A General Framework for Multiple Choice Question Answering based On Mutual Information and Reinforced Co-occurrence*

Jorge Martinez-Gil[1], Bernhard Freudenthaler[1], A Min Tjoa[1,2]

[1]Software Competence Center Hagenberg GmbH
Softwarepark 21, 4232 Hagenberg, Austria

[2]Vienna University of Technology
Favoritenstrasse 9-11/188, 1040 Vienna, Austria
e-mail: name.surname@scch.at

Abstract. As a result of the continuously growing volume of information available, browsing and querying of textual information in search of specific facts is currently a tedious task exacerbated by a reality where data presentation very often does not meet the needs of users. To satisfy these ever-increasing needs, we have designed an solution to provide an adaptive and intelligent solution for the automatic answer of multiple-choice questions based on the concept of mutual information. An empirical evaluation over a number of general-purpose benchmark datasets seems to indicate that this solution is promising.

Keywords: Expert Systems, Knowledge Engineering, Information retrieval, Question answering

1 Introduction

With the increasing amount of information that is available online, efficient and reliable computational techniques for accessing that information are needed. In fact, an ever-increasing amount of professionals from a wide range of disciplines agree that the information explosion that we are currently experiencing makes their work more tedious and even error-prone. The major reasons therefor are the fact that newly generated information is usually formatted in an unstructured way, and that the huge volume and speed at which that information is made available usually lead to information overload in their daily activities.

It is widely known that working with huge volumes of information has always been a major issue for computer scientists and practitioners in their efforts for applying for the latest advances in information technologies to offer solutions for

* This manuscript is an extended version of: J. Martinez-Gil, B. Freudenthaler, A Min Tjoa: Multiple Choice Question Answering in the Legal Domain Using Reinforced Co-occurrence. DEXA (1) 2019: 138-148.

the aforementioned problems. In fact, the latest advances in Big Data and Natural Language Processing have proven to be extremely useful for solving many problems that have traditionally affected the field of information processing. In practice, the daily operations of a wide range of professions require reading a large amount of textual material to identify the relevant documents and to locate the right piece of information needed. One step in the evolution towards the improvement of these processes emerges from Question Answering (QA) systems as a sub-field of Information Retrieval (IR). The design of QA systems is widely considered as an alternative to overcome traditional processes by providing accurate and understandable answers to specific questions, rather than presenting the user with a list of search-related documents [17].

The research community agrees that systems allowing generating automatic responses to textual questions could have a strong impact and practical implications in many diverse disciplines. In this way, efficient techniques for answering specific questions are in high demand and some systems implementing methods for answering questions have been designed to meet this need. However, QA technology faces some problems that prevent its progress. For example, typical approaches try to initially generate many possible responses for each question and then try to choose the right answer from all possible answers. However, the techniques for choosing the right answer need to be further improved. Moreover, the old assumption that answers to most of the questions are often explicitly stated somewhere, and the only remaining factor needed in addition is the access to a sufficiently large corpus have been proved to be inaccurate.

To effectively reason over knowledge derived from the text, QA systems must handle incomplete and potentially noisy knowledge. To tackle this problem, we have focused on computational techniques for mutual information exchange and reinforced co-occurrence analysis. Techniques of this kind have been widely used in various forms of research on content analysis, text mining, thesauri building, and ontology learning. Since the problem to be faced is too huge, our focus in this paper is laid in a particular sub-problem: multiple choice QA, i.e. answering questions in a scenario where the possible answers are already given beforehand [1]. This problem is very common in practice, as many people know how to determine the number of potential answers beforehand, and the fact that some potential clues are already given can also significantly help to reduce the workload. This is mainly because a QA system would be able to automatically process a huge amount of textual resources to find the answer that best matches a question, and that means that QA systems could save resources in the form of effort, costs and time in many fields where the explosion of information is causing problems.

Therefore, we propose here a novel framework intended to operate over huge text corpora to discover latent textual structures of existing textual representations that allow to automatically answer multiple-choice questions on any subject. Unlike our previous work [23], we envision our solution as a general-purpose framework so that conclusions being drawn apply to a wide range of specific do-

mains. Therefore, with this idea in mind, we present here our research from which the following contributions can be highlighted:

- **Contribution 1:** We propose a new method for the automatic answer of multiple-choice questions based on the notion of mutual information exchange and reinforced co-occurrence. The advantages of this proposal in comparison with the existing ones are:
  (1) **Advantage 1**: Our approach can generate a ranking of answers without the need of a training phase over the data
  (2) **Advantage 2**: Unlike most existing systems, our approach does not need to consume textual corpora whereby the correct answer to the question is explicitly stated
  (3) **Advantage 3**: Our approach can explain the results so that a human operator can understand the ranking of the generated answers
- **Contribution 2:** We have empirically evaluated our approach using some of the most common benchmark datasets for the automatic answering of questions in the legal, geographical, and historical field. And we have verified that the results are in line with those of the state-of-the-art despite having the present solution presents the aforementioned advantages over the most advanced techniques.

The remainder of this work is organized as follows: Section 2 reports the state-of-the-art on question answering methods and tools that have proven to be successful in the past. Section 3 presents the fundamentals of our contribution concerning the computation of the reinforced co-occurrence over huge corpora of text. Section 4 reports the empirical evaluation of our novel approach over some benchmark datasets and the analysis of the results that we have achieved from that evaluation. Finally, we outline the conclusions and future lines of research.

2 State-of-the-art

QA systems are traditionally considered as groups of interacting software components intended to automatically reply questions by analyzing different sources of either structured or unstructured information. In practice, these sources are usually called Knowledge Bases (KBs) and can lead to two different approaches to address the problem depending on the nature of the information to be exploited: structured or unstructured solutions. Each of them has different advantages and disadvantages. For example, working with structured KBs allows exploiting the knowledge represented by using the so-called inference engines, to infer new knowledge and to answer questions [38]. However, at present, there is not an automatic way to introduce a new entity into the KB nor to determine with which existing entities should be related and how [25]. Therefore, finding practical solutions is considered as an important research challenge and it currently matters of intense research [14].

The fact is that, in practice, it is not easy to implement these systems, so they have been progressively replaced by another type of more efficient systems based

on lighter knowledge models such as knowledge graphs [9] and other enhanced lexical semantic models [37], but in general, it is widely assumed that building a fully structured KB is expensive in terms of resource consumption, it is subject to many errors, it is usually difficult and expensive to maintain, and last but not least, it is usually hardly reusable in other contexts.

In contrast, systems exploiting unstructured KBs have more practical benefits as most of them have been specifically designed to efficiently process huge amounts of textual data (usually represented in natural language). These huge amounts of data come from existing documents, databases, websites, and so on. For this reason, the most frequent type of QA system that is mentioned in the literature is the one that uses different collections of unstructured natural language KBs. The current generation of QA systems has evolved to extract answers from a wide range of different plain machine-readable resources. These QA systems exploit the massive set of unstructured information available on some data sources to retrieve information about any particular question. It is important to note that these QA systems are only possible mainly due to recent advances in big data [13] and natural language technologies [19]. Moreover, since these novel QA systems are capable of processing questions about different domains and topics, they are now used in a wide range of different scenarios [24].

IR-based solutions represent words in the form of discrete and atomic units. For example, given the fact that todays web search engines can successfully retrieve simple answers to many queries expressed by a human operator just by searching the Web. Therefore, the first approach could be to query the number of Google results for a specific question and a given answer together. However, this solution has brought several problems like the lack of context (not to mention very serious problems related to denial of service). Li et al. proposed the exploitation of structured lexical databases and corpus statistics [22]. However, the method is not optimized for dealing with QA scenarios. To overcome these problems, word processing models such as LSA [8] and term frequency-inverse document frequency (TF-IDF) partially solve these ambiguities by using terms that appear in a similar context based on their vector representation, and then they group the semantic space into the same semantic cluster. In this context, one of the best-known QA systems is IBM Watson [11], that it is very popular for its victory in the televised show *Jeopardy* [12]. Although in recent times, IBM Watson has become a generic umbrella that includes other business analytics capabilities.

There is a second possible classification that distinguishes QA systems between closed-domain and open-domain. If we focus strictly on QA in closed-domain, we find that this technology has been used in real information systems, and especially in knowledge management systems [2]. The logic behind these systems is that given an issue, the extraction of relevant resources and the decision whether or not to use that content to answer the question are two key steps in building a system. In recent times, this approach has delivered many successful applications, e.g. in the legal area. In the literature we can distinguish between two major approaches: a) with structured KB. For example, Lame et al. [20] and

Fawei et al. [10] using ontologies, or Xu et al. [36] by exploiting other KBs such as Freebase. And b) exploiting unstructured KBs. For example, Brueninghaus and Ashley with a classical IR approach [4], Bennet et al. with strong focus on scalability [2], Maxwell and Schafer paying attention to context [27], Mimouni et al. with the possibility to make use of complex queries [28], or most modern deep learning techniques from Marimoto et al. [29] and Nicula et al. [30], the latter with good results, although with issues concerning the interpretability of the results.

Concerning open-domain, several systems capable to operate in general-purpose scenarios have been proposed. For example, the open-source system Calcipher [34], or the more advanced IR solver which uses the Waterloo corpus from Clark et al.[6]. The IR solver tries to determine if the question along with an answer option is explicitly stated in the textual corpus, and returns the confidence that such a statement was found. Another outstanding system is the DrQA system[1] that is available under an open-source license. This system addresses the challenge of open domain question answering using Wikipedia as unstructured KB. This means that the system has to combine the challenges of finding the relevant Wikipedia pages with that of identifying the answers from those pages. What we present here is an open-domain system that uses unstructured KBs to face multiple-choice questionnaires about any subject. Our system benefits from features such as no need for training (typical of systems that use machine learning), no need to find explicit answers in the textual corpus which it is used as background (typical of early QA systems), and the ability to provide answers with a high degree of interpretability (as opposed to proposals based on neural models).

3 General Framework for Multiple Choice Question Answering based On Mutual Information

Our approach is intended to automatically process massive amounts of textual information to look for evidence allowing to infer the most promising answers with regards to the huge range of questions that people can make. In this way, our contribution is a novel framework for automatically answering multiple-choice questions concerning a wide range of topics. This approach needs to fulfill two stages: first, we need to calculate alignment matrices between the question and the possible choices using textual corpora, and in the next stage, we need to normalize the results to produce a final result and associated ranking of possible answers. Next subsections introduce the technical preliminaries, the notion of reinforced co-occurrence, the normalization process, the implementation of our approach, and several running examples that show how this approach works in practice.

[1] https://github.com/facebookresearch/DrQA

3.1 Technical preliminaries

To overcome the current limitations of exiting QA approaches, we propose to automatically analyze the mutual information exchange[7] between a pre-processed version of the question and each of the possible choices in the context of different corpora of unstructured text. In this context, the mutual information exchange of two random variables is a measure of the mutual dependence between the two variables, i.e. the mutual information I(Q; C) between two random variables Q and C is the amount of information that the choice C gives about the question Q. This can be formally defined as:

Let (q, c_n) be a question and a possible answer with values over the space $\mathcal{Q} \times \mathcal{C}$. If their joint distribution is $P_{(Q,C)}$ and the marginal distributions are P_C, the mutual information between them could be:

$$I(Q;C) = D_{\mathrm{KL}}(P_{(Q,C)} \| P_Q \otimes P_C) \tag{1}$$

Our framework considers a pair of entities q and c_n that belong to two discrete random variables Q and C quantifies the probability of their co-occurrence given their joint distribution and their specific distributions. It can be mathematically expressed such as:

$$\mathrm{P}(q, c_n) \equiv \log \frac{p(q, c_n)}{p(q)p(c_n)} = \log \frac{p(q|c_n)}{p(q)}. \tag{2}$$

Our approach minimizes when the information overlap between the question and the potential choice is 0 (i.e. $p(q|c) = 0$ or $p(c|q) = 0$), which means that the two variables considered are independent. On the other hand, our approach maximizes in the (rare case of) the question and the potential choice might be perfectly associated (i.e. $p(q|c) = 1$ or $p(c|q) = 1$), yielding the following bounds:

$$-\infty \leq \mathrm{P}(q, c_n) \leq \min\left[-\log p(q), -\log p(c_n)\right]. \tag{3}$$

Therefore, we treat the QA problem of ranking the choice set such that the correct hypothesis is the one associated with a higher score and therefore, it is placed on the top of the ranking. We learn a scoring function $S(H, z)$ with a normalization parameter z such that the score of the correct choice (i.e. its corresponding co-occurrence probability) is higher than the score of the other hypotheses and their corresponding co-occurrence probabilities. Some interesting properties are:

(1) If $Q \perp C, I(Q;C) = 0$ because $H(Q) = H(Q|C)$
(2) If $Q = C, I(Q;C) = H(Q)$
(3) If $Q = f(C), I(Q;C) = H(Q)$ where f is deterministic
(4) If $C = g(Q), I(Q;C) = H(C)$

As a final note, it is necessary to remark that mutual information is symmetric, i.e. $I(Q;C) = I(C;Q)$, this means that in our application we do not need to worry about one direction than the other since though mathematically they are the same. The symmetry can be proven such as:

$$H(Q) + H(C|Q) = H(Q,C) = H(C,Q) = H(C) + H(Q|C)$$
$$H(Q) - H(Q|C) = H(C) - H(C|Q)$$
$$I(Q;C) = I(C;Q)$$

3.2 Reinforced Co-occurrence

Our text mining approach works under the distributional assumption [21]. This assumption has proven to perform well for several problems in the past. We hypothesize that the most important words of the question and the answers will co-occur in a small fraction of the given textual corpora. Our goal is to identify and analyze this co-occurrence, to present to the user our suggestions based on the automatic interpretation and normalization of that co-occurrence.

As the mutual information method can measure how much the actual probability of a particular co-occurrence of the question and the possible choice differs from what we would expect it to be based on the probabilities of the individual events and the assumption of independence. The question arises when dealing with the concept of co-occurrence itself. Many authors use the same text sentence whereas others assume that a text frame of n-units should be considered. Many others applied the notion of the paragraph, and so on. Our proposal considers an intelligent aggregation of all of them. That is why we call it reinforced co-occurrence. Formally, reinforced co-occurrence takes input a set of numeric values from the different aspects to be analyzed and outputs an aggregated number that it is supposed to represent in a meaningful way some of the most important characteristics of the input set. And it can formally be expressed in the following way:

$$P_r = \prod_{i=0}^{i=n} P_i(q, c_n) \tag{4}$$

The key research question is how this aggregation should be performed to deliver the best possible results. To answer that question, we propose a software framework to experiment on how that aggregation could be carried out. At this point, it is important to remark that we handle the concept of trust in terms of physical proximity [26]. For example, if a given (pre-processed) question and potential answers appear in the same paragraph of a document, we will have, at least, low evidence of a relation between them. But if this pair seems to appear together frequently, in the same sentences, or pre-defined text frames, or even in the context of the same regular expressions from the textual corpora, then we could infer that we could have an answer for the given question. This is precisely what can be achieved through the reinforced co-occurrence. However, exact technical details have to be defined using proper fine-tuning.

Moreover, it is obvious to see that the design of such a framework in this context is far from being trivial. However, our experience in rapid prototyping and testing text mining pipelines has shown us that it is possible to reach a

reasonable level of success [24]. According to our experience, a solution that works very well is a method with four levels of co-occurrence depending on the context whereby the question and the choice being evaluated can be found together. For this reason, we propose to work with various levels of co-occurrence, ranging from quite low degrees of restriction to very high degrees: text frame, regular expression, sentence, and paragraph. This way of working means that very few co-occurrences can be found, but the key to all of this is that those co-occurrences found will be very precise. Therefore, the corpus of text to be used must be huge. Otherwise, it is quite probable that our technique will not be able to obtain the values (since the restrictions that we impose are very strict).

Finally, it is necessary to remark that the problem addressed here is based on short response models. The reason is that these models provide the potentially correct answer in the form of a number, a name, a date, or even a short phrase or text fragment. This makes the work of our text mining engine much easier. It is also important to note, that this assumes that there are different ways of asking questions, and most of them are characterized by the formulation of questions expressed by interrogative particles (i.e. what, who, why, when, where, where) or some kind of is-a or have-a association. At the same time, the aforementioned possible choices are expressed in natural language, and therefore, they need some pre-processing too.

3.3 Normalization

In cases where the decision is not clear, for example, several answers have all the cells in their associated column filled with values, we apply normalization. Normalization is the process of mitigating the impact of the outliers on the final decision. This is done using adjusting the values from different scales to a common scale. Some words are much more common than others. Therefore, their associated co-occurrence values will be always much higher than others. To mitigate this effect, we applied an exponential reduction of the values obtained, so the highest values are significantly reduced in comparison with the lower ones. More formally, $\alpha = 1 - e^{(-1/w)}$. With exponential normalization, the averaging window w includes the desired number of reinforced co-occurrence values, although the lowest values weight more.

3.4 Implementation

Although the concept seems to be not quite straight forward, there is a huge technical limitation for its development from a pure engineering perspective. This approach is limited by an important number of technical issues which should be overcome. These limitations, originally identified by [3], are inherent to the process of massive text mining and include: Limitations concerning the corpora size, variability inherent to the processing of natural language (verbal forms, plurals, etc.), issues concerning the different domain nomenclatures, degree of uncertainty on the accuracy of the contents, and language in which the information is represented.

Accordingly, IR systems are usually designed in the form of a pipeline, i.e. a workflow whereby the data is processed in a way that the output of one module is the input of the next one. Figure 1 shows us an overall view of our IR pipeline. These components are related to each other and process the textual information available on different levels until the QA process has been completed. The questions formulated which serve as input for our system are initially processed by the question analysis component. This process is very important to transfer just meaningful data into the mining phase, whereby the calculation of the reinforced co-occurrence will take place. Answer extraction [35] will assign the proper results to each of the possible choices. Finally, it is necessary to normalize the raw data and create the final ranking to be delivered. The main modules of our QA system could be summarized in the following steps:

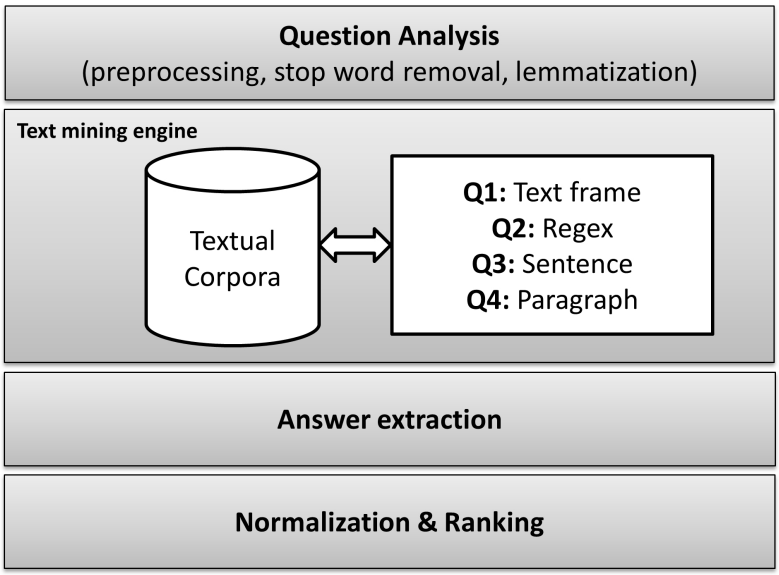

Fig. 1: Overall view of a pipeline designed to answer the multiple-choice tests. First of all, questions and answers need to be pre-processed. After this, a text mining engine is in charge of mining reinforced co-occurrence patterns. Then, these patterns are analyzed. Finally, the results are normalized and a ranking of potential choices is delivered

- *Question Analysis.* It is in charge of pre-processing both the question and the possible answers. To do that, it is necessary to remove the stop words and very common words (prepositions, adverbs, articles, etc.), to proceed with a lemmatization process, determining the root of the words to prevent irregular forms (i.e. plurals, third persons, etc.) to affect the co-occurrence.
- *Reinforced Co-Occurrence Calculation.* The logic behind this module consists of counting how many times the pre-processed question and the evaluated

answer co-occur together in the same text frame, in the same text expression, in the same sentence, and the same paragraph. Some parameters should be manually tuned.
- *Answer Extraction.* It consists of compiling the results and assign them to each of the possible choices. After this process, we have just raw values that need to be refined.
- *Answer Normalization and Ranking.* In this work, we usually work with exponential reductions, but other methods need to be considered in future work. The ranking consists of creating an ordered list of response according to the score obtained after normalization. Also, a heatmap is automatically created to deliver an explanation suitable for humans who need to understand how the ranking has been created.

3.5 Running Examples

To illustrate how our approach works, we have designed a running example to better understand how our pipeline processes the information. Let us see a couple of examples.

Running Example 1 Let us think in a question whereby we would like to know in which island is the volcano Etna situated. Let us think how the question could be, and how the different choices would look like.

```
On which island is Etna volcano located?
a) Sicily
b) Corsica
c) Rhodes
d) Sardinia
```

To do that, we can see in Figure 2 the graphical summary of how this process is performed: The question and the associated choices have to be preprocessed to remove non-relevant words, perform lemmatization, etc. Then, the system continues working by conveniently dividing the information into different parts which will be transferred to the following process which is a text mining engine that looks for the reinforced co-occurrence of the question and each possible answer. As a result, we get the reinforced co-occurrence values that have to be normalized so the outliers might not behave an extreme weight in the final value. As we see, the word island presents relatively high co-occurrence values, which makes sense since the island is a very common word. One would expect to find that word many times in the corpus, so in case that the answer is not clear, this effect will be mitigated with an exponential reduction of the highest values.

After repeating this process for each of the possible choices (Sicily, Corsica, Rhodes, or Sardinia), we have must discern whether it is the correct one. After performing the corresponding pre-processing, and reinforced co-occurrence calculation, we get the Table 1 the raw results. Since just one column has all its cells with values, these results are definitive, and they give a very clear clue that

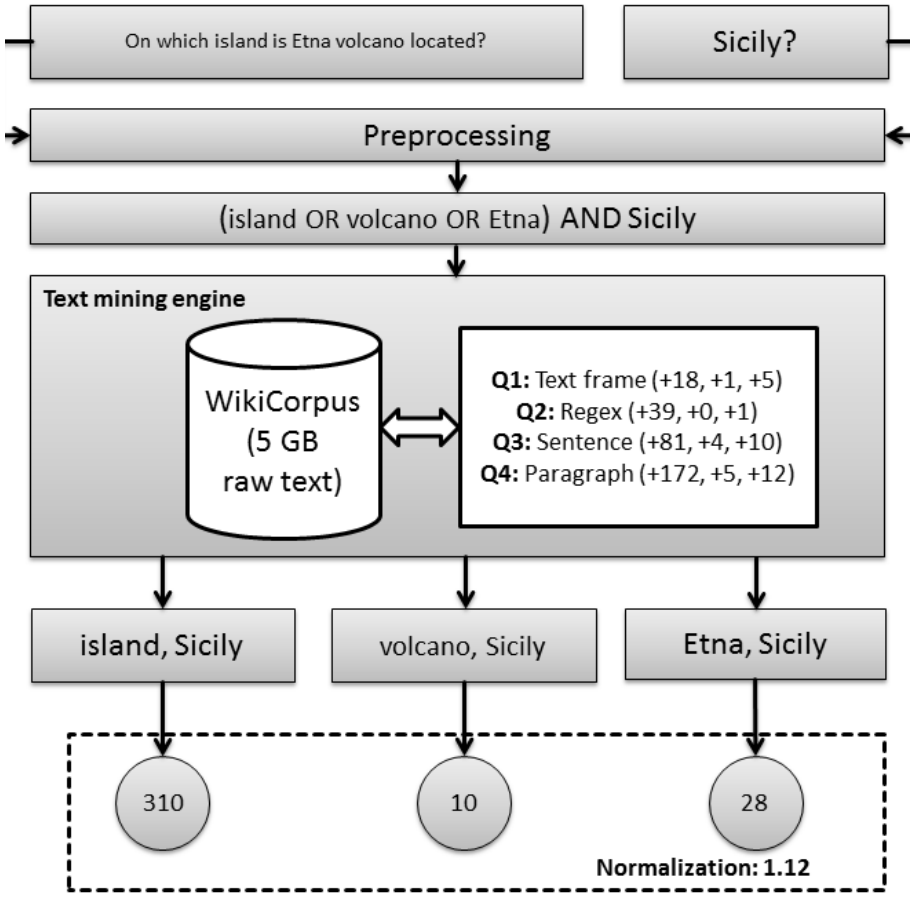

Fig. 2: Overall view of one iteration whereby a question (On which island is Etna volcano located?) and a potential answer (Sicily) is evaluated

Reinforced Co-occur.	Sicily	Corsica	Rhodes	Sardinia
island	310	161	262	142
volcano	10	0	0	0
Etna	28	0	0	0

Table 1: Raw results obtained for the reinforced co-occurrence of using WikiCorpus

the option chosen is going to be Sicily (which on the other hand is the correct answer).

Therefore, the choice that our system would select as the most promising one is a) Sicily, also the correct answer according the ground truth. The other three possible choices has a strong relation with the word island but not to volcano or Etna, so they would not even be considered as the final answer. In the next subsection, we also explain how a heatmap might allow to visually inspect the rationale behind the result for interpretability issues.

Running Example 2 Let us think in a question whereby we would like to know from which country did Papua New Guinea got its independence. Let us think how the question could be, and how the different choices would look like.

```
From which country did Papua New Guinea got its independence?
a) Mozambique
b) Australia
c) Indonesia
d) New Zealand
```

After the three first processing stages, i.e. Question Analysis, Reinforced Co-Occurrence Calculation, and Answer Extraction; we have been able to get the values that we see in Table 2.

Reinforced Co-occur.	Mozambique	Australia	Indonesia	New Zealand
country	256	6028	1033	2309
Papua New Guinea	5	411	137	144
independence	134	225	611	114

Table 2: Raw results obtained for the reinforced co-occurrence of using WikiCorpus

Everything seems to indicate that Australia will be the option finally chosen. Please note that Australia, Indonesia, and New Zealand as countries appear very frequently. However, in this case, normalization will reduce the impact of this fact on the final result. Therefore, the choice that our system would select as the correct one is b) Australia, what is also the correct one according to the ground truth. The second would be b) New Zealand. And the other two possible choices has lower values for reinforced co-occurrence, so they would not even be considered as plausible answers. As in the previous example, a heatmap allows to visually inspect the rationale behind the result for accountability and interpretability issues.

3.6 A brief note on the interpretability of our solution

The higher the interpretability of a solution, the easier it is for a human user to understand why the predictions have been made. A solution is assumed to be

more interpretable than another one if its decisions are easier for a human to understand than decisions from the other solution. For this reason, we envision the result of our process by not just choosing the most promising choice, but also we figure out how to represent the final answer in the form of a heatmap. The idea behind that it is offering a heatmap so that our solution might be more interpretable. This is mainly because, in some scenarios requiring accountability and/or interpretability, it is not just enough to provide the answer, but some reasons for that answer. By visually inspecting the heatmap, a human operator can understand how the decision has been made.

Figure 3 shows the heatmap corresponding to Running Example 1. The presence of values in each of the cells of the column Sicily indicates that the choice Sicily will be ranked first.

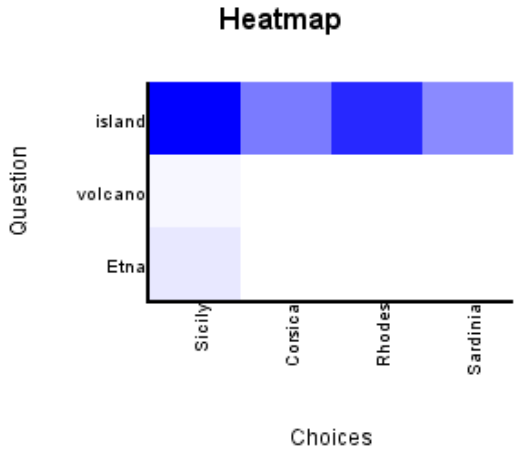

Fig. 3: Heatmap obtained for the scenario whereby an user wants to know in which island is the volcano Etna located. Higher values in the column Sicily correctly indicates that the desired answer is Sicily

Figure 4 shows the heatmap corresponding to Running Example 2. The highest values in the column Australia clearly indicates that this choice will be ranked first.

We want to emphasize that the heatmap can be generated in two different approaches: with raw values or with normalized values. It is up to the user to decide what it is better for a specific purpose.

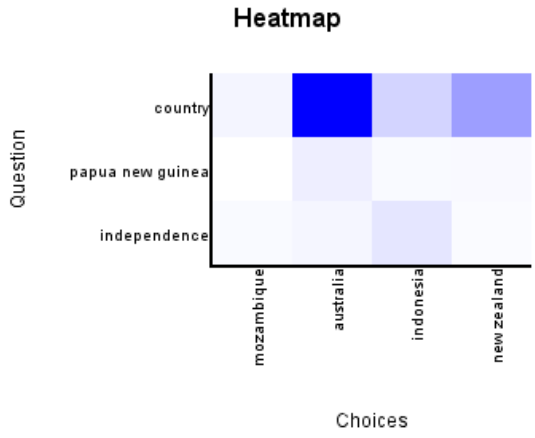

Fig. 4: Heatmap obtained for the scenario whereby an user wants to know from which country did Papua New Guinea got its independence. Higher values in the column Austria indicates that the desired answer is Australia

4 Results

We explain here the results. It is important to remark that results are highly dependent on the base textual corpus that will be processed. Choosing a relevant, specific base corpus to evaluate each of the possible choices is important in this situation. On the other hand, the task of evaluating the system is a vital stage, as it will assess the performance, as well as the accuracy of the techniques. In this work, we have chosen the strictest methodology to evaluate systems, which consists of a binary classification. The answer was right or wrong. However, there are many proposals in this sense, being some of the most popular those that grant a score according to the ranking that the evaluated system gives to the correct answer.

4.1 Setup configuration

We explain here the implementation decisions that we have taken to achieve a prototype for testing our hypothesis. The most important implementation details of our approach include:

– Limitations concerning the corpus size. With the emergence of new paradigms approaches for big data management, this kind of problems is losing importance. In this work, we have used WikiCorpus [32] which is a reduced version of Wikipedia. Wikicorpus is widely popular in the text mining community since it combines a great number of general-purpose articles represented in almost 5 GB of plain text.

– Variability is inherent to the processing of natural language. In this work, we have relied on the Krovetz solution [18] to proceed with the lemmatization. Besides, we have implemented some functionality to avoid processing verbs, common stop words, and nouns with a low meaning load.
– Issues concerning domain nomenclature. The problem for methods trying to exploit information extraction strategies is that they should be adapted to each different domain. It is widely assumed that meaning is usually represented by nouns (and noun phrases) so that it is common to built retrieval methods based on noun representations extracted. Since we are building a framework intended for general purpose, we have not taken design decision within this regard.
– Degree of uncertainty on the accuracy of the contents. In this work, we assume the fact that it is quite likely that the corpus to be analyzed might have some errors or inaccurate information. However, we foresee that the impact of these errors might be blurred by the overwhelming presence of correct information.
– Language in which the information is represented. To overcome this limitation, we have decided to use only English in this version of the work. WikiCorpus [32] is represented in English, so we have no problems with this regard.
– Other additional parameters: Other additional parameters are the size of the text frame (we have chosen a text frame of 5 units), and the kind of regular expressions to be considered that we have chosen are (is-a) and (have-a), the exploration of more sophisticated regular expressions is pending as future work.

4.2 Experiments

As a demonstration, we report here the results from three different datasets: legal, geographical, and historical. We have chosen 60 multiple-choice questions (20 for each dataset), and we have compared the results with those achieved by several publicly available QA systems. The legal questions have been retrieved from textbooks by the editorial Oxford University Press[2].

At the same time, the multiple-choice questionnaires on geography and history have been taken from the OpenTrivia approach [16]. But please note that since ours is a general-purpose framework, in principle, there would be no restriction to operate on other datasets. On the other hand, the proposals we are going to compare with are: the open-source system Calcipher [34], the once-outstanding solution Li et al. [22], and the classic but yet very powerful [8] using the classical configuration. Finally, since there is a 25% chance of making the right choices just by answering randomly, that result will be our baseline.

Legal dataset We have worked with a dataset on questions of legal nature. The reason is to check if our solution could help to alleviate the problem of

[2] http://www.oup.com

information overload in the legal area, which is currently one of the professional fields that needs it the most. An example of question is

```
A procedure of peaceful settlement of international dispute is a:
a) Conciliation (correct)
b) Cooperation procedure
c) Jurisdiction
d) Resolution
```

The summary of results that we have achieved are summarized in Table 3.

Approach	Correct Answers	Accuracy
Baseline	5/20	25%
Calcipher [34]	7/20	35%
Li et al. [22]	9/20	45%
LSA-Classic [8]	9/20	45%
Our Approach	13/20	65%

Table 3: Comparison with other approaches regarding the legal dataset

Our approach is able to beat the rest of solutions by correctly answering around one third of the 20 questions. Rest of QA systems are not able to properly answer even half of the questions, although they manage to beat the baseline.

Geographical dataset The second benchmark dataset is about questions of general geography. Sometimes it is very difficult to know a certain data about geography. We now want to see if our proposal could satisfactorily help a human operator. An example of multiple-choice question is

```
What is the deepest freshwater lake on Earth?
a) Onega
b) Ladoga
c) Huron
d) Baikal (correct)
```

The results achieved by all the approaches considered are summarized in Table 4.

Once again, our solution has managed to correctly answer more questions than the rest of the proposals, which this time also fail to reach half the desired answers.

Historical dataset The third dataset is about questions about the history of mankind, regardless of date or geographical region. It is very difficult for a human operator to store all this encyclopedic knowledge. For this reason, we want to know if our proposal could be useful in this sense. One example question is:

Approach	Correct Answers	Accuracy
Baseline	5/20	25%
Calcipher [34]	7/20	35%
Li et al. [22]	6/20	30%
LSA-Classic [8]	9/20	45%
Our Approach	12/20	60%

Table 4: Comparison with other approaches regarding the geographical dataset

```
Name the first great Greek tragic playwright who is now acknowledged
as the Father of Drama
a) Aeschylus (correct)
b) Aesop
c) Euripides
d) Sophocles
```

The results that the QA systems considered have achieved are summarized in Table 5.

Approach	Correct Answers	Accuracy
Baseline	5/20	25%
Calcipher [34]	5/20	25%
Li et al. [22]	8/20	40%
LSA-Classic [8]	8/20	40%
Our Approach	13/20	65%

Table 5: Comparison with other approaches regarding the historical dataset

Once again, our proposal is ranked first, just ahead of LSA and Li et al. which also fail to reach half the correct answers this time. Calcipher presents the worst performance.

4.3 Discussion

QA technology is becoming an important solution in many areas overloaded by the constant generation of large amounts of information in the form of free text. In this context, being able to automatically answering specific questions correctly can contribute to alleviating the problem of dealing with those huge amounts of unstructured text. This technology, however, faces some obstacles in its development. And it requires engineering work to properly tune some of the parameters associated with the processes that intervene in the pipeline.

The lessons learned from this work can be applied in more advanced situations where the possible choices are not present. At this point, we would need a

way to automatically generate possible choices, which will then be evaluated by our system. Moreover, it is important to remark that the choice of the different alternatives for answering the questions is a critical point. Therefore, it is necessary to evaluate the fairness of the choices to be evaluated. In the future, we want to use the knowledge base YAGO [15] for automatically generate candidate choices.

5 Conclusions and Future Work

Methods and techniques for automatically answering specific questions are in high demand, and as a result, many solutions for QA have been developed to respond to this need. The major reason for that is that the capability to automatically answer questions using computers could help alleviate a problem involving tedious tasks such as an extensive information search what is, in general, time-consuming. By automatically providing hints concerning a wide number of topics, lots of resources in the form of effort, costs and time can be preserved. In this work, we have presented our general framework for automatically addressing multiple-choice questions and the development of techniques for automatically finding the correct answer through mutual information and reinforced co-occurrence.

We have seen that although approaches based on structured KB often yield good results, it is difficult to use them in practice mainly due to the time and associated cost when building such structured KB (i.e. it is expensive in terms of effort, costs and time needed) and it is often very difficult to find experts for curating the KBs. In contrast, our approach is more suitable when selecting the actual right choice from a list of the possible answers due to the advances in big data processing and natural language technology. Although with some limitations, the experiments that we have performed over general-purpose datasets yields good results and seem to be promising. Moreover, in the present work, we have not yet fully explored the characteristics of many texts to utilize these features for building our QA system. For example, properties such as references between articles should be investigated more deeply as part of future work.

As additional future lines of research, we also need to work towards improving the technical limitations that we were not able to overcome within the frame of this work. This includes the capability to work with different multilingual textual corpora at the same time, the proper processing of verbs when formulating questions and preparing potential answers, the sentiment analysis of the questions and answers, and the proper aggregation of the different features through a training phase that can help to appropriately configure the complete pipeline. We think that by successfully addressing these challenges, it is possible to build solutions that can help to the many users to overcome one of the most serious problems that they have to face in their daily activities.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful suggestions to improve this work. This research has been supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

References

1. B. I. Aydin, Y. S. Yilmaz, Y. Li, Q. Li, J. Gao, M. Demirbas: Crowdsourcing for Multiple-Choice Question Answering. AAAI 2014: 2946-2953.
2. Z. Bennett, T. Russell-Rose, K. Farmer: A scalable approach to legal question answering. ICAIL 2017: 269-270.
3. S. Blohm, P. Cimiano: Using the Web to Reduce Data Sparseness in Pattern-Based Information Extraction. PKDD 2007: 18-29.
4. S. Brueninghaus, K. D. Ashley: Improving the representation of legal case texts with information extraction methods. ICAIL 2001: 42-51.
5. Akshay Chaturvedi, Onkar Arun Pandit, Utpal Garain: CNN for Text-Based Multiple Choice Question Answering. ACL (2) 2018: 272-277.
6. P. Clark, O. Etzioni, T. Khot, A. Sabharwal, O. Tafjord, P. D. Turney, D. Khashabi: Combining Retrieval, Statistics, and Inference to Answer Elementary Science Questions. AAAI 2016: 2580-2586.
7. K. W. Church and P. Hanks. Word association norms, mutual information and lexicography. In 27th ACL, pg. 7683, 1989.
8. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, R. A. Harshman: Indexing by Latent Semantic Analysis. JASIS 41(6): 391-407 (1990).
9. J. Ding, Y. Wang, W. Hu, L. Shi, Y. Qu: Answering Multiple-Choice Questions in Geographical Gaokao with a Concept Graph. ESWC 2018: 161-176.
10. B. Fawei, J. Z. Pan, M. J. Kollingbaum, A. Wyner: A Methodology for Criminal Law and Procedure Ontology for Legal Question Answering. JIST 2018: 198-214.
11. D. A. Ferrucci: Introduction to This is Watson. IBM Journal of Research and Development 56(3): 1 (2012).
12. D. A. Ferrucci, Anthony Levas, Sugato Bagchi, David Gondek, Erik T. Mueller: Watson: Beyond Jeopardy! Artif. Intell. 199-200: 93-105 (2013).
13. A. Hameurlain, F. Morvan: Big Data Management in the Cloud: Evolution or Crossroad? BDAS 2016: 23-38.
14. K. Hoeffner, S. Walter, E. Marx, R. Usbeck, J. Lehmann, A.-C. Ngonga Ngomo: Survey on challenges of Question Answering in the Semantic Web. Semantic Web 8(6): 895-920 (2017).
15. J. Hoffart, F. M. Suchanek, K. Berberich, G. Weikum: YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artif. Intell. 194: 28-61 (2013).
16. M. Joshi, E. Choi, D. S. Weld, L. Zettlemoyer: TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. ACL (1) 2017: 1601-1611.
17. O. Kolomiyets, M.-F. Moens: A survey on question answering technology from an information retrieval perspective. Inf. Sci. 181(24): 5412-5434 (2011).

18. R. Krovetz: Viewing morphology as an inference process. Artif. Intell. 118(1-2): 277-294 (2000).
19. S. Kumar Ray, S. Singh, B. P. Joshi: Exploring Multiple Ontologies and WordNet Framework to Expand Query for Question Answering System. IHCI 2009: 296-305.
20. G. Lame: Using NLP Techniques to Identify Legal Ontology Components: Concepts and Relations. Artif. Intell. Law 12(4): 379-396 (2004).
21. L. Lee: Measures of Distributional Similarity. ACL 1999.
22. Y. Li, D. McLean, Z. Bandar, J. O'Shea, K. A. Crockett: Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Trans. Knowl. Data Eng. 18(8): 1138-1150 (2006).
23. J. Martinez-Gil, B. Freudenthaler, A Min Tjoa: Multiple Choice Question Answering in the Legal Domain Using Reinforced Co-occurrence. DEXA (1) 2019: 138-148.
24. J. Martinez-Gil, B. Freudenthaler, T. Natschlaeger: Automatic recommendation of prognosis measures for mechanical components based on massive text mining. IJWIS 14(4): 480-494 (2018).
25. J. Martinez-Gil: Automated knowledge base management: A survey. Computer Science Review 18: 1-9 (2015).
26. J. Martinez-Gil: An overview of textual semantic similarity measures based on web intelligence. Artif. Intell. Rev. 42(4): 935-943 (2014).
27. K. T. Maxwell, B. Schafer: Concept and Context in Legal Information Retrieval. JURIX 2008: 63-72.
28. N. Mimouni, A. Nazarenko, S. Salotti: Answering Complex Queries on Legal Networks: A Direct and a Structured IR Approaches. AICOL 2017: 451-464.
29. A. Morimoto, D. Kubo, M. Sato, H. Shindo, Y. Matsumoto: Legal Question Answering System using Neural Attention. COLIEE@ICAIL 2017: 79-89.
30. B. Nicula, S. Ruseti, T. Rebedea: Improving Deep Learning for Multiple Choice Question Answering with Candidate Contexts. ECIR 2018: 678-683.
31. P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang: SQuAD: 100, 000+ Questions for Machine Comprehension of Text. EMNLP 2016: 2383-2392.
32. S. Reese, G. Boleda, M. Cuadros, L. Padr, G. Rigau: Wikicorpus: A Word-Sense Disambiguated Multilingual Wikipedia Corpus. LREC 2010.
33. A. Shtok, G. Dror, Y. Maarek, I. Szpektor: Learning from the past: answering new questions with past answers. WWW 2012: 759-768.
34. M. Stam: Calcipher System. Retrieved from https://github.com/matt-stam/calcipher, on 01-04-2019.
35. H. Sun, F. Wei, M. Zhou: Answer Extraction with Multiple Extraction Engines for Web-Based Question Answering. NLPCC 2014: 321-332.
36. K. Xu, S. Reddy, Y. Feng, S. Huang, D. Zhao: Question Answering on Freebase via Relation Extraction and Textual Evidence. ACL (1) 2016.
37. W.-T. Yih, M.-W. Chang, C. Meek, A. Pastusiak: Question Answering Using Enhanced Lexical Semantic Models. ACL (1) 2013: 1744-1753.
38. Y. Zhang, S. He, K. Liu, J. Zhao: A Joint Model for Question Answering over Multiple Knowledge Bases. AAAI 2016: 3094-3100