

# Metagnostic Deductive Question Answering with Explanation from Texts

John Kontos, Joseph Armaos, and Ioanna Malagardi

National and Kapodistrian University of Athens  
University Campus,  
Ano Iisia 15771 Athens, Hellas  
ikontos2003@yahoo.com, iarmaos@otenet.gr,  
imalagar@di.uoa.gr

**Abstract.** The present paper presents a system called AMYNTAS for “metagnostic” deductive question answering from texts. This system can logically combine information from texts and answer questions generating explanations for its operation exhibiting “self-awareness”. The deductions are performed directly with the natural language text without previous translation into a formal representation. The “metagnostic” effect is accomplished by representing and processing the state of linguistic processing and reasoning of the system. The system is implemented in Prolog and uses a text grammar to parse sentences that contain the information being sought. The system uses reasoning rules, lexicon, ontology, prerequisite knowledge and the history of its state. The system may easily be adapted to completely different domains such as biomedical texts and texts of the proofs of theorems of Euclidean geometry. An evaluation performed with real sentences from these two completely different domains gave satisfactory results of accuracy and facility of domain adaptation.

**Keywords:** deductive question answering, question answering from texts, inference with texts, self awareness, explanation.

## 1 Introduction

In the present paper we propose the use of the term “metagnostic” instead of the term “metacognitive” used in [1] for computer systems that exhibit self-awareness. The reason for using this neologism is that we want to avoid any confusion that may arise from the psychological connotations of the adjectives “metacognitive” and “self-aware” that may be connected with human metacognition [2].

The nature of the behaviour of this kind of computer systems is manifested by the fact that the explanations they generate refer explicitly to the history of their linguistic processing and reasoning.

Our system AMYNTAS (Automatic MetagnostiCoN Trainingable Answering System) generates explanations either directly or after applying deductive inference and using meta-information concerning the state history of the system. Deduction is accomplished without translation of the texts into any formal representation following the ARISTA method [4],[5].

## 2 Related Work

The system AMYNTAS presented in the present paper answers questions by combining information from unstructured natural language texts and justifies answers to the questions by generating an explanation that exhibits behaviour inspired from human self-awareness. It is a system related to the system AROMA [5] that was listed in [3] as the sole inference-based biological question answering system in existence at the time of the review. The system AROMA used analysis of textual rhetoric relations for model-based question answering for explanation generation that AMYNTAS does not do at present. However AROMA lacked the main capability of AMYNTAS for generating explanations inspired from human self-awareness.

To the best of our knowledge the only modern systems in existence and current development that attempt to exhibit behaviour inspired from human self-awareness are the CASSIE system [6] and the EPILOG system [7]. However these two systems extract information solely from formal knowledge bases and they do not perform inferences and generate explanations directly from text as our system does.

## 3 The System Implementation and Operation

The question answering system AMYNTAS presented here was implemented in Prolog and consists of six modules implemented as separate programs totalling about 50 pages of code. These modules communicate through some temporary files that store intermediate results. The six modules are: the question processing module, the text pre-processing module, the ontology extraction module, the shallow parsing or text chunking module, the question answering module and the metagnostic processing module. The architecture of the system AMYNTAS is shown in Fig. 1.

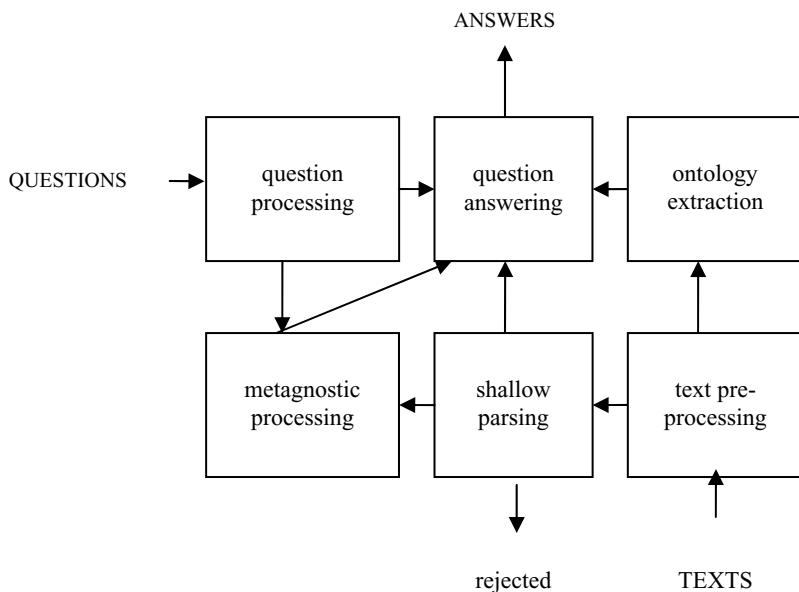
The question processing module extracts information from the input question. The information extracted is a list consisting of the entities mentioned in the question and the relation that connects them. For example in the question “what influences p53” the entity is p53 and the “blank” entity standing for the unknown entity that is sought and the relation is “influence”.

The text pre-processing module represents each word of a sentence as a fact with three arguments the first being the word itself, the second being the identifier of the sentence and the third being the position of the word in the sentence counting from left to right.

The ontology extraction module locates linguistic patterns in the input text corpus that may be used to extract automatically meronymic and taxonomic knowledge that may be used at question answering time.

The shallow parsing or text chunking module locates a verb related to the relation contained in the question and extracts the two substrings of the text sentence being analyzed that appear to the left and the right of the verb and end at some stop-word or punctuation mark.

The question answering module finds the answer to the question from the pre-processed text. The question answering module accepts questions that potentially require the combination of facts with the use of prerequisite knowledge for answering them. The prerequisite knowledge available to our system includes ontological



**Fig. 1.** The architecture of the AMYNTAS system

knowledge, inference rules and synonyms of the named entities involved of the domain which used in order to combine two or more facts mentioned in the text corpus.

At question answering time three looping operations are taking place. The basic loop concerns the search for an entity in a chunk related to the relation of the question. The second loop concerns the transformation of the list obtained from the question by following a particular strategy from the explicit list given to the system. The third loop searches for chains of facts using the matching of named entities occurring in the right part of one fact and the left part of another fact.

**Table 1.** The roles of the arguments of the predicate “strategy”

Variable Name	Role
Synonym	flag determining whether synonyms will be used or not
LSide	flag determining searching for a synonym at the left side of the verb
RSide	flag determining searching for a synonym at the right side of the verb
Inversion	flag determining whether the two sides of the text sentence i.e. to the left and to the right of the verb will be inverted or not
Relstrategy	specification of the relation denoted by the verb of the question
Pk	flag determining whether prerequisite knowledge will be used or not for answering the question
Strategy	the name of the strategy being applied

The metagnostic processing module collects information during the execution of all three loops and stores it either in an internal database or an external file. The metagnostic processing module also processes the history of state of the system in order to supply data for the generation of the explanations.

The strategies for processing the lists obtained from the transformation of the questions and the prerequisite knowledge provided is defined by the predicate: “strategy(Synonym, LSide, RSide, Inversion, Relstrategy, Pk, Strategy)”.

The roles of the variables for each argument of the predicate “strategy” are given in Table 1. An example will be used to illustrate the operation of our system using text fragments from two abstracts found in PubMed that concern the very important proteins p53 and mdm2 for carcinogenesis. This operation includes the question answering process of the system and the manifestation of “self-awareness” or “metagnosis” in the explanation generated.

The first abstract consists of six sentences from which the following two sentence fragments were selected automatically using the entities p53 and/or mdm2 as keywords:

- [1] “The p53 protein regulates the mdm2 gene”
- [2] “regulates both the activity of the p53 protein”

The second abstract consists of seven sentences from which two were selected from which the following two fragments were selected automatically using again p53 and/or mdm2 as keywords:

- [3] “The mdm2 gene enhances the tumorigenic potential of cells”
- [4] “The mdm2 oncogene can inhibit p53-mediated transactivation”

The question answering module answers questions deductively by combining sentences with common entities. More formally given two sentences “A rel1 B” and “B rel2 C” it finds that “A rel3 C” where what rel3 is generated depends on some prerequisite knowledge about the result of combining the relations rel1 and rel2.

A form of the questions accepted by the question processing module is “What is <verb> by <entity> ?” where <verb> stands for the passive form of a member of the group of verbs known to our system as significant verbs and <entity> stands for a noun denoting a named entity.

Examples of named entities are proteins and genes such as p53 and mdm2. Examples of verbs are the verbs regulate, enhance, influence and inhibit.

Another form of question that is accepted by our system is “Why <entity1> <verbphrase><entity2>” where entity1 and entity2 are named entities and <verbphrase> is standing for a relation and may stand e.g. for “is equal to”.

The example question “What influences p53” is input to the system and it is answered using the above sentence fragments. The relation “influences” is not symmetric and therefore none of the above mentioned 16 strategies is useful for answering this question. Another strategy (strategy17) is used which is appropriate for questions with asymmetric relations. The answer in this case after applying strategy17 is “p53 influences p53” i.e. the system detects a causal chain that forms a closed feedback loop based on the sentence fragments [1] and [4]. The verb “influence” is given to the system as common hypernym of the two verbs or relations “regulates” and “inhibits”.

The entity mdm2 is detected in the two sentence fragments “The **p53** protein **regulates** the **mdm2** gene” and “The **mdm2** oncogene can **inhibit p53** mediated transactivation” giving the causal chain “regulates+inhibit”.

Given that a deduction rule is also known to the system stating that “if rel1=regulate(s) and rel2=inhibit(s) then rel3=influences” the chain may be conflated to rel3 i.e. “influences”. This deduction rule enables the system to compose the two facts and generate an answer based on the recognition of the causal chain formed by the two sentence fragments [1] and [4].

The analysis performed by the question processing module extracts the components of the question such as the verb denoting a relation namely “influenced” in the example question and the entity such as “p53” for the example question. These components are stored in the internal data base.

## 4 The Application to Two Unrelated Illustrative Example Domains

The detailed operation of the system and the explanations generated when answering questions by extracting information from texts belonging to two unrelated illustrative example domains is presented in this section. The two illustrative domains are a biomedical and a geometrical domain.

The biomedical example uses a subset of the example text briefly analysed above. A more detailed analysis for this example is given in this section.

In the biomedical domain the system is able to identify the sentences of the input text that contain causal information about the p53 protein. Two example sentences with their identification numbers 1411 and 1421 are:

<1411>: <the p53 protein regulates the mdm2 gene>  
 <1421>: <the mdm2 oncogene can inhibit p53 mediated transactivation >

The example question “What influences p53” is also input to the system.

When the system processes the first sentence the explanation as verbatim generated by our system runs as follows:

I searched the input for the entity <p53>.

in the chunk <the p53 protein> of the sentence <1411>.

Since its first token is not an entity, I tested the rest of the tokens.

Repeating the search I found that

the entity <p53> is one of the tokens of the chunk <the p53 protein>  
 which is the chunk to the left of the verb of the sentence <1411>.

I found that the chunk to the right of the verb of the sentence <1411>  
 is the chunk <the mdm2 gene > and

Since its first token is not an entity, I tested the rest of the tokens.

Repeating the search the entity <mdm2> is one of the tokens of the chunk <the mdm2 gene >

which is the chunk to the right of the verb of the sentence <1411>.

I found that the entity <p53> is included in the sentence

<1411>:<the p53 protein regulates the mdm2 gene >

When the system processes the second sentence namely the sentence  
 <1421>: <the mdm2 oncogene can inhibit p53 mediated transactivation >

the explanation as verbatim generated by our system runs as follows:

I found that the entity <mdm2> is one of the tokens of the chunk <the mdm2 oncogene can>

which is the chunk to the left of the verb of the sentence <1421>.

I found that the chunk to the right of the verb of the sentence <1421>  
 is the chunk <p53 mediated transactivation >

and <p53> is the first token of the chunk

the entity <p53> is one of the tokens of the chunk

<p53 mediated transactivation >

which is the chunk to the right of the verb of <1421>.

And now the system detects a chain formed by <1411> to <1421> and back to  
 <1411> and explains as follows:

In my effort for answering the question “What influences p53”

I found from the following sentences:

<1411>: <the p53 protein regulates the mdm2 gene >

<1421>: <the mdm2 oncogene can inhibit p53 mediated transactivation >

That p53 influences p53

because influence is hypernym of regulate and inhibit.

From which it follows that there is a loop for <p53>.

In the geometrical domain the system is able to identify the sentences of the input text that logically justify a statement included in the English translation [8] of the text of the proof of the first Proposition from Euclid's Elements. For illustration of the answering of questions from the text of the proof of the first Proposition of the Elements is sketched below. The prerequisite knowledge we use involves various kinds such us Euclid's common notions, postulates and definitions. The first Proposition of the Euclid's Elements may be stated in English: “construct an equilateral triangle on a given finite straight line”.

The equilateral triangle abc is constructed by the points of intersection of two equal circles and their centres a and b where these centres lie on each others circumference and the points a and b are the two endpoints of the given finite straight line. The Euclid's Elements text includes a proof that the triangle constructed in this way is equilateral.

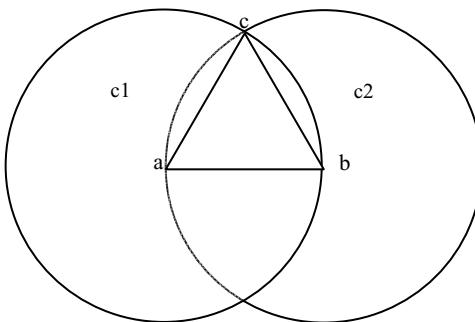
We use the text of this proof as a text base for answering deductively questions and generating explanations concerning the justification of statements present in the proof text. The meaning of the geometrical entities involved in this Proposition may be understood by using the diagram shown in Figure 1.

Three illustrative questions are answered by the system by analyzing automatically the corresponding Euclidean proof as follows:

Question 1: “why is side ab equal to side ac?”

Answer 1: “because they are radii of the same circle c1”

Question 2: “why is side bc equal to side ab?”



**Fig. 2.** The equilateral triangle abc constructed by the straight segment ab and by the point of intersection c of the two equal circles  $c_1$  and  $c_2$  with centres a and b

Answer 2: "because they are radii of the same circle  $c_2$ "

Question 3: "why is side ac equal to side bc?"

Answer 3: "because each of ac and bc are equal to ab"

These answers are only partial. The full answers contain explanations too.

The operation of the system in the case of the Geometry domain will be first explained by showing how Question 1 namely "why is side ab equal to ac" is processed by the system. The answer to this question is found by following strategy1. This strategy uses the expression of the question as it stands without any transformation. It should be noted that no prerequisite knowledge is used when strategy1 is followed. The answering of other questions may follow other strategies that apply various transformations of the questions such as inversion of its terms and replacement of an entity by each synonym. More specifically the inversion of "ac is equal to ab" is "ab is equal to ac" and the synonym of "ac" is "ca".

The name of the current strategy used is stored in an internal data base by the metagnostic sub-module so that the system is aware of the strategy being followed at each step. The chains of the proof steps as mentioned in the proof text are also stored so that they may be used at explanation time.

The original or transformed question is compared with the sentences of the text of the proof. When a match is found an answer is generated giving an explanation that justifies the answer. The transformed parts of the question are retrieved from the internal data base. The program finds the position of the left part of a proof step that matches the left part of the transformed question and continues by matching the relation and the right part of the question with some step of the proof.

Following the satisfaction of the matching operation the system tries the first strategy in the list and a sentence of this form is not found in the proof text. For this reason the system will try all other strategies in order to be able to answer the question. These strategies involve the replacement of entities with their synonyms, the inversion of the entities since the relation equal is symmetric and the use or not of prerequisite knowledge. Concerning the present question the strategy that succeeds is strategy16. This strategy involves the replacement of the two entities ac and bc with their synonyms ca and cb respectively, the inversion of the two entities and the use of prerequisite knowledge for answering the question. The prerequisite knowledge used is:

«things which are equal to the same thing are also equal to one another»

which is a statement called “common notion” in Euclid’s elements.

The answer generated by the system reports all the information it used to find the answer in order to justify it.

This answer runs verbatim as follows:

Your question is:

why  $\langle ac \rangle$  is  $\langle equal \rangle$   $\langle to \rangle$   $\langle bc \rangle$  ?

I analysed this question using strategy :  $\langle strategy16 \rangle$

which involves : mesynonyms, left, right, meinversion, equality, pk

My answer using Euclid's proof text offers the justification:

because

$\langle bc \rangle$  is equal to  $\langle ab \rangle$  and

$\langle ac \rangle$  is equal to  $\langle ba \rangle$

Furthermore because

$\langle ab \rangle$  and  $\langle ba \rangle$  are synonyms, which means that

$\langle ab \rangle$  and  $\langle ac \rangle$  are equal to each other

and because of the common notion

$\langle$  things which are equal to the same thing are equal to one another $\rangle$ .

## 5 The System Evaluation

The performance of the system was evaluated using a set of 127 sentences obtained from the PubMed Data Base that were selected from the titles of papers. The criteria of selection were that they contain the name of the protein p53 and the influence verb “enhance”. These 127 sentences are all that were found from PubMed on October the 7th 2008. This set constituted an input text to the system and was checked by the biologist Dr. Ourania Kosti. The results of evaluation of the parsing module quantified in terms of “precision” and “recall” are: Precision= 80% and Recall= 94 %, where precision and recall were computed by the formulae:

Precision=

# correct answers generated (True Positives) Divided by # total answers extracted  
(True Positives + False Positives)

Recall=

# correct answers generated (True Positives) Divided by # total correct answers identified by the biologist (True Positives + False Negatives).

The flexibility of the system was proven by showing that it may very easily be adapted to a completely different domain such as the proofs of Euclidean geometry. The only changes necessary were the enrichment of the lexicon and the ontology so that it includes the entities, the concepts and the relations of the new domain.

## 6 Conclusion

The present paper presented a system for metagnostic question answering from texts. Two simpler versions of the system were presented in [1] and [9]. The present system can extract information from texts and answer questions and provide explanations that

exhibit self-awareness. This was accomplished by representing and processing its state of automatic text analysis which includes deductive reasoning performed directly with the text without any previous formal representation of its sentences. The explanations generated are offered in order to convince the user of the correctness of its output. Explanations are also provided for the sentences rejected because they do not state a relation of interest of the entity of interest and another entity.

The system was implemented in Prolog it consists of the six modules namely the question processing module, the text pre-processing module, the ontology extraction module, the shallow parsing or text chunking module, the question answering module and the metagnostic processing module and uses a question grammar, a lexicon, an ontology and a history of the state of the system.

The system presented in the present paper constitutes a system appropriate for metagnostic question answering in different domains. Future extensions of the system may include a feedback loop that suggests corrections and/or expansions of its knowledge bases in order to be able to process a larger variety of text sentences. Such a feedback loop will be based on the self awareness capability already exhibited by the present system.

The explanations generated by the system may also be used for the support of handicapped persons when they fail to communicate with the system due to their own reduced capability of phrasing a question.

**Acknowledgement.** We thank the biologist Dr. Ourania Kosti for the invaluable help that she generously provided for the evaluation of our system.

## References

1. Kontos, J., Armaos, J.: Metacognitive Question Answering from Euclid's Elements Text. In: Proceedings of HERCMA 2007, Athens, Hellas (2007)
2. Maniadakis, M., Tani, J.: Acquiring Rules for Rules: Neuro-Dynamical Systems Account for Meta-Cognition. *Adaptive Behavior* 17(1), 58–80 (2009)
3. Athenikos, S.J., Han, H.: Biomedical Question Answering: A survey. *Comput. Methods Programs Biomed.* (2009) (in press)
4. Kontos, J.: ARISTA: Knowledge Engineering with Scientific Texts. *Information and Software Technology* 34(9), 611–616 (1992)
5. Kontos, J., Malagardi, I., Peros, J.: Question answering and rhetoric analysis of biomedical texts in the AROMA system. In: Proceedings of the 7th HERCMA: Hellenic European Conference in Computer Mathematics and Its Applications, Athens, Greece (2005)
6. Shapiro, S.C., Rapaport, W.J., Kandefer, M., Johnson, F.L., Goldfain, A.: Metacognition in SnePS. *AI Magazine* 28(1), 17–31 (2007)
7. Morbini, F.: Computational Infrastructure for a Self-Aware Agent. Ph.D. thesis, University of Rochester, Rochester, New York (October 2009)
8. Heath, T.L.: *The Thirteen books of Euclid's Elements*. Dover, N.Y. (1956)
9. Kontos, J., Armaos, J., Malagardi, I.: Metagnostic Question Answering from Biomedical Texts. In: HCI, San Diego, pp. 332–336 (2009)