

# Using Graphs for Shallow Question Answering on Legal Documents\*

Alfredo Monroy<sup>1</sup>, Hiram Calvo<sup>1</sup>, and Alexander Gelbukh<sup>1,2</sup>

<sup>1</sup> Center for Computing Research, National Polytechnic Institute  
Mexico City, 07738, Mexico

<sup>2</sup> SoNet RC, University of Central Europe in Skalica, Slovakia  
hcalvo@cic.ipn.mx, alopezm301@ipn.mx, gelbukh@gelbukh.com

**Abstract.** This work describes a Shallow Question Answering System (QAS) restricted to legal documents. This system returns a set of relevant articles extracted from several regulation documents. The set of relevant articles allows inferring answers to questions posed in natural language. We take the approach of representing the set of all the articles as a graph; the question is split in two parts (called A and B), and each of them is added as part of the graph. Then several paths are constructed from part A of the question to part B, so that the shortest path contains the relevant articles to the question. We evaluate our method comparing the answers given by a traditional information retrieval system—vector space model adjusted for article retrieval, instead of document retrieval—and the answers to 21 questions given manually by the general lawyer of the National Polytechnic Institute, based on 26 different regulations (academy regulation, scholarships regulation, postgraduate studies regulation, etc.); with the answer of our system based on the same set of regulations. The results show that our system performs twice as better with regard to the traditional Information Retrieval model for Question Answering.

## 1 Introduction

Modern search engines allow retrieving a set of documents given a set of keywords. After this step, the user should make an additional effort to filter them and to find the specific information that he or she needs [1]. In recent years, there has been an extensive research of alternative methods which allow to access information in an easier way than the one provided by the current search engines. Part of this research has focused in Question Answering Systems (QAS). These systems are classified with regard to the collection of documents which they access. If the collection is about several subjects, the system is called an Open Domain QAS; whereas for a single subject, it is called Restricted Domain QAS. Restricted Domain QAS (RDQAS) have received more attention lately: Several RDQAS have been developed, including voice recognition [2], focused in medical areas [3,4,5], domestic areas [2], construction [6], and legal [7,8].

Particularly for the legal domain, the workshop *Question Answering for interrogating legal documents* took place in 2003, in the framework of the JURIX Forum (The

---

\* Work done under partial support of Mexican Government: SNI, SIP-IPN, COFAA-IPN, and PIFI-IPN.

foundation for Legal Knowledge Based Systems). Several works showed that a common problem is that traditional Information Retrieval Methods are not adequate to find the relevant fragments which answer legal questions because they do not consider the logical relationships between articles. In addition, many questions require an answer which cannot be found explicitly in a single article, or fragments of them, but intrinsically in the relationship between articles [9,10]. Some works use logic inference mechanisms such as COGEX System [14] and the system by Quaresma *et al.* [7]. However, these systems need expensive resources such as ontologies, axioms, and are language dependent. To avoid such requirements, we propose using a graph for capturing the relationships between articles in regulations answering with a set of articles related with the question and between them, hence the denomination Shallow QA—see Section 2. For details of the rest of the System, see Section 3.

Unfortunately, RDQAS do not have a standard evaluation system as the Open Domain QAS<sup>1</sup> have, mainly because of the heterogeneous characteristics of these systems. Still there is a long way for exploration in this subject. We test our system with regard to a traditional vector space model information retrieval system to answer questions particularly for the Spanish language given a set of 26 regulation documents from the National Polytechnic Institute. Details of our evaluation and experiments are given in Sections 4 and 5.

## 2 Using Graphs for Question Answering

We propose the architecture of the QAS based on the common characteristics posed by regulation documents, as well as the kind of questions and answers expected by the user. Regulation texts have a defined structure, they are composed of chapters, and these, in turn, are subdivided in articles. There exists a relationship among articles of a single regulation text and also among articles of different document.

The questions we focus on consist mainly on asking if it is possible to perform certain action or not, for example:

*Question: Is it possible to award a honorable mention to a bachelor if he chose to graduate using the qualification option?*

For that kind of questions, a final ‘yes’ or ‘no’ is expected. However, particularly for the legal domain, such a short answer is not desirable, because the answer should be supported by its corresponding articles. For this reason, we focus in providing the set of articles from which it is possible to infer the final answer. For the previously stated question, the general lawyer of the National Polytechnic Institute, answers<sup>2</sup>:

*Answer: As stated by the Professional Graduation Regulation from the National Polytechnic Institute, the following can be concluded: The option of graduating by qualification proceeds if the student’s average is higher than 9.0 and all of his subjects were approved in an ordinary way (Chapter II of “On Graduating Options”, article 13). The candidate can only aspire to the award of honorific mention, if, in addition to covering other requisites*

---

<sup>1</sup> For example, the TREC ([trec.nist.gov](http://trec.nist.gov)) and CLEF ([www.clef-campaign.org](http://www.clef-campaign.org)) competitions.

<sup>2</sup> [www.abogadogeneral.ipn.mx/faqdlc.html](http://www.abogadogeneral.ipn.mx/faqdlc.html)

disposed in this regulation, he presents professional exam (Chapter VII, “On the Professional exam”, *article 43*). Graduating by qualification does not require presenting the professional exam, so that it cannot be included within the article 43 fraction II of the mentioned Regulation; when this option is chosen, the candidate cannot obtain honorific mention.

We can see from this answer that it consists of the articles which support it, as well as the reasoning to obtain it. In our case, the developed system was designed to return, as previously mentioned, only the set of articles which eventually contain those which answer the posed question. Because our system does not return a completely logically evaluated answer such as ‘yes’ or ‘no’, it is considered as part of the Shallow Question Answering Systems.

It is considered that graphs are structures capable of coding naturally the similarity relationships between textual units, and also, its utility in natural language processing tasks has been verified [11]. Because of this, our system uses as fundamental component the representation of a set of documents as a graph where each node corresponds to an article of the set of documents, and the values associated to the edges correspond to the degree of similarity between them. Using this representation we pretend to reflect the structure of the regulation texts, and in turn, use it for the extraction of the articles which constitute the answer to a question to the system.

The fundamental parts of the system are shown in the graph in Figure 1. Question pre-processing consists on constructing the query based on the question, and Answer Extraction consists on adding the generated query as two new nodes (A and B). Then the shortest path between A and B is sought. We will show that this path contains articles highly related to the question, and they share certain degree of similarity between them.

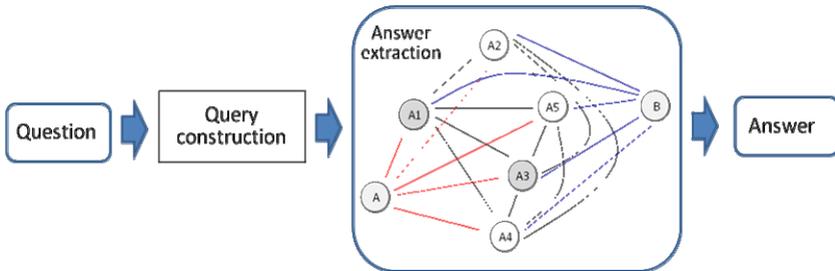


Fig. 1. General architecture of the proposed Question Answering System

### 3 System Description

#### 3.1 Document Collection Preprocessing

The first step is a pre-processing of the set of documents which consists on articles (1698 articles in total) from 26 different regulations: First, convert to lowercase, eliminate punctuation symbols, roman numbering, Arabic numbers, hyphens, isolated chars from a to z; then, eliminate stopwords. This step was optional, and we conducted our tests with both keeping and removing stopwords.

### 3.2 Graph Description

The documents (articles of regulations) were represented as vectors, following the Vector Space Model [12, 13]. Each term was weighted by the TF-IDF measure (Term Frequency—Inverse Document Frequency). See equations (1), (2) and (3).

$$tfidf = tf_{t,j} \cdot idf_i \tag{1}$$

$$tf_{t,j} = \frac{n_{i,j}}{\sum n_{k,j}} \tag{2}$$

Where  $n_{i,j}$  corresponds to the number of occurrences for each term from the article  $a_j$  and the denominator represents the occurrence of all terms in the article  $a_j$

$$idf_i = \log \frac{|A|}{|\{a_j : t_i \in a_j\}|} \tag{3}$$

Where  $|A|$  is the total number of articles in the document collection and  $|\{a_j : t_i \in a_j\}|$  is the number of articles where the term  $t_i$  appears.

Finally, a graph was constructed for each document collection, see Figure 2. Each node represents an article of a regulation text, and the associated values to the edges  $V_{i,j}$  between each pair of nodes represents the inverse value of the standard similarity cosine measure.

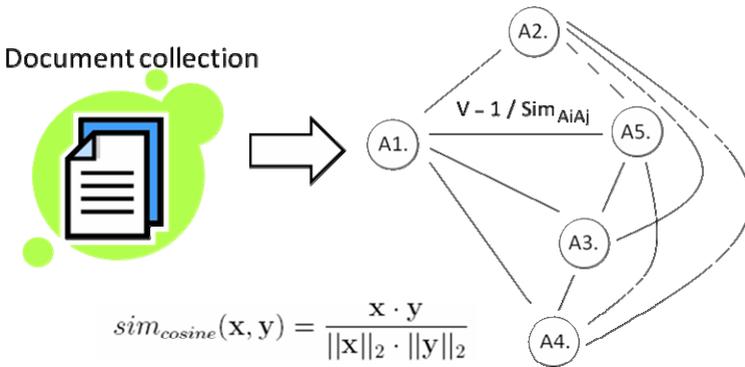


Fig. 2. Representing of the document set as a graph

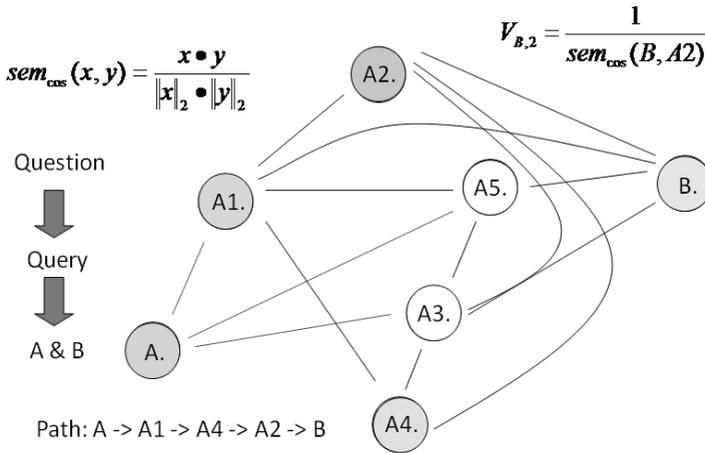
### 3.3 Query Building Module

The question is pre-processed and then integrated into the graph in the same way that the regulation collection. Each question is converted to lowercase, punctuation symbols are eliminated (parenthesis, hyphens, numbers, etc.). Stopwords are kept (QK) or removed (QR) and finally, words that do not exist in the document collection are

removed—this is equivalent to finding a similarity measure of 0 for them. The weighting values are calculated with regard to the document collection (See eq. (2)).

### 3.4 Answer Extraction Module

From the query of the previous module, two new *articles* are added to the graphs (Figure 3). The answer extraction consists on finding the paths of minimal weight using the Dijkstra algorithm: Once the first minimal path is found, the nodes which constitute it are eliminated, and the Dijkstra algorithm is run again until the graph becomes disconnected for the pair of nodes which constitute the query. The *answer paths* are ordered from less to more weight, and are returned to the user with the text of each article corresponding to the node in the regulation collection.



**Fig. 3.** A user question is split in two parts, which are added in the same way that the set of articles

## 4 Evaluation

The evaluation of the QAS is based on the following criteria:

- i* Relevance: The output of the QAS should answer to the questions of the kind “yes or no”; this is measured by determining if the articles that the general lawyer considers to produce an answers were returned by the system.
- ii* Noise degree in the answer: Alien and irrelevant information that the system returns is quantified.

To implement the measure of these both criteria, we use the following procedure: Initially, the answer is limited to 100 articles. It is unlikely that the answer is found after such number of articles. The returned articles were divided in groups of 5 (the maximum number of articles that can be found in a single answer of the lawyer). Each group was given the value  $V_i$ , following equation (4)

$$V_i = 1 - \frac{(n-1)}{N}, \quad (4)$$

where  $n$  is the group number (the first 5 articles constitute the group 1, the next 5, group 2, etc).  $N$  is the maximum number of packages that can be found ( $N=20$ , as  $5 \cdot 20=100$  articles).

Finally, each article returned for a determined answer is graded using the following expression:

$$\text{Grade } CR_i = \frac{\sum_{i=1}^n nV_i}{n_{AR}}, \quad (5)$$

where  $CR_i$  is the grade assigned to the answer of the  $i$ -th question,  $n$  is the package number where the answer-article was found, and  $n_{AR}$  is the number of answer-articles found.

The final mark of the system is the average of the grades  $CR_i$ , i.e.:

$$\text{Grade } CS = \frac{\sum_{i=1}^{np} CR_i}{np}, \quad (6)$$

where  $np$  is the total number of questions for evaluation.

## 5 Experiments and Results

For this report, our system was tested with 21 questions. We tested against a basic Information Retrieval System (IRS) based on the vector space model. This IRS uses the same set of vectors used for the construction of the graph, but this time they were directly compared with the cosine measure with the query. The vectors which are more similar to the query are returned, so that the output of this system is the set of articles relevant to the query. The results were then compared with the results of our QAS.

We performed several experiments varying different parameters of the QAS:

- a. Keeping (K) and removing (R) stopwords.
- b. Different schemes of automatic question division in two parts (A and B) when converting the query.

As we mentioned in Section 3.3, we tested with keeping and removing stopwords. This yields two derived regulation document collections: DCK (Document collection keeping stopwords) and DCR (Document collection removing stopwords), and two kinds of queries, QK and QR which correspond to keeping or removing stopwords in the query.

Additionally, we performed four experiments with regard to the automatic division of the query in part A and part B. For example, consider the following question:

*Cómo se lleva a cabo el procedimiento de elección de representantes alumnos ante el consejo técnico consultivo escolar?* ‘What is the procedure for electing the student representative at the Technical Advisory School Council?’

**Table 1.** Descriptions and name of each experiment. R and K refer to *Remove* or *Keep* stopwords in the document collection or the query, respectively. H and M refer to *Half division* or *Mixed division* of the query, respectively. H' and M' refer to exchanging nodes A and B in the query.

System	Document Collection	Division Type		Average Mark	Answers Found		
		Query QK	Query QR		Complete	Partial	Without ans.
QAS	DCR	H	-	0.4449	9	6	6
		M	-	0.4723	10	5	6
		H'	-	0.4445	10	5	6
		M'	-	0.4716	10	5	6
	DCK	H	-	0.5342	12	5	4
		M	-	0.4988	13	3	5
		<b>H'</b>	-	<b>0.5351</b>	<b>12</b>	<b>5</b>	<b>4</b>
		M'	-	0.5023	13	3	5
		-	H	0.4851	10	5	6
		-	M	0.4865	10	6	5
		-	H'	0.4858	9	6	6
		-	M'	0.4889	10	6	5
IRS	DCK	-	-	0.2253	3	5	13
	DCR	-	-	0.1892	4	6	11

After the procedure described in sections 3.3 and 3.4, the query is divided in two new *articles* (nodes) A and B with the following contents, according to one of the four followings types of division:

**Half Division (H):** Node A will contain the left half of the question, and Node B will contain the rest. If the number of words in the query is odd, the Node A will contain one word more than Node B. (*Node A:* lleva cabo procedimiento elección representantes alumnos, *Node B:* ante consejo técnico consultivo escolar)

**Mixed Division (M):** In this type of division, terms are mixed: odd words are in Node A and even words are in Node B. (*Node A:* lleva procedimiento representantes ante técnico escolar, *Node B:* cabo elección alumnos consejo consultivo)

**Reversed Half Division (H'):** As in Half Division (H) but the contents of node A and B are exchanged: (*Node A:* ante consejo técnico consultivo escolar, *Node B:* lleva cabo procedimiento elección representantes alumnos)

**Reversed Mixed Division (M'):** As in Mixed Division (M) but contents of the Node A and Node B are exchanged. Even words are in Node A and odd words are in Node B. (*Node A:* cabo elección alumnos consejo consultivo, *Node B:* lleva procedimiento representantes ante técnico escolar)

We tested every combination of these parameters, which yielded 12 experiments. Table 1 shows the description and name of each experiment. We compared the

performance of our system with two experiments based in the document collection DCK (Document Collection Keeping stopwords) and DCR (Document Collection Removing stopwords). These experiments are shown in the last two rows of Table 1.

## 6 Conclusions and Future Work

Our proposed graph-based QAS has noticeably better performance with regard to the traditional model for IR; notwithstanding, the marks obtained in Table 1 show that our system still has a lot of room for improvement, considering the performed experiment which provides answers compared to those given by the general lawyer of our Institute to a set of 21 questions.

The experiments show that all marks of the QAS are between the range of 0.4445 and 0.5351. This is a very small variation, meaning that the impact of varying the division type does not have a relevant impact, *i.e.*, the arranging of the nodes has a small effect for the architecture we used.

With regard to keeping or removing stopwords, we found that keeping stopwords yields a small improvement in the performance of the system, being this only a small change.

It is important to note that these results cannot be directly compared with similar systems, because usual competitions such as TREC or CLEF consider that the answer is contained in several text fragments, whereas our system considers that answers that are not contained specifically in one or more snippets, but in their relationship—consider the example presented in Section 2: a traditional answer might be “Article 13” or “Article 43”, but in this case, the answer is not found completely in one or another, but in the combination of both.

Future work includes using natural language techniques such as lemmatization and using related words from a thesaurus to smooth the similarity measures in order to improve the rank of the answer-articles.

## References

1. Hirschman, L., Gaizauskas, R.: Natural Language Question Answering: The View From Here. *Natural Language Engineering* 7(4), 275–300 (2001)
2. Hoojung, C., Song, Y.-I., Han, K.-S., Yoon, D.-S., Lee, J.-Y., Rim, H.-C.: A Practical QA System in Restricted Domains. In: *Workshop on Question Answering in Restricted Domains*. 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain, pp. 39–45 (2004)
3. Erik, T., Sang, K., Bouma, G., de Rijke, M.: Developing Offline Strategies for Answering Medical Questions. In: *Workshop on Question Answering in Restricted Domains*. 20th National Conference on Artificial Intelligence (AAAI 2005), Pittsburgh, PA, pp. 41–45 (2005)
4. Fabio, R., Dowdall, J., Schneider, G.: Answering questions in the genomics domain. In: *Proceedings of the ACL 2004 Workshop on Question Answering in Restricted Domains*, Barcelona, Spain, pp. 46–53 (2004)
5. Niu, Y., Graeme, H.: Analysis of Semantic Classes in Medical Text for Question Answering. In: *Workshop on Question Answering in Restricted Domains*. 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain, pp. 54–61 (2004)

6. Zhuo, Z., Sylva, L.D., Davidson, C., Lizarralde, G., Nie, J.-Y.: Domain-Specific QA for the Construction Sector. In: Workshop of IR4QA: Information Retrieval for Question Answering, 27th ACM-SIGIR, Sheffield (July 2004)
7. Paulo, Q., Rodrigues, I.P.: A question-answering system for Portuguese juridical documents. In: Proceedings of the 10th international conference on Artificial intelligence and law. International Conference on Artificial Intelligence and Law, Bologna, Italy, pp. 256–257 (2005)
8. Paulo, Q., Rodrigues, I.P.: A collaborative legal information retrieval system using dynamic logic programming. In: Proceedings of the 7th international conference on Artificial intelligence and law. International Conference on Artificial Intelligence and Law, Oslo, Norway, pp. 190–191 (1999)
9. Doan-Nguyen, H., Kosseim, L.: The problem of precision in restricted-domain question-answering. Some proposed methods of improvement. In: Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004), Barcelona, Spain, pp. 8–15.
10. Diekema Anne, R., Yilmazel, O., Liddy, E.D.: Evaluation of restricted domain question-answering systems. In: Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain, pp. 2–7 (2004)
11. Rada, M.: Random Walks on Text Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 249–262. Springer, Heidelberg (2006)
12. Manning Christopher, D., Schütze, H.: Foundations of Statistical Natural Language processing. MIT Press, Cambridge (1999)
13. Salton, G., Wong, A., Yang, C.S.: A vector Space Model for Automatic Indexing. Information Retrieval and Language Processing (1975)
14. Moldovan, D., Clark, C., Harabagiu, S., Maiorano, S.: COGEX: a logic prover for question answering. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Edmonton, Canada, vol. 1, pp. 87–93 (2003)