Multiple-choice test item generation: A demo

Le An Ha
Research Group in Computational Linguistics.
University of Wolverhampton.
Stafford Street, Wolverhampton,
WV1 1SB, UK
L.A.Ha@wlv.ac.uk

Abstract

In this paper, we discuss a demo of an NLP based multiple-choice test item generation system. As the methodology behind this system has already been discussed in several papers ([2], [3]), only practical issues related to the system will be presented here. We will discuss users' feedbacks of the initial system, which are then translated into additional features of the system. Some of these features, although are not NLP related, will make an NLP system more user-friendly. We will also present how Wikipedia is mobilised for the task of distractor selection. The utilisation of this popular resource would also make it easy to describe the system to a lay user.

Keywords

Multiple-choice test item generation, NLP user-friendly interface.

1. Introduction

Multiple-choice test item generation (MCTIG) is a new NLP application in which an NLP system is used to generate multiple-choice test items[1] automatically. The items are then presented to users to be post-edited. A methodology for the automatic generation of multiple-choice test items has been proposed by Mitkov and Ha (2003). In that work, an NLP engine is used to process the input texts, extract important terms, construct questions about those terms, and propose distractors that are semantically close to the correct answer (term). The purpose of using semantically close distractors is to prevent not-so-confident test takers from guessing. An example of an automatically generated multiple-choice test item using this methodology is:

> What does a prepositional phrase at the beginning of a sentence constitute?
>
> i. a modifier that accompanies a noun
> ii. an associated modifier
> iii. an introductory modifier
> iv. a misplaced modifier

This item is derived from the statement "a prepositional phrase at the beginning of a sentence constitutes an introductory modifier"[2] (Kies 2003). Using Kies's textbook of English grammar, the system identifies *introductory modifier* as a term, a concept which can be used to construct questions. After this, the system scans the textbook for statements suitable for generating questions (the example sentence beginning with "a prepositional…" is considered to be suitable). Then a set of transformation rules will be applied to turn the statement into a question. An example of these transformation rules is "SVO -> What does S V?" (S: subject, V: verb, O: object; these constituents are identified using a shallow parser), where the correct answer is O. In the sample sentence, S is *a prepositional phrase at the beginning of a sentence*, V is *constitutes*, and O is *an introductory modifier*, thus the question "What does a prepositional phrase at the beginning of a sentence constitute?" is formulated. The other three choices (*a modifier that accompanies a noun*, *an associated modifier*, and *a misplaced modifier*) are distractors that the system finds from the text and considers semantically similar to the correct answer (*an introductory modifier*).

The demo system has two phases: processing phase (Figure 2) and post-editing phase (Figure 3). In the processing phase, the system accepts a document from a user, and produce MCTIs from it (as described above). In the post-editing phase, the user will be able to pick usable items, and post-edit them further. NLP technologies are used in both phases.

The MCTIG engine has been subject to several development circles, and received many feedbacks from different users. In this paper, we focus on features which arise from users' feedbacks. These feedbacks often concentrate more on the interactivity of the system, and less on the actual performance of the NLP engine. This indicates that, perhaps, NLP technologies have been mature enough for certain NLP tasks, and the task now is to put these NLP technologies into actual use. It is thus important to find ways to present the outputs of these NLP technologies to the end users, so that they can work with, and benefit from them easily. These, in turn, would help to improve productivity.

[1] A multiple-choice test item comprises a question, the correct answer, and a set of distractors. (In some of our other studies, we use the term "multiple-choice question".)

[2] It should be noted that, at the moment, the system is unable to assess whether or not the statement "a prepositional phrase at the beginning of a sentence constitutes an introductory modifier" is a valid statement in terms of *content*. In other words, the system assumes that all input sentences are valid.

This paper is organised as follows: this Section introduce the MCTIG system, and the types of feedbacks we received. Section 2 discusses the profiles of users who have been providing these feedbacks and their different needs. Section 3 discusses the interaction between users and the system. Section 4 presents how different outputs of different NLP technologies are presented to users, facilitating the dissemination of these technologies. The paper finishes with conclusion and future developments. The actual demo will show the system in operation, and how the features mentioned in this paper function. The system will accept a document from a user and generate MCTIs. The user will be able to post-edit the results using a web-based interface. The post-edited results can then be exported into XML format. Several screenshots of the system are shown in this paper.

2. Users

Our users include experts in the development of multiple-choice tests and entrepreneurs who are experienced in developing specialised software. Two groups of users have provided two different perspectives for the system. The experts in multiple-choice tests often ask for features which are directly related to the generation of an MCQ item, e.g. the ability to change a distractor, and to edit the correct answer. The entrepreneurs, on the other hand, have asked for features which are related to the ability of the system to interoperate with others (e.g. the ability to process different document formats, the possibility to have a web service, or the ability to export the resulted MCTIs to XML format). We believe that both perspectives are very important for an NLP system: it is important to meet the requirement of end-users, who need to work with the system on a day-to-day basis, as well as system developers, who need to make sure the system can interoperate with other systems currently used by the organisation.

3. Interactivity

According to users' feedback, interactivity is among the most important features of the system. This is expected, as most of human-computer interfaces today are interactive, and a normal user would expect the same for an NLP system. Furthermore, interactivity will provide the users an opportunity to exploit the system more intuitively. In the developed system, interactivity is provided to the users in both processing and post-editing phrases.

3.1 Processing phase (Figure 2)

In processing phase, the system provides users information about the current status of the system, as well as the estimated time to the finish. In the web environment, auto-refresh is used to provide this feature. After the system finishes generating MCTIs, the results are shown to users grouped by topics. When the user moves the mouse over an item, that item will be fully shown in a tooltip (shown in Figure 2). This allows users to have an overall picture of what have been generated by the system. Clicking on a particular item will send the user to the post-editing interface.

3.2 Post-editing phase (Figure 3)

In post-editing phase, the system allows users to post-edit the item interactively. The users can easily edit the question wordings, and change distractors by ticking or unticking a distractor. The resulted item will be shown to the user in a WYSIWYG (What you see is what you get) manner (e.g. as soon as the user click on a distractor, the distractor will be shown in the item immediately), allowing them to review the item at anytime. Definitions of distractors, when available, will also be shown when the user moves the cursor over them (see the tooltip shown in Figure 1).

4. Presenting outputs of different NLP technologies

Several NLP technologies and resources have been integrated into the system: parsing, automatic term extractions, semantic similarity calculation, corpora and ontologies. In our opinion, presenting these outputs to end users in an appropriate way is also as challenging as developing technologies to produce them. For example, in automatic term extraction, the extracted list of terms is often sorted according to some scoring function, but to end users, the list of terms sorted in this way seems to be difficult to work with. As a result, a more logical way to present the list of extracted terms will be required. In the following sections, we discuss how the lists of terms and distractors are presented to users. We will also discuss the incorporation of Wikipedia, as a "user-friendly" resource in this application.

4.1 The list of terms

The list of terms identified by the system is organised into topics before being presented to users. The topic of a term is either its head (in the case of a multi-word term), or its hypernym (in the case of a single-word term). The list of topics are sorted according to the total frequency of terms belong to a topic. In this way, the most frequent topic in a text is shown first. The users' impression is that it seems to them that the system is able to identify the topics of a text, and the first several questions seem to be plausible. For each individual term, questions which either contain the term, or have it as the correct answer, will be shown (Figure 2).

4.2 Distractor list

Although the list of distractors is produced using semantic similarity calculation, we soon realise that presenting them in the order of similarity between a distractor and the correct answer is not the best way. For an end user, the list ordered by similarity seems to be difficult to use. Instead, it is indicated that the distractor list should be sorted alphabetically. In addition, it has also been suggested that distractor list should also be presented hierarchically to facilitate the manipulation of

item difficulty. In order to present distractors hierarchically, an ontology of the domain should be obtainable. Figure 1 shows how the distractor list of "asthma" can be arranged hierarchically.

4.3 Using Wikipedia

End users may not be familiar with NLP jargons and resources such as *ontology*, *terminology*, *corpora*, or *BNC*. As a result, it is important to employ resources which are both useful for the system, and familiar to them, so that they find the system friendlier. During some development circle, a user submits to the system several articles from Wikipedia ([4]). As a result, we decided to exploit Wikipedia in our system, as it would be easier to explain to users that some of the results are derived from this resource (as oppose to some other corpora, such as BNC [1], which may not be as well-known). Currently it (Wikipedia) is employed in the system as a source of distractors. Specifically, entries from Wikipedia which have the same head as the correct answer are used as its distractors. In this way, the pool of distractors is widening, enabling the users to have more choices. Moreover, the inclusion of some distractors which are obviously wrong can make an MCTI more amusing, and thus suitable for certain purposes such as TV quizzes[3]. For example, for the question "Which bonds are formed when a hydrogen atom is shared between two molecules?", "financial bonds" should be an obviously wrong answer, yet for certain types of quizzes, this type of distractors may be still useful.

5. Conclusions and future directions

This paper presents a demo version of an MCTI generation system. It concentrates on the user's feedbacks of the system rather than the technologies employed. The paper has argued that interactivity is very important for the users. The paper has also argued that attention should be paid on how to present NLP technologies to the users so that they can work with and benefit from them efficiently. The actual demo will show in detail how the system functions as well as different features presented in this paper in action. The system will accept a text document (in either plaintext or word format), and produce MCTIs. Users then will be able to post-edit the generated items, and export them into xml format for future use.

We intend to make the system more user-friendly, by integrating it into popular word processing programs such as Microsoft Word. In this way, a user can see potential MCTIs generated from a text while compiling it. We also want to develop a web service, so that the system can also be easily integrated into other systems.

We also want to perform user evaluation, in order to gain insights into how actual users can benefit from the developed system. We also want to test the system on different domains, in order to identify strengths and weaknesses of the system, and improve the system further.

[3] MCQs used in TV quizzes often include some obviously wrong distractors.

6. References

[1] BNC. 2001. The British National Corpus, version 2 (BNC World). Distributed by Oxford University Computing Services on behalf of the BNC Consortium.

[2] Karamanis, N., L. A. Ha, and R. Mitkov. 2006. Generating Multiple-Choice Test Items from Medical Text: A Pilot Study. In Proceedings of *INLG 2006*, Sydney, Australia.

[3] Mitkov, R., L. A. Ha, and N. Karamanis. 2006. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering* 12(2): 177-194.

[4] http://www.wikipedia.org

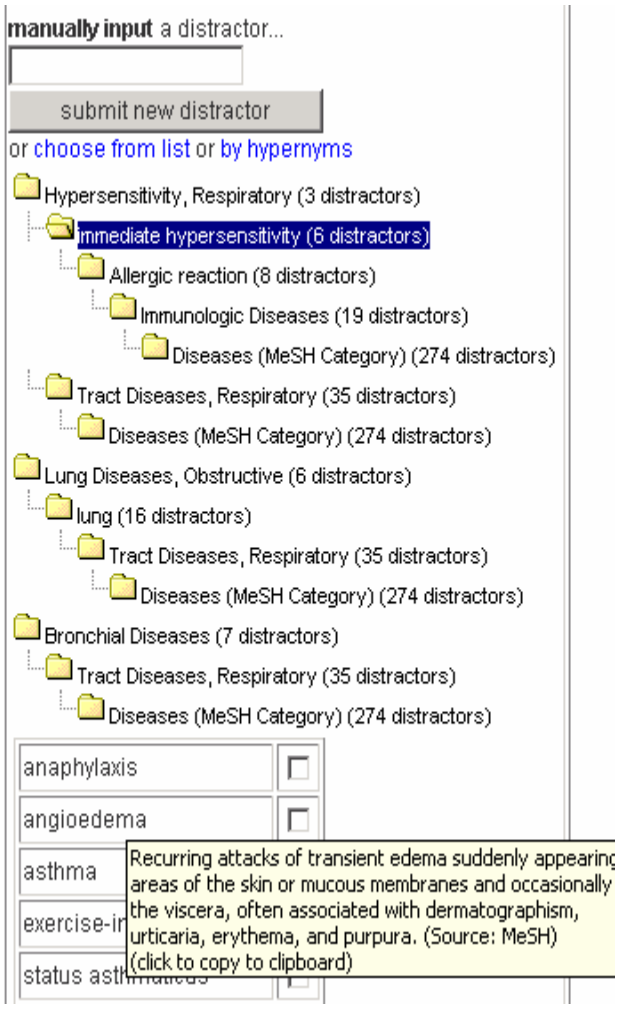

Figure 1: Distractors arranged hierarchically.

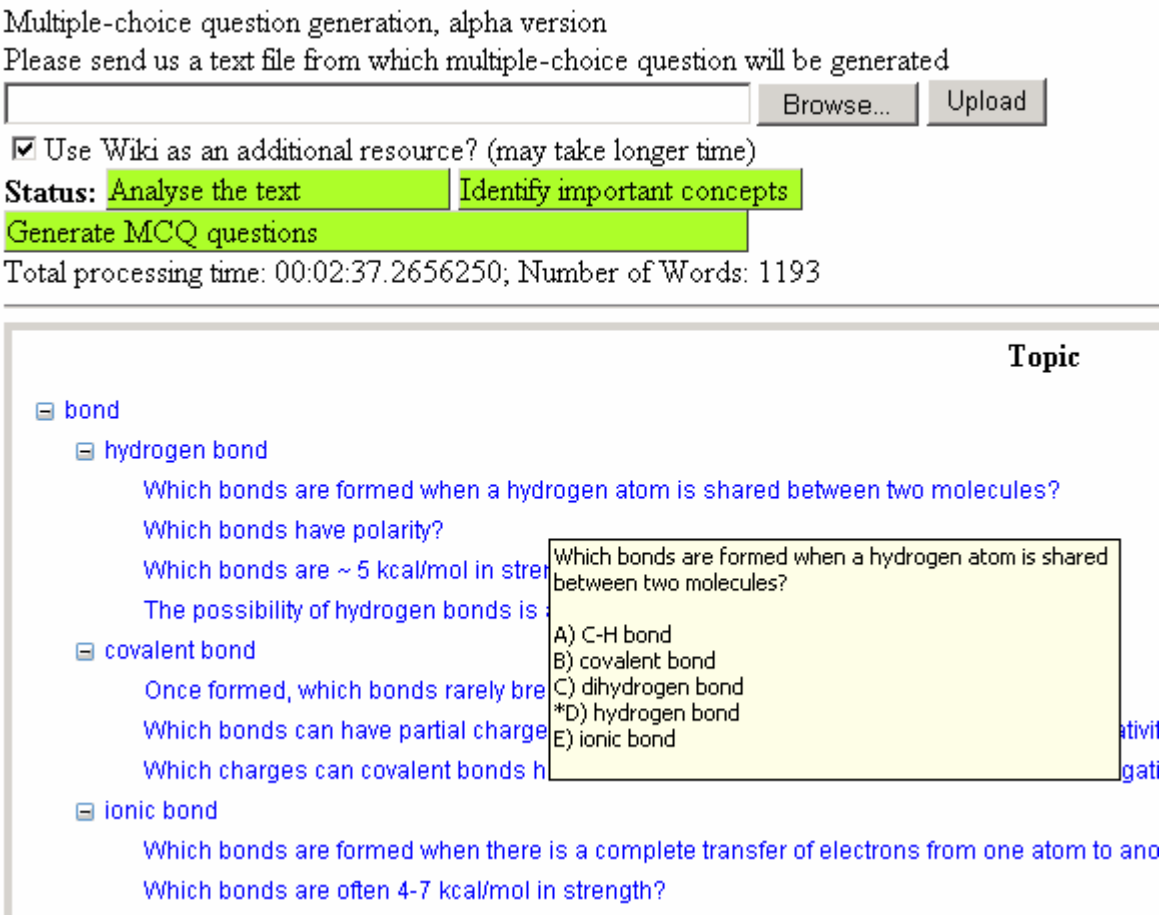

Figure 2: Processing phase

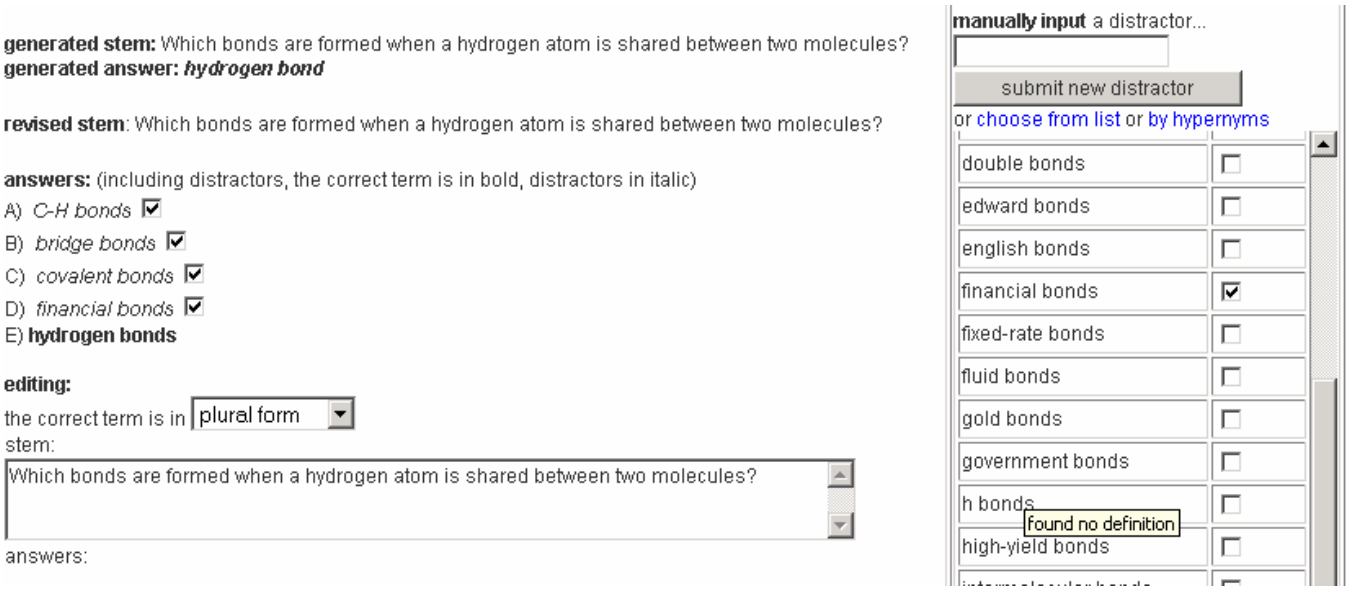

Figure 3: Post-editing phase