Applications of Lexical Information for Algorithmically Composing Multiple-Choice Cloze Items

Chao-Lin Liu† **Chun-Hung Wang**† **Zhao-Ming Gao**‡ **Shang-Ming Huang**†
†Department of Computer Science, National Chengchi University, Taipei 11605, Taiwan
‡Dept. of Foreign Lang. and Lit., National Taiwan University, Taipei 10617, Taiwan
†chaolin@nccu.edu.tw, ‡zmgao@ntu.edu.tw

ABSTRACT[1]
We report experience in applying techniques for natural language processing to algorithmically generating test items for both reading and listening cloze items. We propose a word sense disambiguation-based method for locating sentences in which designated words carry specific senses, and apply a collocation-based method for selecting distractors that are necessary for multiple-choice cloze items. Experimental results indicate that our system was able to produce a usable item for every 1.6 items it returned. We also attempt to measure distance between sounds of words by considering phonetic features of the words. With the help of voice synthesizers, we were able to assist the task of composing listening cloze items. By providing both reading and listening cloze items, we would like to offer a somewhat adaptive system for assisting Taiwanese children in learning English vocabulary.

1 Introduction

Computer-assisted item generation (CAIG) allows the creation of large-scale item banks, and has attracted active study in the past decade (Deane and Sheehan, 2003; Irvine and Kyllonen, 2002). Applying techniques for natural language processing (NLP), CAIG offers the possibility of creating a large number of items of different challenging levels, thereby paving a way to make computers more adaptive to students of different competence. Moreover, with the proliferation of Web contents, one may search and sift online text files for candidate sentences, and come up with a list of candidate cloze items economically. This unleashes the topics of the test items from being confined by item creators' personal interests.

NLP techniques serve to generate multiple-choice cloze items in different ways. (For brevity, we use *cloze items* or *items* for *multiple-choice cloze items* henceforth.) One may create sentences from scratch by applying template-based methods (Dennis et al., 2002) or more complex methods based on some predetermined principles (Deane and Sheehan, 2003). Others may take existing sentences from a corpus, and select those that meet the criteria for becoming test items. The former approach provides specific and potentially well-controlled test items at the costs of more complex systems than the latter, e.g., (Sheehan et al., 2003). Nevertheless, as the Web provides ample text files at our disposal, we may filter the text sources stringently for obtaining candidate test items of higher quality. Administrators can then select really usable items from these candidates at a relatively lower cost.

Some researchers have already applied NLP techniques to the generation of sentences for multiple-choice cloze items. Stevens (1991) employs the concepts of concordance and collocation for generating items with general corpora. Coniam (1997) relies on factors such as word frequencies in a tagged corpus for creating test items of particular types.

There are other advanced NLP techniques that may help to create test items of higher quality. For instance, many words in English may carry multiple senses, and test administrators usually want to test a particular usage of the word in an item. In this case, blindly applying a keyword matching method, such as a concordancer, may lead us to a list of irrelevant sentences that would demand a lot of postprocess-

[1] A portion of results reported in this paper will be expanded in (Liu et al., 2005; Huang et al., 2005).

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 1–8, Ann Arbor, June 2005.

1. My sister is _____, that is, I am going to be an uncle soon.
   (A) supposing (B) assigning
   (C) expecting (D) scheduling

Figure 1: A multiple-choice cloze item for English

ing workload. In addition, composing a cloze item requires not just a useful sentence.

Figure 1 shows a multiple-choice item, where we call the sentence with a gap the **stem**, the answer to the gap the **key**, and the other choices the **distractors**. Given a sentence, we still need distractors for a multiple-choice item. The selection of distractors affects the *item facility* and *item discrimination* of the cloze items (Poel and Weatherly, 1997). Therefore, the selection of distractors calls for deliberate strategies, and simple considerations alone, such as word frequencies, may not satisfy the demands.

To remedy these shortcomings, we employ the techniques for word sense disambiguation (WSD) for choosing sentences in which the keys carries specific senses, and utilize the techniques for computing collocations (Manning and Schütze, 1999) for selecting distractors. Results of empirical evaluation show that our methods could create items of satisfactory quality, and we have actually used the generated cloze items in freshmen-level English classes.

For broadening the formats of cloze items, we also design software that assists teachers to create listening cloze items. After we defining a metric for measuring similarity between pronunciations of words, our system could choose distractors for listening cloze items. This addition opens a door to offering different challenging levels of cloze items.

We sketch the flow of the item generation process in Section 2, and explain the preparation of the source corpus in Section 3. In Section 4, we elaborate on the application of WSD to selecting sentences for cloze items, and, in Section 5, we delve into the application of collocations to distractor generation. Results of evaluating the created reading cloze items are presented in Section 6. We then outline methods for creating listening cloze items in Section 7 before making some concluding remarks.

2 System Architecture

Figure 2 shows major steps for creating cloze items. Constrained by test administrator's specifications and domain dependent requirements, the *Sentence Retriever* chooses a candidate sentence from the

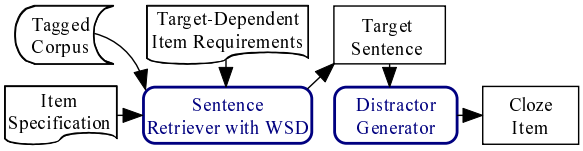

Figure 2: Main components of our item generator

Tagged Corpus. *Target-Dependent Item Requirements* specify general principles that should be followed by all items for a particular test. For example, the number of words in cloze items for College Entrance Examinations in Taiwan (CEET) ranges between 6 and 28 (Liu et al., 2005), and one may want to follow this tradition in creating drill tests.

Figure 3 shows the interface to the *Item Specification*. Through this interface, test administrators select the key for the desired cloze item, and specify part-of-speech and sense of the key that will be used in the item. Our system will attempt to create the requested number of items. After retrieving the target sentence, the *Distractor Generator* considers such constraining factors as word frequencies and collocations in selecting the distractors at the second step.

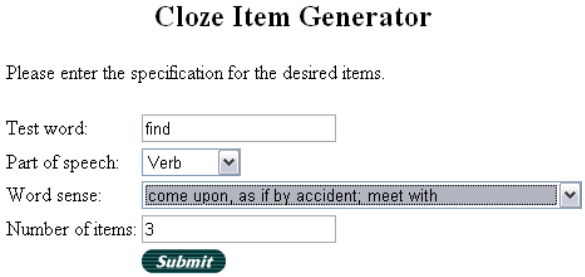

Figure 3: Interface for specifying cloze items

Figure 4 shows a sample output for the specification shown in Figure 3. Given the generated items, the administrator may choose and edit the items, and save the edited items into the item bank. It is possible to retrieve previously saved items from the item bank, and compile the items for different tests.

3 Source Corpus and Lexicons

Employing a web crawler, we retrieve the contents of *Taiwan Review* <publish.gio.gov.tw>, *Taiwan Journal* <taiwanjournal.nat.gov.tw>, and *China Post* <www.chinapost.com.tw>. Currently, we have 127,471 sentences that consist of 2,771,503 words in 36,005 types in the corpus. We look for useful sentences from web pages that are encoded in the HTML format. We need to extract texts from

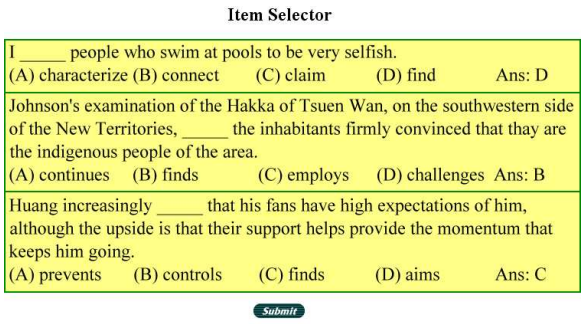

Figure 4: An output after Figure 3

the mixture of titles, main body of the reports, and multimedia contents, and then segment the extracted paragraphs into individual sentences. We segment sentences with the help of MXTERMINATOR (Reynar and Ratnaparkhi, 1997). We then tokenize words in the sentences before assigning useful tags to the tokens.

We augment the text with an array of tags that facilitate cloze item generation. We assign tags of part-of-speech (POS) to the words with MXPOST that adopts the Penn Treebank tag set (Ratnaparkhi, 1996). Based on the assigned POS tags, we annotate words with their lemmas. For instance, we annotate *classified* with *classify* and *classified*, respectively, when the original word has *VBN* and *JJ* as its POS tag. We also employ MINIPAR (Lin, 1998) to obtain the partial parses of sentences that we use extensively in our system. Words with direct relationships can be identified easily in the partially parsed trees, and we rely heavily on these relationships between words for WSD. For easy reference, we will call words that have direct syntactic relationship with a word W as W's **signal words** or simply **signals**.

Since we focus on creating items for verbs, nouns, adjectives, and adverbs (Liu et al., 2005), we care about signals of words with these POS tags in sentences for disambiguating word senses. Specifically, the signals of a verb include its subject, object, and the adverbs that modify the verb. The signals of a noun include the adjectives that modify the noun and the verb that uses the noun as its object or predicate. For instance, in "Jimmy builds a grand building.", both "build" and "grand" are signals of "building". The signals of adjectives and adverbs include the words that they modify and the words that modify the adjectives and adverbs.

When we need lexical information about English words, we resort to electronic lexicons. We use WordNet <www.cogsci.princeton.edu/~wn/> when we need definitions and sample sentences of words for disambiguating word senses, and we employ HowNet <www.keenage.com> when we need information about classes of verbs, nouns, adjectives, and adverbs.

HowNet is a bilingual lexicon. An entry in HowNet includes slots for Chinese words, English words, POS information, etc. We rely heavily on the slot that records the semantic ingredients related to the word being defined. HowNet uses a limited set of words in the slot for semantic ingredient, and the leading ingredient in the slot is considered to be the most important one generally.

4 Target Sentence Retriever

The sentence retriever in Figure 2 extracts qualified sentences from the corpus. A sentence must contain the desired key of the requested POS to be considered as a candidate target sentence. Having identified such a candidate sentence, the item generator needs to determine whether the sense of the key also meets the requirement. We conduct this WSD task based on an extended notion of selectional preferences.

4.1 Extended Selectional Preferences

Selectional preferences generally refer to the phenomenon that, under normal circumstances, some verbs constrain the meanings of other words in a sentence (Manning and Schütze, 1999; Resnik, 1997). We can extend this notion to the relationships between a word of interest and its signals, with the help of HowNet. Let w be the word of interest, and π be the first listed class, in HowNet, of a signal word that has the syntactic relationship μ with w. We define the strength of the association of w and π as follows:

$$A_\mu(w,\pi) = \frac{\Pr_\mu(w,\pi)}{\Pr_\mu(w)}, \quad (1)$$

where $\Pr_\mu(w)$ is the probability of w participating in the μ relationship, and $\Pr_\mu(w,\pi)$ is the probability that both w and π participate in the μ relationship.

4.2 Word Sense Disambiguation

We employ the generalized selectional preferences to determine the sense of a polysemous word in a sentence. Consider the task of determining the sense

of "spend" in the candidate target sentence "They say film makers don't spend enough time developing a good story." The word "spend" has two possible meanings in WordNet.

1. (99) spend, pass – (pass (time) in a specific way; "How are you spending your summer vacation?")
2. (36) spend, expend, drop – (pay out; "I spend all my money in two days.")

Each definition of the possible senses include (1) the **head words** that summarize the intended meaning and (2) a sample sentence for sense. When we work on the disambiguation of a word, we do not consider the word itself as a head word in the following discussion. Hence, "spend" has one head word, i.e., "pass", in the first sense and two head words, i.e., "extend" and "drop", in the second sense.

An intuitive method for determining the meaning of "spend" in the target sentence is to replace "spend" with its head words in the target sentence. The head words of the correct sense should go with the target sentence better than head words of other senses. This intuition leads to the a part of the scores for senses, i.e., $\mathfrak{S}_t$ that we present shortly.

In addition, we can compare the similarity of the contexts of "spend" in the target sentence and sample sentences, where *context* refers to the classes of the signals of the word being disambiguated. For the current example, we can check whether the subject and object of "spend" in the target sentence have the same classes as the subjects and objects of "spend" in the sample sentences. The sense whose sample sentence offers a more similar context for "spend" in the target sentence receives a higher score. This intuition leads to the other part of the scores for senses, i.e., $\mathfrak{S}_s$ that we present below.

Assume that the key w has n senses. Let $\Theta = \{\theta_1, \theta_2, \cdots, \theta_n\}$ be the set of senses of w. Assume that sense θ_j of word w has m_j head words in WordNet. (Note that we do not consider w as its own head word.) We use the set $\Lambda_j = \{\lambda_{j,1}, \lambda_{j,2}, \cdots, \lambda_{j,m_j}\}$ to denote the set of head words that WordNet provides for sense θ_j of word w.

When we use the partial parser to parse the target sentence T for a key, we obtain information about the signal words of the key. Moreover, for each of these signals, we look up their classes in HowNet, and adopt the first listed class for each of the signals when the signal covers multiple classes. Assume that there are $\mu(T)$ signals for the key w in a sentence T. We use the set $\Psi(T, w) = \{\psi_{1,T}, \psi_{2,T}, \cdots, \psi_{\mu(T),T}\}$ to denote the set of signals for w in T. Correspondingly, we use $\upsilon_{j,T}$ to denote the syntactic relationship between w and $\psi_{j,T}$ in T, use $\Upsilon(T, w) = \{\upsilon_{1,T}, \upsilon_{2,T}, \cdots, \upsilon_{\mu(T),T}\}$ for the set of relationships between signals in $\Psi(T, w)$ and w, use $\pi_{j,T}$ for the class of $\psi_{j,T}$, and use $\Pi(T, w) = \{\pi_{1,T}, \pi_{2,T}, \cdots, \pi_{\mu(T),T}\}$ for the set of classes of the signals in $\Psi(T, w)$.

Equation (2) measures the average strength of association of the head words of a sense with signals of the key in T, so we use (2) as a part of the score for w to take the sense θ_j in the target sentence T. Note that both the strength of association and $\mathfrak{S}_t$ fall in the range of [0,1].

$$\mathfrak{S}_t(\theta_j|w, T) = \frac{1}{m_j}\sum_{k=1}^{m_j}\frac{1}{\mu(T)}\sum_{l=1}^{\mu(T)} A_{\mu_{l,T}}(\lambda_{j,k}, \pi_{l,T}) \quad (2)$$

In (2), we have assumed that the signal words are not polysemous. If they are polysemous, we assume that each of the candidate sense of the signal words are equally possible, and employ a slightly more complicated formula for (2). This assumption may introduce errors into our decisions, but relieves us from the needs to disambiguate the signal words in the first place (Liu et al., 2005).

Since WordNet provides sample sentences for important words, we also use the degrees of similarity between the sample sentences and the target sentence to disambiguate the word senses of the key word in the target sentence. Let T and S be the target sentence of w and a sample sentence of sense θ_j of w, respectively. We compute this part of score, $\mathfrak{S}_s$, for θ_j using the following three-step procedure. If there are multiple sample sentences for a given sense, say θ_j of w, we will compute the score in (3) for each sample sentence of θ_j, and use the average score as the final score for θ_j.

Procedure for computing $\mathfrak{S}_s(\theta_j|w, T)$

1. Compute signals of the key and their relationships with the key in the target and sample sentences.

$$\begin{aligned}
\Psi(T,w) &= \{\psi_{1,T},\psi_{2,T},\cdots,\psi_{\mu(T),T}\},\\
\Upsilon(T,w) &= \{\upsilon_{1,T},\upsilon_{2,T},\cdots,\upsilon_{\mu(T),T}\},\\
\Psi(S,w) &= \{\psi_{1,S},\psi_{2,S},\cdots,\psi_{\mu(S),S}\}, \text{and}\\
\Upsilon(S,w) &= \{\upsilon_{1,S},\upsilon_{2,S},\cdots,\upsilon_{\mu(S),S}\}
\end{aligned}$$

2. We look for $\psi_{j,T}$ and $\psi_{k,S}$ such that $\upsilon_{j,T} = \upsilon_{k,S}$, and then check whether $\pi_{j,T} = \pi_{k,S}$. Namely, for each signal of the key in T, we check the signals of the key in S for matching syntactic relationships and word classes, and record the counts of matched relationship in $M(\theta_j, T)$ (Liu et al., 2005).
3. The following score measures the proportion of matched relationships among all relationships between the key and its signals in the target sentence.

$$\mathfrak{S}_s(\theta_j|w,T) = \frac{M(\theta_j,T)}{\mu(T)} \tag{3}$$

The score for w to take sense θ_j in a target sentence T is the sum of $\mathfrak{S}_t(\theta_j|w,T)$ defined in (2) and $\mathfrak{S}_s(\theta_j|w,T)$ defined in (3), so the sense of w in T will be set to the sense defined in (4) when the score exceeds a selected threshold. When the sum of $\mathfrak{S}_t(\theta_j|w,T)$ and $\mathfrak{S}_s(\theta_j|w,T)$ is smaller than the threshold, we avoid making arbitrary decisions about the word senses. We discuss and illustrate effects of choosing different thresholds in Section 6.

$$\arg\max_{\theta_j\in\Theta} \quad \mathfrak{S}_t(\theta_j|w,T) + \mathfrak{S}_s(\theta_j|w,T) \tag{4}$$

5 Distractor Generation

Distractors in multiple-choice items influence the possibility of making lucky guesses to the answers. Should we use extremely impossible distractors in the items, examinees may be able to identify the correct answers without really knowing the keys. Hence, we need to choose distractors that appear to fit the gap, and must avoid having multiple answers to items in a typical cloze test at the same time.

There are some conceivable principles and alternatives that are easy to implement and follow. Antonyms of the key are choices that average examinees will identify and ignore. The part-of-speech tags of the distractors should be the same as the key in the target sentence. We may also take cultural background into consideration. Students in Taiwan tend to associate English vocabularies with their Chinese translations. Although this learning strategy works most of the time, students may find it difficult to differentiate English words that have very similar Chinese translations. Hence, a culture-dependent strategy is to use English words that have similar Chinese translations with the key as the distractors.

To generate distractors systematically, we employ ranks of word frequencies for selecting distractors (Poel and Weatherly, 1997). Assume that we are generating an item for a key whose part-of-speech is ρ, that there are n word types whose part-of-speech may be ρ in the dictionary, and that the rank of frequency of the key among these n types is m. We randomly select words that rank in the range $[m-n/10, m+n/10]$ among these n types as candidate distractors. These distractors are then screened by their fitness into the target sentence, where *fitness* is defined based on the concept of collocations of word classes, defined in HowNet, of the distractors and other words in the stem of the target sentence.

Recall that we have marked words in the corpus with their signals in Section 3. The words that have more signals in a sentence usually contribute more to the meaning of the sentence, so should play a more important role in the selection of distractors. Since we do not really look into the semantics of the target sentences, a relatively safer method for selecting distractors is to choose those words that seldom collocate with important words in the target sentence.

Let $T = \{t_1, t_2, \cdots, t_n\}$ denote the set of words in the target sentence. We select a set $T' \subset T$ such that each $t'_i \in T'$ has two or more signals in T and is a verb, noun, adjective, or adverb. Let κ be the first listed class, in HowNet, of the candidate distractor, and $\aleph = \{\tau_i | \tau_i$ is the first listed class of a $t'_i \in T'\}$. The fitness of a candidate distractor is defined in (5).

$$\frac{-1}{|\aleph|}\sum_{\tau_i\in\aleph}\log\frac{\Pr(\kappa,\tau_i)}{\Pr(\kappa)\Pr(\tau_i)} \tag{5}$$

The candidate whose score is better than 0.3 will be admitted as a distractor. $\Pr(\kappa)$ and $\Pr(\tau_i)$ are the probabilities that each word class appears individually in the corpus, and $\Pr(\kappa,\tau_i)$ is the probability that the two classes appear in the same sentence. Operational definitions of these probabilities

Table 1: Accuracy of WSD

POS	baseline	threshold=0.4	threshold=0.7
verb	38.0%(19/50)	57.1%(16/28)	68.4%(13/19)
noun	34.0%(17/50)	63.3%(19/30)	71.4%(15/21)
adj.	26.7%(8/30)	55.6%(10/18)	60.0%(6/10)
adv.	36.7%(11/30)	52.4%(11/21)	58.3%(7/12)

are provided in (Liu et al., 2005). The term in the summation is a pointwise mutual information, and measures how often the classes κ and τ_i collocate in the corpus. We negate the averaged sum so that classes that seldom collocate receive higher scores. We set the threshold to 0.3, based on statistics of (5) that are observed from the cloze items used in the 1992-2003 CEET.

6 Evaluations and Applications

6.1 Word Sense Disambiguation

Different approaches to WSD were evaluated in different setups, and a very wide range of accuracies in [40%, 90%] were reported (Resnik, 1997; Wilks and Stevenson, 1997). Objective comparisons need to be carried out on a common test environment like SENSEVAL, so we choose to present only our results.

We arbitrarily chose, respectively, 50, 50, 30, and 30 sentences that contained polysemous verbs, nouns, adjectives, and adverbs for disambiguation. Table 1 shows the percentage of correctly disambiguated words in these 160 samples.

The *baseline* column shows the resulting accuracy when we directly use the most frequent sense, recorded in WordNet, for the polysemous words. The rightmost two columns show the resulting accuracy when we used different thresholds for applying (4). As we noted in Section 4.2, our system selected fewer sentences when we increased the threshold, so the selected threshold affected the performance. A larger threshold led to higher accuracy, but increased the rejection rate at the same time. Since the corpus can be extended to include more and more sentences, we afford to care about the accuracy more than the rejection rate of the sentence retriever.

We note that not every sense of all words have sample sentences in the WordNet. When a sense does not have any sample sentence, this sense will receive no credit, i.e., 0, for $\mathfrak{S}_s$. Consequently, our current reliance on sample sentences in WordNet makes us discriminate against senses that do not have sample sentences. This is an obvious drawback in our current design, but the problem is not really detrimental and unsolvable. There are usually sample sentences for important and commonly-used senses of polysemous words, so the discrimination problem does not happen frequently. When we do want to avoid this problem once and for all, we can customize WordNet by adding sample sentences to all senses of important words.

Table 2: Correctness of the generated sentences

POS of the key	# of items	% of correct sentences
verb	77	66.2%
noun	62	69.4%
adjective	35	60.0%
adverb	26	61.5%
	overall	65.5%

Table 3: Uniqueness of answers

item category	key's POS	number of items	results
cloze	verb	64	90.6%
	noun	57	94.7%
	adjective	46	93.5%
	adverb	33	84.8%
		overall	91.5%

6.2 Cloze Item Generation

We asked the item generator to create 200 items in the evaluation. To mimic the distribution over keys of the cloze items that were used in CEET, we used 77, 62, 35, and 26 items for verbs, nouns, adjectives, and adverbs, respectively, in the evaluation.

In the evaluation, we requested one item at a time, and examined whether the sense and part-of-speech of the key in the generated item really met the requests. The threshold for using (4) to disambiguate word sense was set to 0.7. Results of this experiment, shown in Table 2, do not differ significantly from those reported in Table 1. For all four major classes of cloze items, our system was able to return a correct sentence for less than every 2 items it generated. In addition, we checked the quality of the distractors, and marked those items that permitted unique answers as good items. Table 3 shows that our system was able to create items with unique answers for another 200 items most of the time.

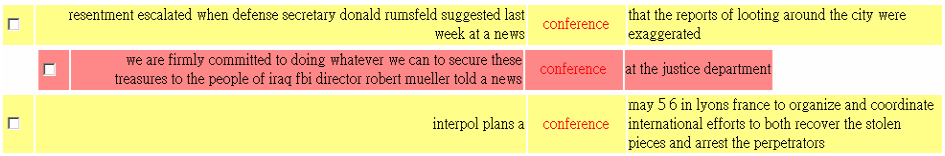

Figure 5: A phonetic concordancer

6.3 More Applications

We have used the generated items in real tests in a freshman-level English class at National Chengchi University, and have integrated the reported item generator in a Web-based system for learning English. In this system, we have two major subsystems: the authoring and the assessment subsystems. Using the authoring subsystem, test administrators may select items from the interface shown in Figure 4, save the selected items to an item bank, edit the items, including their stems if necessary, and finalize the selection of the items for a particular examination. Using the assessment subsystem, students answer the test items via the Internet, and can receive grades immediately if the administrators choose to do so. The answers of students are recorded for student modelling and analysis of the item facility and the item discrimination.

7 Generating Listening Cloze Items

We apply the same infrastructure for generating reading cloze items, shown in Figure 2, for the generation of listening cloze items (Huang et al., 2005). Due to the educational styles in Taiwan, students generally find it more difficult to comprehend messages by listening than by reading. Hence, we can regard listening cloze tests as an advanced format of reading cloze tests. Having constructed a database of sentences, we can extract sentences that contain the key for which the test administrator would like to have a listening cloze, and employ voice synthesizers to create the necessary recordings.

Figure 5 shows an interface through which administrators choose and edit sentences for listening cloze items. Notice that we employ the concept that is related to ordinary concordance in arranging the extracted sentences. By defining a metric for measuring similarity between sounds, we can put sentences that have similar phonetic contexts around the key near each other. We hope this would better help teachers in selecting sentences by this rudimentary clustering of sentences.

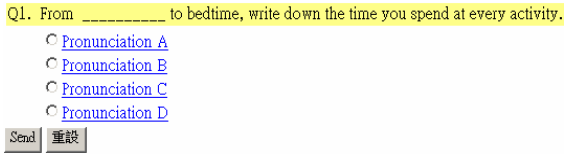

Figure 6: The most simple form of listening cloze

Figure 6 shows the most simple format of listening cloze items. In this format, students click on the options, listen to the recorded sounds, and choose the option that fit the gap. The item shown in this figure is very similar to that shown in Figure 1, except that students read and hear the options. From this most primitive format, we can image and implement other more challenging formats. For instance, we can replace the stem, currently in printed form in Figure 6, into clickable links, demanding students to hear the stem rather than reading the stem. A middle ground between this more challenging format and the original format in the figure is to allow the gap to cover more words in the original sentence. This would require the students to listen to a longer stream of sound, so can be a task more challenging than the original test. In addition to controlling the lengths of the answer voices, we can try to modulate the speed that the voices are replayed. Moreover, for multiple-word listening cloze, we may try to find word sequences that sound similar to the answer sequence to control the difficulty of the test item.

Defining a metric for measuring similarity between two recordings is the key to support the aforementioned functions. In (Huang et al., 2005), we consider such features of phonemes as place and manner of pronunciation in calculating the similarity between sounds. Using this metric we choose as distractors those sounds of words that have similar pronunciation with the key of the listening cloze. We have to define the distance between each phoneme so that we could employ the minimal-edit-distance algorithm for computing the distance between the sounds of different words.

8 Concluding Remarks

We believe that NLP techniques can play an important role in computer assisted language learning, and this belief is supported by papers in this workshop and the literature. What we have just explored is limited to the composition of cloze items for English vocabulary. With the assistance of WSD techniques, our system was able to identify sentences that were qualified as candidate cloze items 65% of the time. Considering both word frequencies and collocation, our system recommended distractors for cloze items, resulting in items that had unique answers 90% of the time. In addition to assisting the composition of cloze items in the printed format, our system is also capable of helping the composition of listening cloze items. The current system considers features of phonemes in computing distances between pronunciations of different word strings.

We imagine that NLP and other software techniques could empower us to create cloze items for a wide range of applications. We could control the formats, contents, and timing of the presented material to manipulate the challenging levels of the test items. As we have indicated in Section 7, cloze items in the listening format are harder than comparable items in the printed format. We can also control when and what the students can hear to fine tune the difficulties of the listening cloze items.

We must admit, however, that we do not have sufficient domain knowledge in how human learn languages. Consequently, tools offered by computing technologies that appear attractive to computer scientists or computational linguists might not provide effective assistance for language learning or diagnosis. Though we have begun to study item comparison from a mathematical viewpoint (Liu, 2005), the current results are far from being practical. Expertise in psycholinguistics may offer a better guidance on our system design, we suppose.

Acknowledgements

We thank anonymous reviewers for their invaluable comments on a previous version of this report. We will respond to some suggestions that we do not have space to do so in this report in the workshop. This research was supported in part by Grants 93-2213-E-004-004 and 93-2411-H-002-013 from the National Science Council of Taiwan.

References

D. Coniam. 1997. A preliminary inquiry into using corpus word frequency data in the automatic generation of English language cloze tests. *Computer Assisted Language Instruction Consortium*, 16(2–4):15–33.

P. Deane and K. Sheehan. 2003. Automatic item generation via frame semantics. Education Testing Service: http://www.ets.org/research/dload/ncme03-deane.pdf.

I. Dennis, S. Handley, P. Bradon, J. Evans, and S. Nestead. 2002. Approaches to modeling item-generative tests. In *Item generation for test development* (Irvine and Kyllonen, 2002), pages 53–72.

S.-M. Huang, C.-L. Liu, and Z.-M. Gao. 2005. Computer-assisted item generation for listening cloze tests and dictation practice in English. In *Proc. of the 4th Int. Conf. on Web-based Learning*. to appear.

S. H. Irvine and P. C. Kyllonen, editors. 2002. *Item Generation for Test Development*. Lawrence Erlbaum Associates, Mahwah, NJ.

D. Lin. 1998. Dependency-based evaluation of MINIPAR. In *Proc. of the Workshop on the Evaluation of Parsing Systems in the 1st Int. Conf. on Language Resources and Evaluation*.

C.-L. Liu, C.-H. Wang, and Z.-M. Gao. 2005. Using lexical constraints for enhancing computer-generated multiple-choice cloze items. *Int. J. of Computational Linguistics and Chinese Language Processing*, 10:to appear.

C.-L. Liu. 2005. Using mutual information for adaptive item comparison and student assessment. *J. of Educational Technology & Society*, 8(4):to appear.

C. D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge.

C. J. Poel and S. D. Weatherly. 1997. A cloze look at placement testing. *Shiken: JALT (Japanese Assoc. for Language Teaching) Testing & Evaluation SIG Newsletter*, 1(1):4–10.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pages 133–142.

P. Resnik. 1997. Selectional preference and sense disambiguation. In *Proc. of the Applied NLP Workshop on Tagging Text with Lexical Semantics: Why, What and How*, pages 52–57.

J. C. Reynar and A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proc. of the Conf. on Applied Natural Language Processing*, pages 16–19.

K. M. Sheehan, P. Deane, and I. Kostin. 2003. A partially automated system for generating passage-based multiple-choice verbal reasoning items. Paper presented at the Nat'l Council on Measurement in Education Annual Meeting.

V. Stevens. 1991. Classroom concordancing: vocabulary materials derived from relevant authentic text. *English for Specific Purposes*, 10(1):35–46.

Y. Wilks and M. Stevenson. 1997. Combining independent knowledge sources for word sense disambiguation. In *Proc. of the Conf. on Recent Advances in Natural Language Processing*, pages 1–7.