE.

Nivre, J. e. a. (2015). Universal dependencies 1.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Ojokoh, Bolanle ans Adebisi, E. (2018). A review of question answering systems. *Journal of Web Engineering*, 17(8):717–758.

Passaro, L. C. and Lenci, A. (2016). Extracting terms with Extra. *Computerised and Corpus-based Approaches to Phraseology: Monolingual and Multilingual Perspectives*, pages 188–196.

Passaro, L. C., Lenci, A., and Gabbolini, A. (2017). Informed pa: A ner for the italian public administration domain. In *Fourth Italian Conference on Computational Linguistics CLiC-it*, pages 246–251.

Paşca, M. (2003). *Open-Domain Question Answering from Large Text Collections*. CSLI.

Pundge, A. M., Khillare, S. A., and Namrata Mahender, C. (2016). Question answering system, approaches and techniques: A review. *International Journal of Computer Applications*, 141(3):0975–8887.

Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Association for Computational Linguistics conference (ACL)*. Association for Computational Linguistics.

8. Language Resource References

Baroni, M., Bernardini, S., Comastri, F., Piccioni, L., Volpi, A., Aston, G. and Mazzoleni, M. (2004). *La Repubblica*.

Passaro, Lucia C. and A. Lenci. (2017). *PaWaC - Public Administration Web As Corpus*.