



Design of Corpus-Generated EFL Placement and Progress Tests for University Students

Marina Kustova¹

Abstract

The main aim of the paper is to introduce a tool which makes it possible to generate questions for an EFL placement test and for progress tests required in the course of Academic Writing in English on the basis of a learner corpus. The learner corpus in question – REALEC (Russian Error-Annotated Learner English Corpus) was set up at the School of Linguistics, a department of the Higher School of Economics. It comprises student essays written in preparation for IELTS examination as well as some other pieces of academic writing with mistakes outlined and expertly annotated by their English instructors. Sentences with mistake from this corpus have been used as a source of questions for the tests. The proposed system (whose working title is RETM – REALEC English Test Maker) through a reiterative procedure automatically generates a pool of tasks, and these tasks are then analysed by the expert, who makes a decision on their suitability for a particular test, and then those that are considered to be pedagogically appropriate are divided into three sets according to how difficult the required correction is going to be for a student. The last step in the expert's work is to choose a type of testing question for each item in the set. During the test, the correction offered by a test-taker is automatically compared with the correction that had been previously given by an annotator (often an EFL instructor). The suggestion is either automatically accepted or rejected, and, correspondingly, assigned or not assigned a score available at this level. Another interesting feature in RETM is that depending on the student's success in the previous question, the next question in the test will be either at a higher or lower level of difficulty, or will stay at the same level if the highest or the lowest level has been reached. Even though the process of setting up a test is not fully automated and requires human effort at some stages, it is still much more timesaving than the traditional ways of composing tests. In conclusion, I outline the points demonstrating advantages of this tool both for instructors and students.

1. Introduction

Developing placement and progress test for students learning English as a second language can be effort- and time-consuming for a range of reasons. For instance, the design of a progress tests is fraught with difficulties in choosing and/or making up texts or phrases in English, as well as deciding which questions reflect the part of the curriculum studied, and, finally, which types of testing questions are bound to engage the testee's language ability in the appropriate way [1]. It is a complicated and extremely time-consuming process. In the present paper I introduce a new approach to help English teachers in collecting questions for EFL tests. This tool is a system of automatic generation of sentences from the annotated corpus which will be used as testing questions. For this purpose I am using the learner corpus of works written by students of the Higher School of Economics (REALEC). The working title of the test-generating system is RETM, which stands for REALEC English Test Maker.

2. Ideology

One of the main difficulties one encounters when composing an EFL test is the choice of questions to be included in it. It does not seem an easy task to prepare a pool of questions which would, on the one hand, let the students practice their weakest skills, and, on the other, correspond to their level of proficiency in English they need to demonstrate on the test. The system I am introducing is capable of satisfying both needs.

The corpus I use as a source is made up of papers written by HSE students learning English as a second language. Mistakes in the papers have been annotated and corrected by the students' EFL instructors working at the same institution. Since all the mistakes that served as a source of test questions had been made by students themselves, it can be claimed that the tests which are built up with those questions are in a way targeted to reveal the weak points in the areas of English students have to master [2].

¹ National Research University — Higher School of Economics, Russia



Not all the mistakes, however, are equally easy to correct, which makes the automatically generated test questions differ one from another in terms of difficulty. Due to this fact, it is necessary to introduce an important manual stage in building a test out of automatically generated questions, namely, the one involving the expert's choice. Before the students get the questions that are automatically generated, the expert is supposed to throw out the questions which do not seem suitable for any possible reason, and assign the difficulty level to all the others. This is how the automatically built tests are supposed to correspond to the testee's level of English.

3. Technical details

3.1 Brat and REALEC annotation scheme

According to [3], "annotation... is the process of manually or automatically adding information into text for a given purpose". The codification of this information is called the annotation scheme [3]. Before proceeding to the description of the ways RETM itself functions, it seems useful to give a brief overview of the annotation scheme applied in the corpus.

In REALEC, students' papers are annotated at the following levels: grammatical error (e. g. wrong choice of article or inappropriate verb agreement), the supposed cause of the error (e. g. L1 influence), and linguistic and pragmatic damage caused by the error (which can be minor, moderate or major) [4]. Every mistake outlined by the expert in addition to being corrected gets the annotation on the three levels. Each of the levels bears a significant piece of information; however, only the layer which contains the information about the supposed type of the error is engaged in the task generation.

3.2 Question generation

REALEC (Russian Error-Annotated Learner English Corpus), which served as a source of data for generating test questions, is using brat rapid annotation tool [5] for outlining and annotating mistakes. Brat annotations are stored in a standoff format; in other words, the annotations are stored separately from the original text files which are annotated [2]. Basically, a single annotation file corresponds to one text file with the same name. Consequently, the first thing that RETM does is finding every pair of text and annotation files in the working folder. All the further manipulations are performed on every single text-annotation file pair.

According to the REALEC annotation scheme, there are hierarchical classes and subclasses of types of errors; not all of them are equally suitable for test questions. For each type of test, sentences with only certain classes of mistakes are chosen, and those types are set by the expert and submitted in the form of a certain list for becoming a testing area. Now that the system has taken a text-annotation file pair, the error annotations get filtered according to the error type. After this stage, the program gets a filtered list of text-bound annotations.

At the next step, the original text gets separated into sentences. The system looks up annotations, bound to the sentence in the filtered annotation list. For every annotation, found in the list, a test task is generated. The task includes a question, which is the sentence with all the mistakes corrected, except one, and the answer, in which that one mistake is corrected as well. If a sentence contains more than one error, test tasks are generated for each of them.

As the last step, objects on the list from the previous stage are joined to those generated at preceding iterations of the algorithm. After every text-annotation file pair in the working folder has undergone the manipulations, the whole thing gets a nice xml-wrapping, which turns it into something that is suitable for further web-exploration.

It is significant to note that the internal organisation of the system is not firmly bound with the REALEC annotation scheme. Basically, RETM is being developed as a universal tool that can be used with minimal changes for generating test questions on the basis of any corpus that uses brat for annotation.

3.3 Expert's choice

When the program finishes its work, it leaves the expert with the enormous pool of automatically generated test questions. Since no system works perfectly, this pool of tasks may contain something that does not represent a valid test question. At this point, the expert has to manually exclude the worse questions from the pool, using a web-interface developed specially for this purpose, whose working title is RECI (RETM Expert Choice Interface).

Another task the expert has to perform is assigning the level of difficulty to all the questions chosen for the test. Since the resulting test is supposed to be flexible in terms of difficulty, the questions have to



be divided into groups according to how complicated the answers to them are. The expert also performs this task with the help of RECI.

4. Conclusion

The system this short paper is devoted to, RETM, offers a great alternative to the traditional (i. e. fully manual) ways of composing placement and progress tests for EFL students. Tasks of all types, and in particular of the type «find the mistake and correct it», when the question is based on some other students' errors, by definition make students target aware of their weakest points. The fact that all the questions are manually assigned the level of difficulty, on the other hand, makes the test more flexibly adjusted to a testee's level of English. As the system is not bound to a particular annotation scheme, it can be applied to any corpus annotated with the help of brat.

References

- [1] Bachman, L. F. "What does language testing have to offer?" *Tesol Quarterly*, 25(4), 1991, 671-704
- [2] Granger, S. "Learner Corpora in Foreign Language Education", *Encyclopedia of Language and Education*. Volume 4. Second and Foreign Language Education, Springer, 2007, 337-351
- [3] Hovy, E. "Corpus Annotation", *The Oxford Handbook of Computational Linguistics* 2nd edition, Oxford University Press, 2015
- [4] <http://realec.org/about/>
- [5] <http://brat.nlplab.org/standoff.html>