

Article

# Medi-Test: Generating Tests from Medical Reference Texts

Ionuț Pistol \*, Diana Trandabăț \* and Mădălina Räschip \*

Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași, Iași 700483, Romania

\* Correspondence: ipistol@info.uaic.ro (I.P.); dtrandabat@info.uaic.ro (D.T.); mraschip@info.uaic.ro (M.R.)

Received: 6 November 2018; Accepted: 11 December 2018; Published: 19 December 2018



**Abstract:** The Medi-test system we developed was motivated by the large number of resources available for the medical domain, as well as the number of tests needed in this field (during and after the medical school) for evaluation, promotion, certification, etc. Generating questions to support learning and user interactivity has been an interesting and dynamic topic in NLP since the availability of e-book curricula and e-learning platforms. Current e-learning platforms offer increased support for student evaluation, with an emphasis in exploiting automation in both test generation and evaluation. In this context, our system is able to evaluate a student’s academic performance for the medical domain. Using medical reference texts as input and supported by a specially designed medical ontology, Medi-test generates different types of questionnaires for Romanian language. The evaluation includes 4 types of questions (multiple-choice, fill in the blanks, true/false, and match), can have customizable length and difficulty, and can be automatically graded. A recent extension of our system also allows for the generation of tests which include images. We evaluated our system with a local testing team, but also with a set of medicine students, and user satisfaction questionnaires showed that the system can be used to enhance learning.

**Keywords:** e-learning; automatic test generation; medical ontology; data mining for medical texts

## 1. Introduction

The recent availability of significant text resources for the health domain has opened the door for a new research direction in the natural language processing area, namely the adaptation of existing technologies and resources to the particularities of the medical domain. We now enjoy the availability of resources such as free research articles (like PubMed (<https://www.ncbi.nlm.nih.gov/pubmed/>)), free courses (such as those available in Coursera (Coursera Medicine Section: <https://www.coursera.org/courses?languages=en&query=medicine>)) or even specialized Wikipedia (Wikipedia Medicine section: <https://en.wikipedia.org/wiki/Medicine>) articles and various references. In this context, it becomes possible to adapt established techniques, originally developed to be used for the general language, to particular goals specific to this domain.

Two such methods, concerning the area of Natural Language Processing (NLP), are the development of ontologies and the automatic generation of evaluation tests from supporting resources.

Current e-learning platforms offer increased support for student evaluation, with growing interest in exploiting automation in both test generation and evaluation. Generating questions to support learning and user interactivity has been an interesting and dynamic topic in NLP since the availability of e-book curricula and e-learning platforms. From the earlier working systems [1,2] to the more recent examples [3–5], the typical method is to use a knowledge component and a support domain resource to identify information and to use it to generate a natural language question (usually also a correct answer).

Existing systems generally work by first identifying some concepts relevant for the domain, by using TF-IDF [6] or similar algorithms. Then, the identified concepts are used, in combination with simple patterns, to identify questions and answers in a source text. Originally, only multiple-choice questions and rudimentary question generation methods were used, while subsequently more types of questions and more complex methods to build and re-formulate questions are incorporated.

A domain ontology can significantly improve the quality of generated questions, just as the inclusion of images extracted from source materials increase the attractiveness of the tests. The development of medical ontologies has started more than 20 years ago, including LinKBase, with a knowledge base of over 1 million language-independent medical concepts [7]; GALEN and the “Galen-Core” high-level ontology for medicine [8]; the ON9 medical ontology [9] with its own methodology; Foundational Model of Anatomy—a domain ontology that represents a coherent body of explicit declarative knowledge about human anatomy [10]; Gene Ontology [11], aiming to produce a controlled vocabulary that can be applied to all organisms even as knowledge of gene and protein roles in cells is accumulating and changing, etc. The multitude of medical ontologies becomes an obstacle in selecting an ontology fitted for a specific problem, as well as in data integration. Unified Medical Language System [12] and OBO Foundry (Open Biomedical Ontologies) [13] are projects started to counteract this problem. Our proposal is to build a custom ontology starting from a medical textbook, and to try to enhance it with relations extracted from various other upper-level ontologies.

The generation of questions from text identified in an image is based on optical character recognition (OCR). The OCR method is used to convert printed text such as scanned documents, digital images, and PDF files into editable and searchable text data. There are many applications of OCR in domains like education, healthcare, insurance, and legal industries. Examples of applications include: extracting text from scanned documents, recognizing handwritten characters, license plate recognition, etc. The problem is difficult due to the different print quality of the source documents and the error-prone pattern matching. A survey of techniques used in optical character recognition can be found in [14]. Various approaches, such as matrix matching, feature extraction, structural analysis, or neural networks were developed for the design of OCR systems [15]. Many recognition systems are available, but few of them are open source and free. Tesseract [16] is one of the most accurate open-source optical character recognition engines. Tesseract 4.0 adds a new OCR engine based on long short-term memory (LSTM) networks [17]. It has support for English and other additional languages, including Romanian, which is why it was our choice for the system we built.

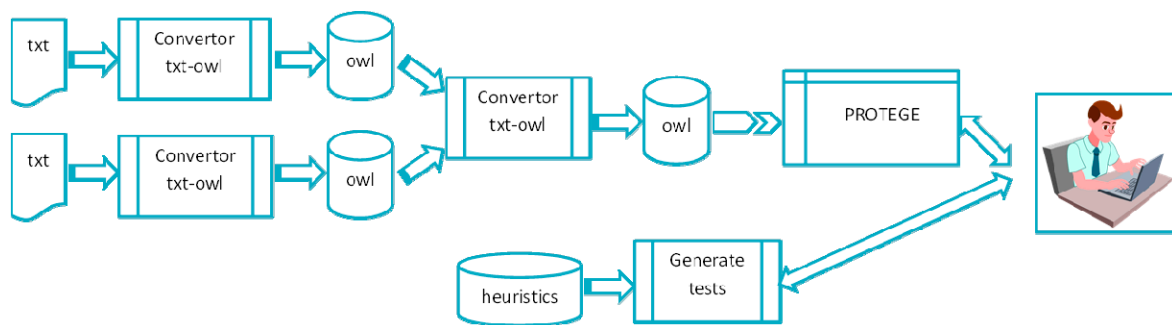
In this context, our system intends to automatically generate test questions from raw medical texts, in several main steps: (1) create an ontology from a raw medical textbook/combine several ontologies; (2) generate test questions based on the ontology; and (3) generate additional test questions based on images from the textbooks.

The next section presents the overall architecture of our system. Section 3 discusses the way in which the reference texts were processed in order to identify terms and relation candidates and the way in which those were used to build a domain ontology. Section 4 shows how the ontology, as well as the reference texts was used to identify possible questions as well as the correct and incorrect answers and Section 5 shows how those were used to generate and evaluate a customizable test. Section 6 describes the identification of relevant images in the reference texts and their usage in generated questions. Section 7 presents some conclusions and proposed future work.

## 2. Medi-Test Architecture

The Medi-test system is based on a modular architecture, presented in Figure 1. It has two main modules, one dealing with developing an ontology from various knowledge sources, and the other dedicated to creating tests. The first module is divided into three steps: generating ontology from a reference text (the submodule marked with 1 in the figure below), combining multiple such generated ontologies (submodule 2), and allowing them to be viewed or edited in Protégé (submodule 3). The

second module generates test questions and answers, process images to be transformed into evaluation images, and groups all these elements together to create a test (submodule 4 in the architecture).

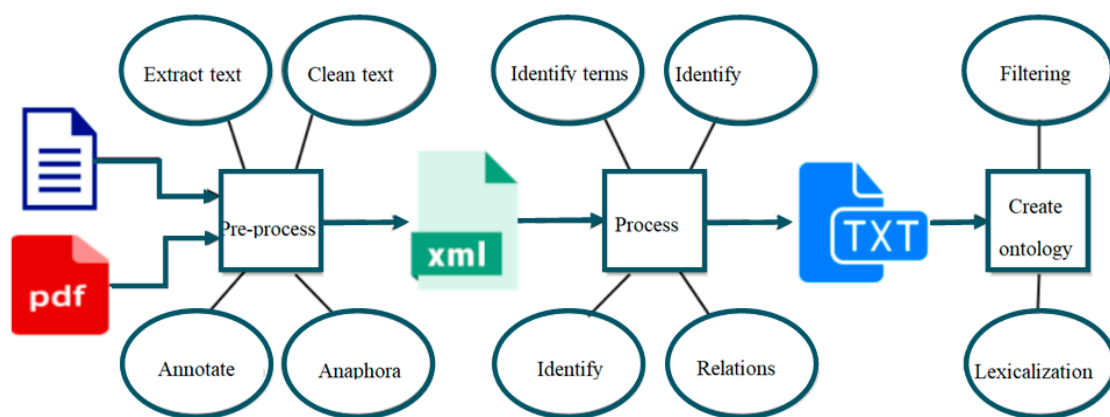


**Figure 1.** Architecture of the Medi-test system.

Medical knowledge and patient data are communicated by physicians in a very specialized language. In order to facilitate a computational processing of this language, terminological vocabulary and relations between concepts can be recorded in the form of ontologies. These resources, once created, can also be used by students of the medical schools for getting familiar with the specific language. This is exactly what Medi-test does, as explained in the following sections.

### 3. From Text to Ontology

The input format for Medi-test is raw texts in various formats (.doc, .pdf) taken from different sources: PubMed, Coursera, Wikipedia, or medical literature. These texts need to be transformed into an ontology, before being sent to the test generation module. Figure 2 describes the architecture of the ontology builder module.

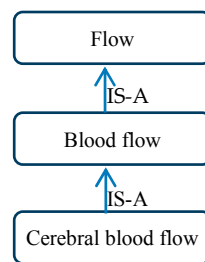


**Figure 2.** Description of the first ontology generation module.

The pre-processing stage involves extracting the text from various sources and formats. The extracted texts need then to be cleaned and annotated with lemma, part of speech, and named entities. This step is important because the ontology cannot include inflected word forms, nor parts of speech other than nouns. However, verbs and adjectives are very useful in identifying different relations or properties for the ontology concepts. Another important step was identifying and solving pronominal anaphora, since references are as frequent in medical texts as in any kind of text. The output of the pro-processing submodule is saved in an .xml file.

The most common semantic relation used in ontologies is the generic inheritance relation, i.e. the IS-A relation, indicating that the left member is a more specialized concept included in the more

general right-most concept. An example of such a relation can be “heart” IS-A “circulatory system”. Additional examples are shown in Figure 3.



**Figure 3.** Example of the IS-A relation.

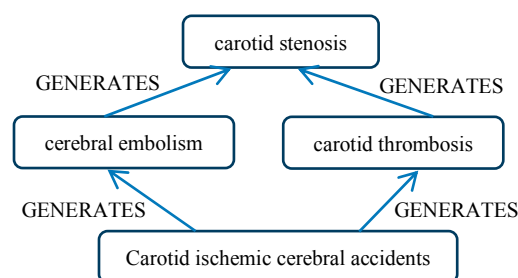
This .xml output is fed to the processing submodule, mainly responsible for the identification of concepts and instances, properties and relations. The identification of terms and instances is based on part of speeches. Thus, noun phrases have been considered to be terms/concepts for the ontology. One of the main challenges here was the handling of multiple concepts in the same noun phrase, such as:

*[Cerebral [blood [flow]]],*

a complex noun phrase containing two other noun phrases, “flow” and “blood flow”. This imbrication is very important for proper identification of the IS-A relation, being for this example:

Another method for the identification of the IS-A relation is based on definitions. Manual abound in definitions, introduces by typical expressions such as: “is defined by”, “represents”, “occurs when” etc.

For the identification of properties, parts of speech are again essentials, since noun phrases formed by noun plus adjective are indicators of properties, as in “red blood cell”, where “red” identifies as the *color* property of “blood cell”. In order to determine the property type (i.e., *color* in this example), adjectives are classified in various categories based on their use context: color, density, hardness, etc. Examples of a relation determined by verbs are shown in Figure 4.



**Figure 4.** Example of relations based on the identification of different verbs.

Besides the IS-A relation discussed above, our ontology building module also identifies the EQ-relation based on synonymy. Thus, concepts are merged through the EQ relation if they are identified as synonyms using external dictionaries. At this step, duplicated synonyms are removed, so that the ontology only contains one example of each synonym sets. Other considered relations are the one suggested by verbs, such as the ones in the following example:

“The symptomatology of carotid stenosis is due to cerebral embolism or carotid thrombosis which it can generate and which is at the basis of carotid ischemic cerebral accidents”.

Since we have different source texts from which we can create ontologies, we also included an ontology merging module in the Medi-test application which combines the owl files of several ontologies using existing APIs. For this aim, all [term-relation-term] pairs are checked for duplicated synonyms. Additionally, since the number of relations was increasing exponentially when merging

ontologies, we only kept the following relation types in the merged ontology: equivalence (based on synonymy), contains (based on enumerations, expressions such as “has components”, “includes”, “is composed of” etc.), generates (based on expression indicating a cause/effect), and IS-A (based on complex noun phrases and definitions). Since the owl format of the ontology is difficult to be read by medical students, we offer the possibility to visualize and edit the merged ontology in Protégé.

This section presented the first main module, responsible for building an ontology using manuals and other knowledge sources. The next section will introduce the test generation module, by discussing the submodule responsible for creating questions and answers.

#### 4. Generating Questions and Answers

The second main module of Medi-test generates tests in three steps: first, a set of questions and answers are created. Then, images are processed to be included in the evaluation tests. In the final step, tests are generated using the output of the previous two steps and a series of parameters set by the user.

The first step aims to produce a comprehensive description of candidate question topics, by identifying the following attributes:

- *name*: the name as it appears in the source text;
- *id*: the id of the corresponding ontology term, if it exists;
- *relations*: the id and members of any relations presents in the ontology;
- *domain*: the relevant domain which carries over to the generated questions;
- *paronyms*: alternative words to be used in question/answers generation;
- *synonyms*: alternative words to be used in question/answers generation;
- *description*: the context in which this topic appears in the source texts (multiple entries, if found).

The *name*, *relation*, and *domain* attributes were extracted from the ontology, *domain* being considered as the second level term in the *is-a* relation hierarchy, if above the current term, otherwise the current term was preserved as *domain*. The *paronym* and *synonyms* attributes are extracted from a freely available lexical resource for Romanian, DexOnline (<https://dexonline.ro/>), but can also be extracted from the equivalence relation in the ontology. A sample of the data used to generate questions/answers can be seen in Figure 5.

Using the above data, four types of questions were considered for automatic generation: multiple-choice, fill in the blanks, yes/no questions, and matching patterns.

For **multiple-choice** questions, a set of patterns have been developed to generate them using the ontology build by the previous main module. Examples of such patterns are:

- If the ontology contains C2 in a IS-A relation to C1, it generates questions such as: “Which of the following is an example of C1?” with C2 being one of the options for selecting the answer.
- The definition is used to generate Wh-questions, such as “What is the organ used by a human or animal being to detect sounds?”
- The other incorrect options are generated for each question using paronyms of C2 or other terms in the ontology that have no direct relation with C1 or C2.

**Fill in the blanks** questions are also generated from the definitions, relying on the syntactic annotation of the texts, by removing either the subject or the direct complement.

**Yes/no questions** are generated from the definition by either transforming it in a question directly or by negating the verb.

**Matching questions** are built from the relations marked in the ontology. If two terms are connected by a relation, they are selected as members of each set, with the other members being different terms in the ontology connected by the same relation.

id: 100	id: 200
name: ear	name: external ear
linked_nodes: [200,300,400]	linked_nodes: [500]
relations: [contains]	relations: [contains]
domain: ear	domain: ear
synonyms: [hearing analyzer]	synonyms: []
is_concept: True	is_concept: True
definitions: is the organ used by a human or being to detect sounds	definition: is the visible part of the ear

Figure 5. Sample of the data available for candidate terms.

For all questions, terms may be changed with their synonyms.

Since answers may be used in various questions, even of different type (an answer may be correct for some questions and incorrect for others), we stored a table of answers, where each entry is identified by a question\_id (the same for all answers for the same question), a body (the actual answer to be offered as option), and an is\_correct tag, identifying whether this answer is correct (1) or incorrect (0) for the current question. A short excerpt from this table is given in Figure 6, in Romanian (the language for which the system was tested).

id	question_id	body	is_correct
1158	1038	Cavitatea superioara stanga a inimii	1
1159	1038	Valva bicuspidă este alcătuită din două cuspe: una...	0
1160	1038	Are forma unui cub, prezentând un perete lateral, ...	0
1161	1038	Peretele posterior prezintă o proeminență ce poart...	0
1162	1039	Cavitatea superioara dreapta a inimii	1
1163	1039	La nivelul peretelui inferior se pot identifica: o...	0
1164	1039	Este o regiune de materie cenușie gasita intr-una ...	0
1165	1039	Fața pulmonară este orientată spre posterior și la...	0
1166	1040	Cavitatea inferoara stanga a inimii	1
1167	1040	Peretele anterior este neregulat, prezentând numer...	0
1168	1040	Este al doilea din cele trei oscioare ale urechii.	0
1169	1040	Un tip de fibre endogene	0
1170	1041	Cavitatea inferoara dreapta a inimii	1
1171	1041	Peretele anterior este reprezentat de orificiul at...	0
1172	1041	Peretele medial sau septal este reprezentat de sep...	0
1173	1041	Peretele superior nu prezintă elemente anatomice i...	0
1174	1042	Scheletul este o parte a corpului care are rolul d...	1
1175	1042	Un tip de fibre endogene	0
1176	1042	Peretele lateral are aspect neregulat și la acest ...	0
1177	1042	Inima este unul dintre cele mai importante organe ...	0
1178	1043	Citoscheletul este scheletul unei celule, care asi...	1
1179	1043	Element structural si functional de baza al sistem...	0
1180	1043	Fiecare din cele doua despartituri superioare ale ...	0
1181	1043	Este o structura osoasa care contine si protejeaza...	0
1182	1044	Acesta fiind un înveliș fluid și care se află ca a...	1
1183	1044	Un tip de fibre ascendente	0
1184	1044	Cavitatea inferoara stanga a inimii	0

Figure 6. Sample of the answers table.

For the same question, we can generate multiple versions, differing in ordering of the possible candidate answers. They are generated according to three difficulty levels attached to a set (questions-answers): easy, medium, and hard. The difficulty is measured automatically by considering:

- the inclusion of paronyms as alternative answers;
- the distance (by ontology relations) between answers;
- the Levenstein distance between the correct answer and the other options.

For each of the above possible difficulties, scores are computed, normalized, and averaged for each question. A question was considered easy if found in the bottom 33% of scores, hard if found in the top 33%, and medium otherwise. If more than one version of the same question exists for each difficulty level, alternative answers are changed until versions are created for all difficulty levels. For example, if we want to increase the difficulty of a question, we can include a paronym as an incorrect answer, select incorrect answers closer (in the ontology) to the correct answer or include terms (or paronyms) closer, according to the Levenstein distance, to the correct answer.

Some examples of questions and answers for each of the four types, all of medium difficulty, are given below.

*Multiple choice:*

Q: Which is part of the cranial box?

A: Intracranial space (Correct)

A: Left ventricle (Incorrect)

A: Conus medullaris (Incorrect)

A: Base (Incorrect)

*Fill in the blanks:*

Q: .... is the visible part of the ear.

A: Outer ear.

*Yes/no:*

Q: The endoschelet concept is identical to the internal skeleton concept.

*Matching:*

A: Ear, pinna.

A: Cranial box, cerebrospinal fluid.

A: Epidermis, melanocytes.

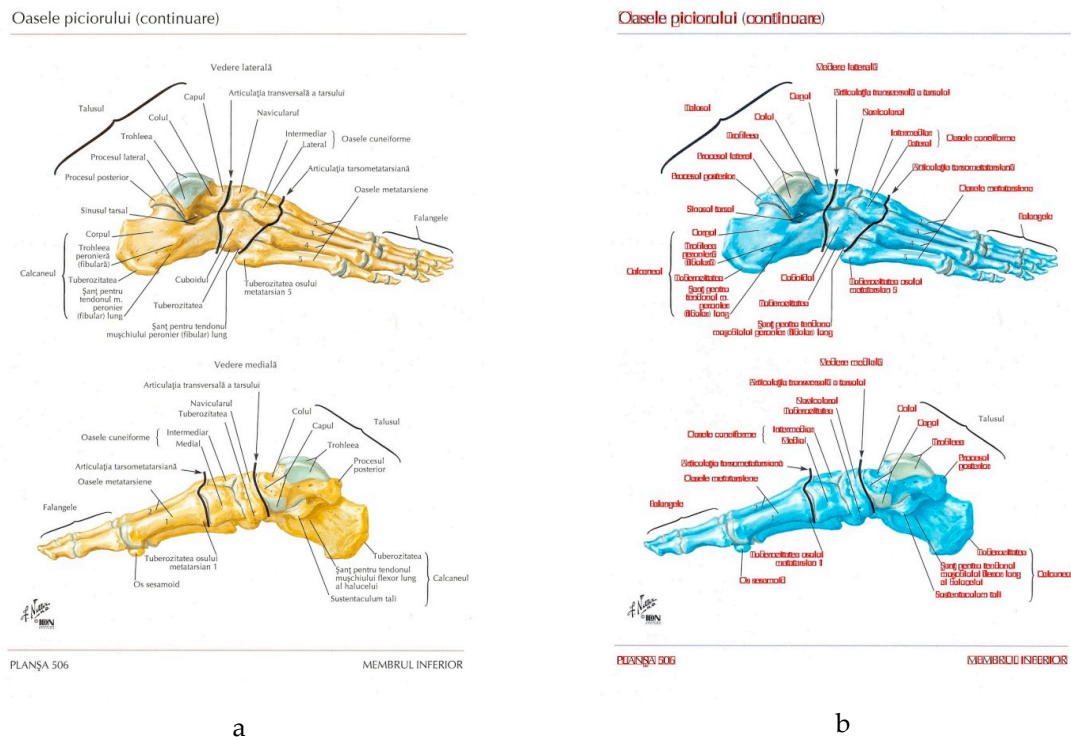
Although other types of questions were considered and could be generated from our data, such as factoid and definition, they were not added to our system in this version. The reason for not including them is that, while the automated evaluation of the answers is an important feature, it cannot be accurate for those types of questions, as they require a more complex semantic analysis of the answer in order to be accurately evaluated.

## 5. Adding Images in Questions

A further development of our system targeted the inclusion of images in some questions. The images can be found in the supporting reference documents.

The image processing module had the following tasks: (1) the acquisition and the preprocessing of medical images, (2) the identification of text from images, and (3) the generation and (4) validation of questions. The task of collecting medical images assumed the selection of images came from scanned documents like books, atlases, or courses, or from odf which had images presented as photos, not editable. Selected images were manually preprocessed in order to eliminate the header and footer, redundant text, etc.

To identify the text from the image, the Tesseract engine version 4.0 was used. Tesseract works in a step-by-step manner. We will use the image in Figure 7a to exemplify this process, an image detailing the foot bones, in Romanian. First, the images are converted into binary images. Then, the character outlines are extracted and gathered together into Blobs. Blobs are organized into text lines. Text is divided into words according to some definite and fuzzy spaces. Recognition then starts as a two-pass process. An attempt to recognize each word is made in the first pass; the words that passed satisfactory are given to an adaptive classifier as training data.

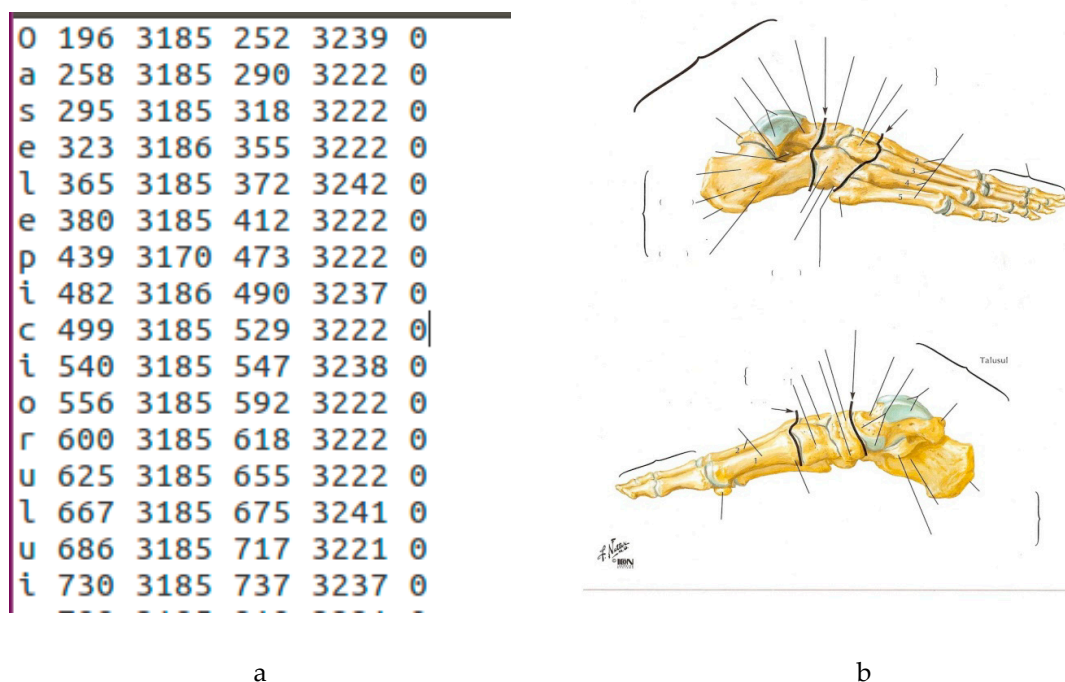


**Figure 7.** First two steps of processing by the Tesseract; (a) the original image (b) processed image, with outline for the identified characters.

The adaptive classifier tries to improve the accuracy in recognizing text. Words that were not recognized well enough are given for a second pass. Figure 7b presents the output Tesseract gives after this step.

Tesseract then generates .box files to specify the extracted text and their coordinates (see Figure 8a for an example). The text is then deleted from the image files in order to let the student fill in the empty spaces to check his knowledge. The output of Tesseract is the image without the text, and separately, the text extracted from the image. An example of an image processed by Tesseract is given in Figure 8b.





**Figure 8.** The processing made by the Tesseract engine, version 4.0; (a) the .box file; (b) the output image.

The extracted text is used to generate questions. In general, the questions are built using some predefined patterns. A particular type of question consists of identifying the elements presented in an image. The question is constructed by using the title identified by the OCR engine.

Examples of generated questions are:

Q: Identify the foot bones in the following picture.

The student will fill up the information. His answer will be checked against the correct answers stored in the system. The answer is considered correct if the two texts are similar. To verify the similarity, the texts are stemmed first and then the cosine similarity measure is computed.

Another type of question generated from images is fill in the blanks. The elements which must be filled in are generated by using the extracted text associated with the image. Example of such a question:

Q: Identify in the picture below the following terms: [a], [b]

where [a] and [b] could be, for example, the *femur* and the *coxal* bone.

Using the extracted texts, general questions can be created. Such questions do not use the image for the verification. Examples include multiple choice questions which check if an element is part of a system, based on the assumption that images usually explain a concept by presenting its components. The multiple choices are generated using keywords extracted from the current image, but also from other images (for incorrect versions of the answers). Such an example for the presented image is:

Q: Which muscle belong to the upper limb?

A: (a) Deltoid

(b) Triceps

(c) Trapezius

(d) Pectoralis minor

## 6. Creating Tests

An ontology including over 10,000 concepts and instances, with the four relation types between them, was developed following the process described in Section 3. Using this ontology and a set of 20 documents (mostly didactic materials) totaling over 800 million words, the process described in Section 4 was used to generate approximately 3200 answers and 1200 questions. The patterns used to discover possible questions/answers were selected to allow almost no ambiguity, thus the number of questions is relatively small compared to the size of the source documents. The number of actual possible questions is, however, much larger, as a question can be used with multiple answer sets and various difficulty levels.

Using the process described in Section 5, a further set of questions/answers was added. We considered a small number of images for generating questions. For each image, we generated three types of questions, as presented in Section 5. For example, for the image considered in Figure 7 for the fill-in-the-blank type of question, ten questions were generated (five with two terms and five with three terms).

A basic web interface was developed which uses the questions and answers produced to generate tests. No tests are pre-built, they are only generated by user request and according to their specified parameters:

- test duration
- test length (number of questions)
- test difficulty (easy, medium, or hard)

The default values for these three parameters are set at 30 min duration, 15 questions, and medium difficulty. If the user increases one or two parameters, the others are increased automatically. Similarly, a decrease in test time leads to a decrease either in difficulty or in the number of questions selected. Some examples of correlations between test parameters, depending on their values, are given below:

If the user sets duration at 60 min and length at 20 questions, the difficulty is automatically set at hard.

- If the user sets difficulty at easy and the other two parameters are left at their default values, then the system adjusts the length of the test at 20 min.

A medium difficulty test can also include questions of easy and hard difficulty, but the average difficulty will be medium. An easy difficulty test will not include hard questions, and reciprocally, a hard test will not include easy questions. A question, although it can have multiple difficulty levels (as described in Section 4), will not be included multiple times in the same test.

The test is automatically evaluated and a score is computed. The results are presented only as a final score, no correct answers are shown to the test takers. The answers given are stored by our system to be used for further fine-tuning of the question difficulty.

Some generated tests were evaluated by students in Medicine School, and their feedback was used to make minor corrections to the questions/answers and to fine-tune the difficulty of the generated tests.

## 7. Conclusions

As Medi-test is still in the development stage, even with virtually all functionalities working, no large-scale testing phase was carried out yet. Some tests have been performed in a limited context, involving part of the development team and medicine student volunteers. The user satisfaction rate was more than satisfactory, and minor improvement suggestions have been made.

The complexity of the system proved manageable, although it involved multiple NLP modules, as well as various resources adapted and several specially developed for our system. Further developments are still being carried out to improve the linguistic preprocessing steps and enrich the support ontology. Another important development considered is to extend Medi-test's functionalities

for languages other than Romanian, with English being the initial candidate due to the availability of resources and language familiarity. Most modules of Medi-test are language independent, so this addition is expected to be ready shortly.

A limitation of this automated test generation system is the reliance on high quality text sources, which is not always available for all topics. The reason for the requirement is the very strict patterns used to discover concepts, questions, and answers, needed in order to minimize the chance to produce erroneous results. A more flexible set of patterns and an ideally automated method of validating concepts/questions/answers candidates would reduce the size of the required text resources needed to produce varied tests for a topic.

Including other types of questions, such as factual or expository questions, would further improve our system. Generating these types of questions would be within the current capabilities of our system but evaluating their answers would require either human involvement or the inclusion of a semantic analysis module in our system.

Another significant benefit would be provided by allowing human experts to assist our system at various steps, from building the ontology to question/answers selection, steps which don't require human assistance in the current version of our system. However, human assistance would increase the specificity of the generated tests either directly, by writing/correcting questions, or indirectly, by improving the ontology (thus the quality of the automatically generated questions/answers).

**Author Contributions:** Formal analysis, validation and investigation, I.P., D.T. and M.R.; writing—original draft preparation, I.P.; writing—review and editing, D.T. and M.R.; visualization, M.R.; supervision, I.P. and D.T.; project administration, I.P.

**Funding:** This work was supported by a grant of the Romanian Ministry of Research and Innovation, PCCDI-UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/73PCCDI, within PNCDI III.

**Acknowledgments:** Part of the work described was done by third year Computer Science students as part of their seminar work for the Artificial Intelligence class of 2017-2018, supervised by the authors of this paper. The authors would also like to acknowledge the major role Dan Cristea had on designing the project's architecture and providing textual resources.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Mitkov, R.; Le An, H.; Karamanis, N. A computer-aided environment for generating multiple-choice test items. *Nat. Lang. Eng.* **2006**, *12*, 177–194. [[CrossRef](#)]
2. Sumita, E.; Sugaya, F.; Yamamoto, S. Measuring non-native speakers' proficiency of English by using a test with automatically-generated fill-in-the-blank questions. In Proceedings of the Second Workshop on Building Educational Applications Using NLP, Ann Arbor, Michigan, 29 June 2005; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005.
3. Chali, Y.; Hasan, S.A. Towards topic-to-question generation. *Comput. Linguist.* **2015**, *41*, 1–20. [[CrossRef](#)]
4. Labutov, I.; Basu, S.; Vanderwende, L. Deep questions without deep understanding. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; Volume 1.
5. Heilman, M.; Smith, N.A. Good question! statistical ranking for question generation. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, CA, USA, 2–4 June 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010.
6. Salton, G.; Christopher, B. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* **1988**, *24*, 513–523. [[CrossRef](#)]
7. Van Gorp, M.; Decoene, M.; Holvoet, M.; dos Santos, M.C. LinKBase, a Philosophically-Inspired Ontology for NLP/NLU Applications. In Proceedings of the KR-MED 2006 Workshop, Baltimore, MD, USA, 8 November 2006.

8. Rogers, J.; Rector, A. The GALEN ontology. In Proceedings of the Medical Informatics Europe (MIE 96), Copenhagen, Denmark, 19–22 August 1996; pp. 174–178.
9. Gangemi, A.; Pisanelli, D.M.; Steve, G. An overview of the ONIONS project: Applying ontologies to the integration of medical terminologies. *Data Knowl. Eng.* **1999**, *31*, 183–220. [[CrossRef](#)]
10. Rosse, C.; Mejino, J.L., Jr. A reference ontology for biomedical informatics: The Foundational Model of Anatomy. *J. Biomed. Inform.* **2003**, *36*, 478–500. [[CrossRef](#)]
11. Gene Ontology Consortium. The Gene Ontology (GO) database and informatics resource. *Nucl. Acids Res.* **2004**, *32* (Suppl. 1), D258–D261. [[CrossRef](#)] [[PubMed](#)]
12. Bodenreider, O. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucl. Acids Res.* **2004**, *32* (Suppl. 1), D267–D270. [[CrossRef](#)] [[PubMed](#)]
13. Smith, B.; Ashburner, M.; Rosse, C.; Bard, J.; Bug, W.; Ceusters, W.; Goldberg, L.J.; Eilbeck, K.; Ireland, A.; Mungall, C.J. The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.* **2007**, *25*, 1251. [[CrossRef](#)] [[PubMed](#)]
14. Impedovo, S.; Ottaviano, L.; Occhinegro, S. Optical character recognition—a survey. *Int. J. Pattern Recognit. Artif. Intell.* **1991**, *5*, 1–24. [[CrossRef](#)]
15. Chang, S.L.; Chen, L.S.; Chung, Y.C.; Chen, S.W. Automatic license plate recognition. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 42–53. [[CrossRef](#)]
16. Smith, R. An overview of the Tesseract OCR engine. In Proceedings of the IEEE International Conference on Document Analysis and Recognition (ICDAR 2007), Parana, Brazil, 23–26 September 2007; Volume 2, pp. 629–633.
17. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [[CrossRef](#)] [[PubMed](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).