A real-time multiple-choice question generation for language testing – a preliminary study–

Ayako Hoshino
Interfaculty Initiative in Information Studies
University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo,
113-0033, JAPAN
qq36126@iii.u-tokyo.ac.jp

Hiroshi Nakagawa
Information Technology Center
University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo,
113-0033, JAPAN
nakagawa@dl.itc.u-tokyo.ac.jp

Abstract

An automatic generation of multiple-choice questions is one of the promising examples of educational applications of NLP techniques. A machine learning approach seems to be useful for this purpose because some of the processes can be done by classification. Using basic machine learning algorithms as Naive Bayes and K-Nearest Neighbors, we have developed a real-time system which generates questions on English grammar and vocabulary from on-line news articles. This paper describes the current version of our system and discusses some of the issues on constructing this kind of system.

1 Introduction

Multiple-choice question exams are widely used and are effective to assess students' knowledge, however it is costly to manually produce those questions. Naturally, this kind of task should be done with a help of computer.

Nevertheless, there have been very few attempts to generate multiple-choice questions automatically. Mitkov et al.(2003) generated questions for a linguistics exam in a semi-automatic way and evaluated that it exceeds manually made ones in cost and is at least equivalent in quality. There are some other researches that involve generating questions with multiple alternatives (Dicheva and Dimitrova, 1998). But to the best of our knowledge, no attempt has been made to generate this kind of questions in a totally automatic way.

This paper presents a novel approach to generate multiple-choice questions using machine learning techniques. The questions generated are those of fill-in-the-blank type, so it does not involve transforming declarative sentences into question sentences as in Mitkov's work. This simplicity makes the method to be language independent.

Although this application can be very versatile, in that it can be used to test any kind of knowledge as in history exams, as a purpose of this research we limit ourselves to testing student's proficiency in a foreign language. One of the purposes of this research is to automatically extract important words or phrases in a text for a learner of the language.

2 System Design

The system we have implemented works in a simple pipelined manner; it takes an HTML file and turns it into the one of quiz session. The process of converting the input to multiple-choice questions includes extracting features, deciding the blank positions, and choosing the wrong alternatives (which are called *distractors*), which are all done in a moment when the user feeds the input. When the user submits their answer, it shows the text with the correct answers as well as an overall feed back.

3 Methodology

The process of deciding blank positions in a given text follows a standard machine learning framework, which is first training a classifier on a training data

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 17–20, Ann Arbor, June 2005.

Table 1: the full list of test instances classified as *true* in test-on-train

certainty	a test instance (sentence with a blank)	the answer
0.808	Joseph is preparing for tomorrow's big [ ] to the president.	presentation
0.751	Ms. Singh listened [ ] to the president's announcement.	carefully
0.744	The PR person is the one in charge of [ ] meetings and finding accommodations for our associates.	scheduling
0.73	Ms. Havlat received a memo from the CEO [ ] the employees' conduct.	regarding
0.718	The amount of money in the budget decreased [ ] over the past year.	significantly
0.692	Mr. Gomez is [ ] quickly; however it will be a log time before he gets used to the job.	learning
0.689	The boss can never get around to [ ] off his desk.	cleaning
0.629	The interest rate has been increasingly [ ] higher.	getting
0.628	Employees are [ ] to comply with the rules in the handbook.	asked
0.62	The lawyer [ ] his case before the court.	presented
0.59	The secretary was [ ] to correspond with the client immediately.	supposed
0.576	The maintenance worker checked the machine before [ ] it on.	turning
0.523	The [ ] manager's office is across the corridor.	assistant

(i.e. TOEIC questions), then applying it on an unseen test data, (i.e. the input text). In the current system, the mechanism of choosing distractors is implemented with the simplest algorithm, and its investigation is left to future work.

3.1 Preparing the Training Data

The training data is a collection of fill-in-the-blank questions from a TOEIC preparation book (Matsuno et al., 2000). As shown in the box below, a question consists of a sentence with a missing word (or words) and four alternatives one of among which best fits into the blank.

> Many people showed up early to [ ] for the position that was open.
> 1. apply 2. appliance 3. applies 4. application

The training instances are obtained from 100 questions by shifting the blank position. The original position is labeled as *true*, while sentences with a blank in a shifted position are at first labeled as *false*. The instance shown above therefore yields instances *[ ] people showed up early to apply for the position that was open.*, *Many [ ] showed up early to apply for the position that was open.*, and so on, all of which are labeled as *false* except the original blank position. 1962 (100 *true* and 1862 *false*) instances were obtained.

The label *true* here is supposed to indicate that it is possible to make a question with the sentence with a blank in the specified position, while many of the shifted positions which are labeled *false* can also be good blanks. A semi-supervised learning (Chakrabarti, 2003) [1] is conducted in the following manner to retrieve the instances that are potentially *true* among the ones initially classified as *false*.

We retrieved the 13 instances (shown in Table 1.) which had initially been labeled as *false* and classified as *true* in a test-on-train result with a certainty [2] of more than 0.5 with a Naive Bayes classifier [3]. The labels of those instances were changed to *true* before re-training the classifier. In this way, a training set with 113 *true* instances was obtained.

3.2 Deciding Blank Positions

For the current system we use news articles from BBC.com [4], which consist approximately 200-500 words. The test text goes through tagging and feature extraction in the same manner as the training

[1] Semi-supervised learning is a method to identify the class of unclassified instances in the dataset where only some of the instances are classified.

[2] The result of a classification of a instance is obtained along with a *certainty* value between 0.0 to 1.0 for each class, which indicates how certain it is that an instance belongs to the class.

[3] Seven features which are word, POS, POS of the previous word, POS of the next word, position in the sentence, sentence length, word length and were used.

[4] `http://news.bbc.co.uk/`

data, and the instances are classified into *true* or *false*. The positions of the blanks are decided according to the certainty of the classification so the blanks (i.e. questions) are generated as many as the user has specified.

3.3 Choosing Distractors

In the current version of the system, the distractors are chosen randomly from the same article excluding punctuations and the same word as the other alternatives.

4 Current system

The real-time system we are presenting is implemented as a Java servlet, whose one of the main screens is shown below. The tagger used here is the Tree tagger (Schmid, 1994), which uses the Penn-Treebank tagset.

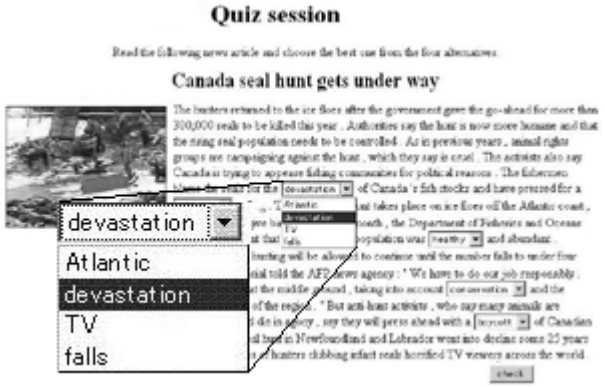

Figure 1: a screen shot of the question session page with an enlarged answer selector.

The current version of the system is available at `http://www.iii.u-tokyo.ac.jp/~qq36126/mcwa1/`. The interface of the system consists of three sequenced web pages, namely 1)the parameter selection page, 2)the quiz session page and 3)the result page.

The parameter selection page shows the list of the articles which are linked from the top page of the BBC website, along with the option selectors for number of blanks (5-30) and the classifier (Naive Bayes or Nearest Neighbors).

The question session page is shown in Figure 1. It displays the headline and the image from the chosen article under the title and a brief instruction. The alternatives are shown on option selectors, which are placed in the article text.

The result page shows the text with the right answers shown in green when the user's choice is correct, red when it is wrong.

5 Evaluation

To examine the quality of the questions generated by the current system, we have evaluated the blank positions determined by a Naive Bayes classifier and a KNN classifier (K=3) with a certainty of more than 50 percent in 10 articles.

Among 3138 words in total, 361 blanks were made and they were manually evaluated according to their possibility of being a multiple-choice question, with an assumption of having alternatives of the same part of speech. The blank positions were categorized into three groups, which are **E** (possible to make a question), and **D** (difficult, but possible to make a question), **NG** (not possible or not suitable e.g. on a punctuation). The guideline for deciding **E** or **D** was if a question is on a grammar rule, or it requires more semantic understanding, for instance, a background knowledge [5].

Table 2. shows the comparison of the number of blank positions decided by the two classifiers, each with a breakdown for each evaluation. The number in braces shows the proportion of the blanks with a certain evaluation over the total number of blanks made by the classifier. The rightmost column **I** shows the number of the same blank positions selected by both classifiers.

The KNN classifier tends to be more accurate and seems to be more robust, although given the fact that it produces less blanks. The fact that an instance-based algorithm exceeds Naive Bayes, whose decision depends on the whole data, can be ascribed to a mixed nature of the training data. For example, blanks for grammar questions might have different features from ones for vocabulary questions.

The result we sampled has exhibited another problem of Naive Bayes algorithm. In two articles among the data, it has shown the tendency to make a blank on *be-verb*s. Naive Bayes tends to choose the

[5] A blank on a verbs or a part of idioms (as [according] to) was evaluated as **E**, most of the blanks on an adverbs, and (as [now]) were **D** and a blank on a punctuation or a quotation mark was **NG**.

Table 2: The evaluation on the blank positions decided by a Naive Bayes (**NB**) and a KNN classifier.

	NB				KNN				I
	blanks	E(%)	D(%)	NG(%)	blanks	E(%)	D(%)	NG(%)	blanks
Article1	69	44(63.8)	21(30.4)	4(5.8)	33	20(60.6)	11(33.3)	2(6.1)	18
Article2	22	5(22.7)	3(13.6)	14(63.6)	8	5(62.5)	3(37.5)	0(0.0)	0
Article3	38	21(55.3)	15(39.5)	2(5.3)	18	12(66.7)	5(27.8)	1(5.6)	8
Article4	19	10(52.6)	9(47.4)	0(0.0)	9	7(77.8)	2(22.2)	0(0.0)	3
Article5	28	18(64.3)	10(35.7)	0(0.0)	14	10(71.4)	4(28.6)	0(0.0)	6
Article6	26	17(65.4)	8(30.8)	1(3.8)	11	6(54.5)	5(45.5)	0(0.0)	4
Article7	18	9(50.0)	5(27.8)	4(22.2)	6	3(50.0)	3(50.0)	0(0.0)	3
Article8	24	14(58.3)	9(37.5)	1(4.2)	5	3(60.0)	2(40.0)	0(0.0)	5
Article9	20	16(80.0)	4(20.0)	0(0.0)	6	2(33.3)	4(66.7)	0(0.0)	4
Article10	30	18(60.0)	12(40.0)	0(0.0)	14	11(78.6)	3(21.4)	0(0.0)	6
	294	172(58.5)	96(32.7)	26(8.8)	124	79(63.7)	42(33.9)	3(2.4)	57

same word as a blank position, therefore generates many questions on the same word in one article.

Another general problem of these methods would be that the blank positions are decided without consideration of one another; the question will be sometimes too difficult when another blank is next to or in the vicinity of the blank.

6 Discussion and Future work

From the problems of the current system, we can conclude that the feature set we have used is not sufficient. It is necessary that we use larger number of features, possibly including semantic ones, so a blank position would not depend on its superficial aspects. Also, the training data should be examined in more detail.

As it was thought to be a criteria of evaluating generated questions, if a question requires simply a grammatical knowledge or a farther knowledge (i.e. background knowledge) can be a critical property of a generated question. We should differentiate the features from the ones which are used to generate, for example, history questions, which require rather background knowledge. Selecting suitable distractors, which is left to future work, would be a more important process in generating a question. A semantic *distance* between an alternative and the right answer are suggested (Mitkov and Ha, 2003), to be a good measure to evaluate an alternative. We are investigating on a method of measuring those distances and a mechanism to retrieve best alternatives automatically.

7 Conclusion

We have presented a novel application of automatically generating fill-in-the-blank, multiple-choice questions using machine learning techniques, as well as a real-time system implemented. Although it is required to explore more feature settings for the process of determining blank positions, and the process of choosing distractors needs more elaboration, the system has proved to be feasible.

References

Soumen Chakrabarti. 2003. *Mining the Web*. Morgan Kaufmann Publishers.

Darina Dicheva and Vania Dimitrova. 1998. An approach to representation and extraction of terminological knowledge in icall. In *Journal of Computing and Information Technology*, pages 39 – 52.

Shuhou Matsuno, Tomoko Miyahara, and Yoshi Aoki. 2000. *STEP-UP Bunpo mondai TOEIC TEST*. Kirihara Publisher.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17 – 22, Edmonton, Canada, May.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK, September.