WebExperimenter for multiple-choice question generation

Ayako Hoshino
Interfaculty Initiative in Information Studies
University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo,
113-0033, JAPAN
{hoshino, nakagawa}@dl. itc. u-tokyo. ac. jp

Hiroshi Nakagawa
Information Technology Center
University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo,
113-0033, JAPAN

1 Aim
Automatic generation of multiple-choice questions is an emerging topic in application of natural language processing. Particularly, applying it to language testing has been proved to be useful (Sumita et al., 2005).

This demo presents an novel approach of question generation using machine learning we have introduced in (Hoshino and Nakagawa, 2005). Our study aims to generate TOEIC-like multiple choice, fill-in-the-blank questions from given text using a classifier trained on a set of human-made questions. The system comprises of a question pool, which is a database of questions, an instance converter which does feature extraction, etc. for machine learning and a question generator. Each step of learning and generation is conducted through a web-browser.

2 Processes
An experiment is performed in a sequence of processes in each of which the system allows the user to change input/parameters and shows the result. The demo follows the processes described in the following.

Input Questions
The questions in the question pool are listed on the browser. The user can modify those questions or add new ones.

Convert to Instances
Each question in the question pool is automatically converted into instances each of which represents a possible blank position.

A sentence is [ ] to instances.
1.convert 2. converted 3. converts 4. conversion

Above question sentence is converted into the following instances, then, features such as POS, lemma, POS of the previous word, POS of the next word, position-in-sentence, sentence length are assigned to each instance in a totally automatic fashion.

We decide a blank position for a question by classifying an instance into true or false. Temporally,

1 Part-of-speech tags are tagged by a modified version of the Tree Tagger by the University of Stuttgart.
the original blank positions are labeled true, and the shifted ones are labeled as false.

<table>
<thead>
<tr>
<th>false</th>
<th>A [] sentence is converted to multiple instances.</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>A sentence [ ] converted to multiple instances.</td>
</tr>
<tr>
<td>true</td>
<td>A sentence is [ ] to multiple instances.</td>
</tr>
<tr>
<td>false</td>
<td>A sentence is converted [ ] multiple instances.</td>
</tr>
<tr>
<td>false</td>
<td>A sentence is converted to [ ] instances.</td>
</tr>
<tr>
<td>false</td>
<td>A sentence is converted to multiple [ ]</td>
</tr>
<tr>
<td>false</td>
<td>A sentence is converted to multiple instances [ ]</td>
</tr>
</tbody>
</table>

**First Training**
The instances are fed to a classifier selected among ones of Naive Bayes, K-Nearest Neighbors, Logistic Regression.

**Test on Train**
A semi-supervised learning is conducted here for the purpose of discovering falsely labeled true instances (which correspond with blank positions shifted from the original ones, but has the same properties with true instances) and the labels of those instances are changed. The classifier is re-trained on the data with new labels. This process can be iterated several times.

**Test on Training data**

![Figure 2: A screenshot of a result of test on train](image)

The instances classified as true are shown along with its temporal label and its certainty value (certainty for an instance to belong to a class true) given by the classifier.

**Supply Test Data**
The user supplies a source text for question generation from a text area. The test data is converted into instances in the same way as the training data.

**Classify Test**
The test instances are classified by the classifier which has been trained through semi-supervised learning. True instances which represents blank position are shown. Instances with a label true are passed to the next step of deciding distractors, where instances with false are discarded.

**Generate Questions**
A set of wrong answers (called distractors) are decided. The user can choose a method of deciding distractors among WordNet, Edit Distance, Mutual Information and Random. The resulting four-choice questions are shown.

**Question Session**
An interface to collect the students’ answers to generated questions is scheduled. The students’ performance is used to evaluate the questions.

3 Related Studies
The application of NLP techniques to generation of multiple-choice questions does not have a long history. Few attempts had been made before (Mitkov and Ha, 2003), in which a semi-automatic question generation on student’s knowledge of linguistic terms are evaluated. Sumita et al. used automatically generated questions to measure test taker’s proficiency in English (2005). We are proposing a machine learning approach which depends on a training on a collection of manually made questions (Hoshino and Nakagawa, 2005).

**References**
