

Generating Multiple-Choice Test Items from Medical Text: A Pilot Study

Nikiforos Karamanis
Computer Laboratory
University of Cambridge
CB3 0FD, UK
nk304@cam.ac.uk

Le An Ha and **Ruslan Mitkov**
Computational Linguistics Research Group
University of Wolverhampton
WV1 1SB, UK
{L.A.Ha, R.Mitkov}@wlv.ac.uk

Abstract

We report the results of a pilot study on generating Multiple-Choice Test Items from medical text and discuss the main tasks involved in this process and how our system was evaluated by domain experts.

1 Introduction

Although Multiple-Choice Test Items (MCTIs) are used daily for assessment, authoring them is a laborious task. This gave rise to a relatively new research area within the emerging field of Text-to-Text Generation (TTG) called Multiple-Choice Test Item Generation (MCTIG).¹

Mitkov et al. (2006) developed a system which detects the important concepts in a text automatically and produces MCTIs testing explicitly conveyed factual knowledge.² This differs from most related work in MCTIG such as Brown et al. (2005) and the papers in BEAUNLP-II (2005) which deploy various NLP techniques to produce MCTIs for vocabulary assessment, often using preselected words as the input (see Mitkov et al. for more extensive comparisons).

The approach of Mitkov et al. is semi-automatic since the MCTIs have to be reviewed by domain experts to assess their usability. They report that semi-automatic MCTIG can be more than 3 times quicker than authoring of MCTIs without the aid of their system.

¹TTG, in which surface text is used as the input to algorithms for text production, contrasts with Concept-to-Text Generation (better known as Natural Language Generation) which is concerned with the automatic production of text from some underlying non-linguistic representation of information (Reiter and Dale, 2000).

²Mitkov et al. used an online textbook on Linguistics as their source text. Clearly, their approach is not concerned with concepts or facts derived through inferencing. Neither does it address the problem of compiling a balanced test from the generated MCTIs.

Moreover, analysis of MCTIs produced semi-automatically and used in the classroom reveals that their educational value is not compromised in exchange for time and labour savings. In fact, the semi-automatically produced MCTIs turn out to fare better than MCTIs produced without the aid of the system in certain aspects of item quality.

This paper reports the results of a pilot study on generating MCTIs from medical text which builds on the work of Mitkov et al.

2 Multiple-Choice Test Item Generation

A MCTI such as the one in example (1) typically consists of a question or *stem*, the correct answer or *anchor* (in our example, “chronic hepatitis”) and a list of *distractors* (options b to d):

- (1) Which disease or syndrome may progress to cirrhosis if it is left untreated?
 - a) chronic hepatitis
 - b) hepatic failure
 - c) hepatic encephalopathy
 - d) hypersplenism

The MCTI in (1) is based on the following clause from the source text (called the *source clause*; see section 2.3 below):

- (2) Chronic hepatitis may progress to cirrhosis if it is left untreated.

We aim to automatically generate (1) from (2) using our simple Rapid Item Generation (RIG) system that combines several components available off-the-shelf. Based on Mitkov et al., we saw MCTIG as consisting of at least the following tasks: a) Parsing b) Key-Term Identification c) Source Clause Selection d) Transformation to Stem e) Distractor Selection. These are discussed in the following sections.

2.1 Sentence Parsing

Sentence Parsing is crucial for MCTIG since the other tasks rely greatly on this information. RIG employs Charniak’s (1997) parser which appeared to be quite robust in the medical domain.

2.2 Key-Term Identification

One of our main premises is that an appropriate MCTI should have a *key-term* as its anchor rather than irrelevant concepts. For instance, the concepts “chronic hepatitis” and “cirrhosis” are quite prominent in the source text that example (2) comes from, which in turn means that MCTIs containing these terms should be generated using appropriate sentences from that text.

RIG uses the UMLS thesaurus³ as a domain specific resource to compute an initial set of *potential key terms* such as “hepatitis” from the source text. Similarly to Mitkov et al., the initial set is enlarged with NPs featuring potential key terms as their heads and satisfying certain regular expressions. This step adds terms such as “acute hepatitis” (which was not included in the version of UMLS utilised by our system) to the set.

The *tf.idf* method (that Mitkov et al. did not find particularly effective) is used to promote the 30 most prominent potential key terms within the source text for subsequent processing, ruling out generic terms such as “patient” or “therapy” which are very frequent within a larger collection of medical texts (our reference corpus).

2.3 Source Clause Selection

Mitkov et al. treat a clause in the source text as eligible for MCTIG if it contains at least one key term and is finite as well as of the SV(O) structure. They acknowledge, however, that this strategy gives rise to a lot of inappropriate source clauses, which was the case in our domain too.

To address this problem, we implemented a module which filters out inappropriate structures for MCTIG (see Table 1 for examples). This explains why the number of key terms and MCTIs varies among texts (Table 2).

A finite main clause which contains an NP headed by a key term and functioning as a subject or object with all the subordinate clauses which depend on it is a source clause eligible for MCTIG provided that it satisfies our filters. Example (2) is such an eligible source clause.

³<http://www.nlm.nih.gov/research/umls/>

Structure	Example (key term in italics)
Subordinate clause	Although <i>asthma</i> is a lung disease, ...
Negated clause	<i>Autoimmune hepatitis</i> should not be treated with interferon.
Coordinated NP	Excessive salt intake causes <i>hypertension</i> and hypokalemia.
Initial pronoun	It associates with <i>hypertension</i> instead.

Table 1: Inappropriate structures for MCTIG.

Experimentation during development showed that our module improves source clause selection by around 30% compared to the baseline approach of Mitkov et al.

2.4 Transformation to Stem

Once an appropriate source clause is identified, it has to be turned to the stem of a MCTI. This involves getting rid of discourse cues such as “however” and substituting the NP headed by the key term such as “chronic hepatitis” in (1) with a wh-phrase such as “which disease or syndrome”. The wh-phrase is headed by the *semantic type* of the key-term derived from UMLS.

RIG utilises a simple transformational component which produces a stem via minimal changes in the ordering of the source clause. The filtering module discussed in the previous section disregards the clauses in which the key term functions as a modifier or adjunct. Additionally, most of the key terms in the eligible source clauses appear in subject position which in turn means that wh-fronting and inversion is performed in just a handful of cases. The following example, again based on the source clause in (2), is one such case:

- (3) To which disease or syndrome may chronic hepatitis progress if it is left untreated?

2.5 Selection of Appropriate Distractors

MCTIs aim to test the ability of the student to identify the correct answer among several distractors. An appropriate distractor is a concept semantically close to the anchor which, however, cannot serve as the right answer itself.

RIG computes a set of potential distractors for a key term using the terms with the same semantic type in UMLS (rather than WordNet coordinates employed by Mitkov et al.). Then, we apply a simple measure of distributional similarity derived from our reference corpus to select the best scoring distractors. This strategy means that MCTIs with the same answer feature very similar distractors.

Chapter	Words	# of Key-terms	# of Items	Usable Items	Usable Items w/out post-edited stems	Replaced distractors per term	Total Time	Average Time per Item
Asthma	8,843	9	66	42 (64%)	18 (27%)	2.0	140 mins	3 mins 20 secs
Hepatitis	10,259	17	92	49 (53%)	19 (21%)	0.9	150 mins	3 mins 04 secs
Hypertension	12,941	22	121	59 (49%)	15 (12%)	0.8	200 mins	3 mins 23 secs
Total	32,043	40	279	150 (54%)	52 (19%)	—	490 mins	3 mins 16 secs

Table 2: Usability and efficiency of Multiple-Choice Test Item Generation from medical text.

3 Evaluation

RIG is a simple system which often avoids tough problems such as dealing with key-terms in syntactic positions that might puzzle the parser or might be too difficult to question upon. So how does it actually perform?

Three experts in producing MCTIs for medical assessment jointly reviewed 279 MCTIs (each featuring four distractors) generated by the system. Three chapters from a medical textbook served as the source texts while a much larger collection of MEDLINE texts was used as the reference corpus.

The domain experts regarded a MCTI as *unable* if it could not be used in a test or required too much revision to do so. The remaining items were considered to be *usable* and could be post-edited by the experts to improve their content and readability or replace inappropriate distractors.

As Table 2 shows, more than half of the items in total were judged to be usable. Additionally, about one fifth of the usable items did not require any editing. The Table also shows the total number of key-terms identified in each chapter as well as the average number of distractors replaced per term.

The last column of Table 2 reports on the efficiency of MCTIG in our domain. This variable is calculated by dividing the total time it took the experts to review all MCTIs by the amount of usable items which represent the actual end-product. This is a bit longer than 3 minutes per usable item across all chapters. Anecdotal evidence and the experts' own estimations suggest that it normally takes them at least 10 minutes to produce an MCTI manually.

Given the distinct domains in which our system and the one of Mitkov et al. were deployed (as well as the differences between them), a direct comparison between them could be misleading. We note, however, that our usability scores are always higher than their worst score (30%) and quite close to their best score (57%). The amount of directly usable items in Mitkov et al. was between just 3.5% and 5%, much lower than

what we achieved. They also report an almost 3-fold improvement in efficiency for computer-aided MCTIG, which is very similar to our estimate. These results indicate what our work has contributed to the state of the art in MCTIG.

In our future work, we aim to address the following issues: (a) As in Mitkov et al., the anchor of a MCTI produced by RIG always corresponds to a key-term. However, the domain experts pointed out several cases in which it is better for the key-term to stay in the stem and for another less prominent concept to serve as the answer. (b) Students who simply memorise the input chapter might be able to answer the MCTI if its surface form is too close to the source clause so another interesting suggestion was to paraphrase the stem during MCTIG. (c) We also intend to introduce greater variability in our process for distractor selection by investigating several other measures of semantic similarity.

Acknowledgments

We are grateful to Tony LaDuca, Manfred Straehle and Robert Galbraith from the National Board of Medical Examiners (NBME) for their expert-based feedback and to three anonymous reviewers for their comments.

References

- BEAUNLP-II. 2005. Papers on MCTIG by Hoshino and Nakagawa, Liu et al., and Sumita et al. In *Proceedings of the 2nd Workshop on Building Educational Applications Using NLP*.
- Jonathan Brown, Gwen Frishkoff, and Maxine Eskenazi. 2005. Automatic question generation for vocabulary assessment. In *Proceedings of HLT-EMNLP 2005*, pages 249–254.
- Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of AAAI 1997*, pages 598–603.
- Ruslan Mitkov, Le An Ha, and Nikiforos Karamanis. 2006. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2):177–194.
- Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.