ACL-05

The Second Workshop on Building Educational Applications Using NLP

Proceedings of the Workshop

29 June 2005
University of Michigan
Ann Arbor, Michigan, USA

Production and Manufacturing by
Omnipress Inc.
Post Office Box 7214
Madison, WI 53707-7214

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
75 Paterson Street, Suite 9
New Brunswick, NJ 08901
USA
Tel: +1-732-342-9100
Fax: +1-732-342-9339
acl@aclweb.org

Introduction

The two main research areas in educational applications, automated evaluation of students free-responses and intelligent tutoring systems (ITS), have developed fairly autonomously within the NLP community. We made progress toward bridging this gap in the First Workshop on Building Educational Applications Using NLP in 2003, where researchers in a wide variety of educational applications met in Edmonton to share their work and ideas - both in the speech- and text-based communities. Papers dealt with automated evaluation of essay-length texts and classification of brief responses that students enter into a tutoring system. Other research that was reported included exploring the value of using grammar checking within a tutoring system, comparing speech- and text-based tutoring systems, and automatically generating multiple-choice questions.

There continues to be a significant and fast-growing body of research toward developing educational applications that incorporate NLP. This has become apparent as, since the First Workshop in 2003, subsequent workshops have been held by scientists working in this field (InSTIL/ICALL 2004 Symposium on Computer Assisted Learning and the eLearning International Workshop, COLING 2004).

The themes in the 2005 workshop fall into four broad categories. Several papers explore the automated assessment of written text - a field that is fast becoming mainstream. These papers describe methods to score essay-length responses, evaluate content-based short answer responses, and identify plagiarized material. Other papers look at methods for generating assessment questions automatically. A third major focus is in teaching language skills - both speech and text-based. Finally, two papers evaluate tools that NLP software developers can use to build educational applications.

We hope that this workshop will continue to facilitate communication between researchers who work on all types of instructional applications, for K-12, undergraduate, graduate school and professional or industrial settings. Our goal is to continue to expose the NLP research community to these technologies with the hope that they may see novel opportunities for use of their tools in educational applications.

We wish to thank the members of the Program Committee, listed below, for reviewing the large number of workshop submissions on a very tight schedule. We owe special thanks to Slava Andreyev for production work on these proceedings (also on a tight schedule!)

Jill Burstein
Claudia Leacock

Organizers:

Jill Burstein, Educational Testing Service
Claudia Leacock, Pearson Knowledge Technologies

Program Committee:

Martin Chodorow, Hunter College, City University of New York
Paul Deane, Educational Testing Service
Derrick Higgins, Educational Testing Service
Karen Kukich, National Science Foundation
Michael Levinson, Queens University, Canada
Diane Litman, University of Pittsburgh
Karen Lochbaum, Pearson Knowledge Technologies
Daniel Marcu, Information Sciences Institute/University of Southern California
Thomas Morton, Educational Testing Service
Jack Mostow, Carnegie Mellon University
Carolyn Penstein Rose, University of Pittsburgh
Frederique Segond, Xerox Research Centre Europe, France
C-C Shei, University of Swansea, UK
Randall Sparks, Pearson Knowledge Technologies
Jana Sukkarieh, Oxford University, UK
Lee Schwartz, Microsoft Corp.
Susanne Wolff, Princeton University
Keiji Yasuda, ATP, Japan
Ming Zhou, Microsoft Asia, Beijing

Table of Contents

Conference Program

Wednesday, June 29, 2005

8:45–9:00 Opening Remarks

10:30–11:00 Break

12:30–02:00 Lunch

03:30–04:00 Break

Session 1

09:00–09:30 *Applications of Lexical Information for Algorithmically Composing Multiple-Choice Cloze Items*
Chao-Lin Liu, Chun-Hung Wang, Zhao-Ming Gao and Shang-Ming Huang

09:30–10:00 *Automatic Short Answer Marking*
Stephen G Pulman and Jana Z Sukkarieh

Short Students Talks

10:00–10:10 *A real-time multiple-choice question generation for language testing – a preliminary study–*
Ayako Hoshino and Nakagawa Hiroshi

10:10–10:20 *Predicting Learning in Tutoring with the Landscape Model of Memory*
Arthur Ward and Diane Litman

10:20–10:30 *Towards Intelligent Search Assistance for Inquiry-Based Learning*
Weijian Xuan and Meilan Zhang

Wednesday, June 29, 2005 (continued)

Session 2

11:00–11:30 *Automatic Essay Grading with Probabilistic Latent Semantic Analysis*
Tuomo Kakkonen, Niko Myller, Jari Timonen and Erkki Sutinen

11:30–12:00 *Using Syntactic Information to Identify Plagiarism*
Ozlem Uzuner, Boris Katz and Thade Nahnsen

12:00–12:30 *Towards a Prototyping Tool for Behavior Oriented Authoring of Conversational Agents for Educational Applications*
Gahgene Gweon, Jaime Arguello, Carol Pai, Regan Carey, Zachary Zaiss and Carolyn Rosé

Session 3

02:00–02:30 *Direkt Profil: A System for Evaluating Texts of Second Language Learners of French Based on Developmental Sequences*
Jonas Granfeldt, Pierre Nugues, Emil Persson, Lisa Persson, Fabian Kostadinov, Malin Ågren and Suzanne Schlyter

02:30–03:00 *Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions*
Eiichiro Sumita, Fumiaki Sugaya and Seiichi Yamamoto

03:00–03:30 *Evaluating State-of-the-Art Treebank-style Parsers for Coh-Metrix and Other Learning Technology Environments*
Christian F. Hempelmann, Vasile Rus, Arthur C. Graesser and Danielle S. McNamara

Session 4

04:00–04:30 *A Software Tool for Teaching Reading Based on Text-to-Speech Letter-to-Phoneme Rules*
Marian Macchi and Dan Kahn

04:30–05:00 *Situational language training for hotel receptionists*
Frédérique Segond, Thibault Parmentier, Roberta Stock, Ran Rosner and Mariola Usteran Muela

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.

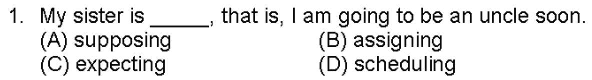
NLP techniques serve to generate multiple-choice cloze items in different ways. (For brevity, we use *cloze items* or *items* for *multiple-choice cloze items* henceforth.) One may create sentences from scratch by applying template-based methods (Dennis et al., 2002) or more complex methods based on 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. ©Association for Computational Linguistics, 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 *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.

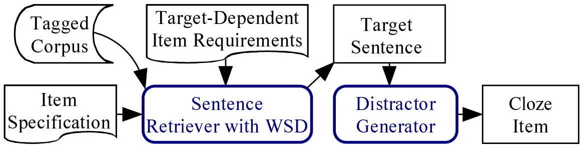
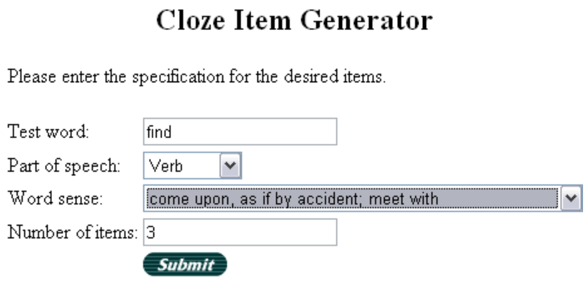


Figure 2: Main components of our item generator

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.



Cloze Item Generator

Please enter the specification for the desired items.

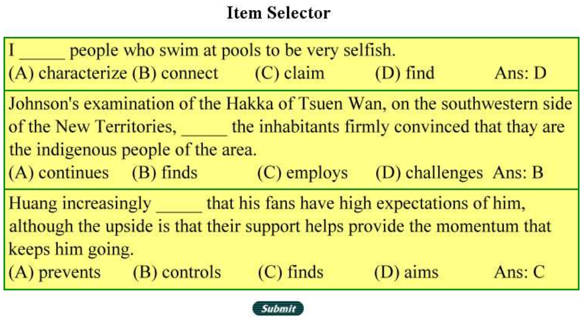
Test word: find
Part of speech: Verb
Word sense: come upon, as if by accident; meet with
Number of items: 3

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



Item Selector

I _____ people who swim at pools to be very selfish.
(A) characterize (B) connect (C) claim (D) find Ans: D

Johnson's examination of the Hakka of Tsuen Wan, on the southwestern side of the New Territories, _____ the inhabitants firmly convinced that thay are the indigenous people of the area.
(A) continues (B) finds (C) employs (D) challenges Ans: B

Huang increasingly _____ that his fans have high expectations of him, although the upside is that their support helps provide the momentum that keeps him going.
(A) prevents (B) controls (C) finds (D) aims Ans: C

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:

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

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].

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

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.

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

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.

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

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).

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

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%

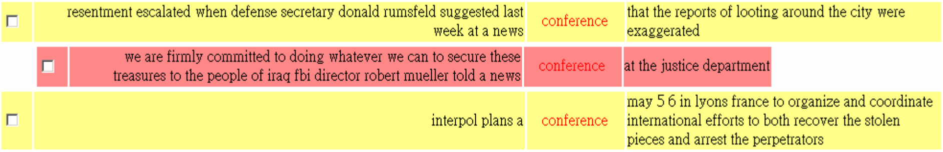
Table 3: Uniqueness of answers

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.



□	resentment escalated when defense secretary donald rumsfeld suggested last week at a news	conference	that the reports of looting around the city were exaggerated
□	we are firmly committed to doing whatever we can to secure these treasures to the people of iraq fbi director robert mueller told a news	conference	at the justice department
□	interpol plans a	conference	may 5 6 in lyons france to organize and coordinate international efforts to both recover the stolen pieces and arrest the perpetrators

Figure 5: A phonetic concordancer

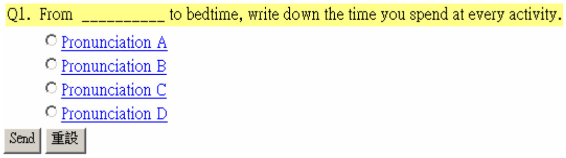
6.3 More Applications

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.



Q1. From _________ to bedtime, write down the time you spend at every activity.

○ Pronunciation A
○ Pronunciation B
○ Pronunciation C
○ Pronunciation D

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.

Automatic Short Answer Marking

Stephen G. Pulman
Computational Linguistics Group,
University of Oxford.
Centre for Linguistics and Philology,
Walton St., Oxford, OX1 2HG, UK
sgp@clg.ox.ac.uk

Jana Z. Sukkarieh
Computational Linguistics Group,
University of Oxford.
Centre for Linguistics and Philology,
Walton St., Oxford, OX1 2HG, UK
Jana.Sukkarieh@clg.ox.ac.uk

Abstract

Our aim is to investigate computational linguistics (CL) techniques in marking short free text responses automatically. Successful automatic marking of free text answers would seem to presuppose an advanced level of performance in automated natural language understanding. However, recent advances in CL techniques have opened up the possibility of being able to automate the marking of free text responses typed into a computer without having to create systems that fully understand the answers. This paper describes some of the techniques we have tried so far vis-à-vis this problem with results, discussion and description of the main issues encountered.[1]

1. Introduction

Our aim is to investigate computational linguistics techniques in marking short free text responses automatically. The free text responses we are dealing with are answers ranging from a few words up to 5 lines. These answers are for factual science questions that typically ask candidates to state, describe, suggest, explain, etc. and where there is an objective criterion for right and wrong. These questions are from an exam known as GCSE (General Certificate of Secondary Education): most 16 year old students take up to 10 of these in different subjects in the UK school system.

2. The Data

Consider the following GCSE biology question:

Statement of the question	***Marking Scheme (full mark 3)***[2]
The blood vessels help to maintain normal body temperature. **Explain how the blood vessels reduce heat loss if the body temperature falls below normal.**	**any three:** vasoconstriction; explanation (of vasoconstriction); less blood flows to / through the skin / close to the surface; less heat loss to air/surrounding/from the blood / less radiation / conduction / convection;

Here is a sample of real answers:

1. all the blood move faster and dose not go near the top of your skin they stay close to the moses
2. The blood vessels stops a large ammount of blood going to the blood capillary and sweat gland. This prents the presonne from sweating and loosing heat.
3. When the body falls below normal the blood vessels 'vasoconstrict' where the blood supply to the skin is cut off, increasing the metabolism of the

[1] This is a 3-year project funded by the University of Cambridge Local Examinations Syndicate.

[2] X;Y/D/K;V is equivalent to saying that each of X, [L]={Y, D,K}, and V deserves 1 mark. The student has to write only 2 of these to get the full mark. [L] denotes an equivalence class i.e. Y, D, K are equivalent. If the student writes Y and D s/he will get only 1 mark.

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 9–16, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

body. This prevents heat loss through the skin, and causes the body to shake to increase metabolism.

It will be obvious that many answers are ungrammatical with many spelling mistakes, even if they contain more or less the right content. Thus using standard syntactic and semantic analysis methods will be difficult. Furthermore, even if we had fully accurate syntactic and semantic processing, many cases require a degree of inference that is beyond the state of the art, in at least the following respects:

- **The need for reasoning and making inferences:** a student may answer with *we do not have to wait until Spring,*which only implies the marking key *it can be done at any time*. Similarly, an answer such as *don't have sperm or egg* will get a 0 incorrectly if there is no mechanism to infer *no fertilisation*.
- **Students tend to use a negation of a negation (for an affirmative):** An answer like *won't be done only at a specific time* is the equivalent to *will be done at any time*. An answer like *it is not formed from more than one egg and sperm* is the same as saying *formed from one egg and sperm*. This category is merely an instance of the need for more general reasoning and inference outlined above. We have given this case a separate category because here, the wording of the answer is not very different, while in the general case, the wording can be completely different.
- **Contradictory or inconsistent information:** Other than logical contradiction like *needs fertilisation and does not need fertilisation*, an answer such as *identical twins have the same chromosomes but different DNA* holds inconsistent scientific information that needs to be detected.

Since we were sceptical that existing deep processing NL systems would succeed with our data, we chose to adopt a shallow processing approach, trading robustness for complete accuracy. After looking carefully at the data we also discovered other issues which will affect assessment of the accuracy of any automated system, namely:

- **Unconventional expression for scientific knowledge:** Examiners sometimes accept unconventional or informal ways of expressing scientific knowledge, for example, 'sperm and egg get together' for 'fertilisation'.
- **Inconsistency across answers:** In some cases, there is inconsistency in marking across answers. Examiners sometimes make mistakes under pressure. Some biological information is considered relevant in some answers and irrelevant in others.

In the following, we describe various implemented systems and report on their accuracy.
We conclude with some current work and suggest a road map.

3. Information Extraction for Short Answers

In our initial experiments, we adopted an Information Extraction approach (see also Mitchell et al. 2003). We used an existing Hidden Markov Model part-of-speech (HMM POS) tagger trained on the Penn Treebank corpus, and a Noun Phrase (NP) and Verb Group (VG) finite state machine (FSM) chunker. The NP network was induced from the Penn Treebank, and then tuned by hand. The Verb Group FSM (i.e. the Hallidayean constituent consisting of the verbal cluster without its complements) was written by hand. Relevant missing vocabulary was added to the tagger from the tagged British National Corpus (after mapping from their tag set to ours), and from examples encountered in our training data. The tagger also includes some suffix-based heuristics for guessing tags for unknown words.

In real information extraction, template merging and reference resolution are important components. Our answers display little redundancy, and are typically less than 5 lines long, and so template merging is not necessary. Anaphors do not occur very frequently, and when they do, they often refer back to entities introduced in the text of the question (to which the system does not have access). So at the cost of missing some correct answers, the information extraction components really consists of little more than a set of patterns applied to the tagged and chunked text.

We wrote our initial patterns by hand, although we are currently working on the development of a tool to take most of the tedious effort out of this task. We base the patterns on recurring head words or phrases, with syntactic annotation where necessary, in the training data. Consider the following example training answers:

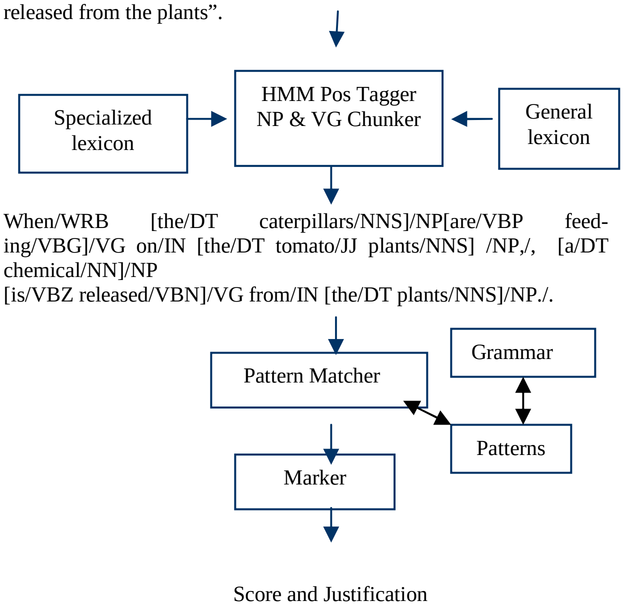
the egg after fertilisation splits in two	the fertilised egg has divided into two
The egg was fertilised it split in two	One fertilised egg splits into two
one egg fertilised which split into two	1 sperm has fertilized an egg.. that split into two

These are all paraphrases of *It is the same fertilised egg/embryo*, and variants of what is written above could be captured by a pattern like:

singular_det + <fertilised egg> +{<split>; <divide>; <break>} + {in, into} + <two_halves>, where
<fertilised egg> = NP with the content of 'fertilised egg'
singular_det = {the, one, 1, a, an}
<split> = {split, splits, splitting, has split, etc.}
<divide> = {divides, which divide, has gone, being broken...}
<two_halves> = {two, 2, half, halves}
etc.

The pattern basically is all the paraphrases collapsed into one. It is essential that the patterns use the linguistic knowledge we have at the moment, namely, the part-of-speech tags, the noun phrases and verb groups. In our previous example, the requirement that <fertilised egg> is an NP will exclude something like '*one sperm has fertilized an egg'* while accept something like '*an egg which is fertilized ...*'.

System Architecture:
"When the caterpillars are feeding on the tomato plants, a chemical is released from the plants".



When/WRB [the/DT caterpillars/NNS]/NP[are/VBP feeding/VBG]/VG on/IN [the/DT tomato/JJ plants/NNS] /NP,/, [a/DT chemical/NN]/NP
[is/VBZ released/VBN]/VG from/IN [the/DT plants/NNS]/NP./.

Score and Justification

Table 1 gives results for the current version of the system. For each of 9 questions, the patterns were developed using a training set of about 200 marked answers, and tested on 60 which were not released to us until the patterns had been written. Note that the full mark for each question ranges between 1-4.

Question	Full Mark	% Examiner Agreement	% Mark Scheme Agreement
1	2	89.4	93.8
2	2	91.8	96.5
3	2	84	94.2
4	1	91.3	94.2
5	2	76.4	93.4
6	3	75	87.8
7	1	95.6	97.5
8	4	75.3	86.1
9	2	86.6	92
Average	----	84	93

Table 1. Results for the manually-written IE approach.

Column 3 records the percentage agreement between our system and the marks assigned by a human examiner. As noted earlier, we detected a certain amount of inconsistency with the marking scheme in the grades actually awarded. Column 4 reflects the degree of agreement between the grades awarded by our system and those which would have been awarded by following the marking scheme consistently. Notice that agreement is correlated with the mark scale: the system appears less accurate on multi-part questions. We adopted an extremely strict measure, requiring an exact match. Moving to a pass-fail criterion produces much higher agreement for questions 6 and 8.

4. Machine Learning

Of course, writing patterns by hand requires expertise both in the domain of the examination, and in computational linguistics. This requirement makes the commercial deployment of a system like this problematic, unless specialist staff are taken on. We have therefore been experimenting with ways in which a short answer marking system might be developed rapidly using machine learning methods on a training set of marked answers.

Previously (Sukkarieh et al. 2003) we reported the results we obtained using a simple Nearest

Neighbour Classification techniques. In the following, we report our results using three different machine learning methods: Inductive Logic progamming (ILP), decision tree learning(DTL) and Naive Bayesian learning (Nbayes). ILP (Progol, Muggleton 1995) was chosen as a representative symbolic learning method. DTL and NBayes were chosen following the Weka (Witten and Frank, 2000) injunction to `try the simple things first'. With ILP, only 4 out of the 9 questions shown in the previous section were tested, due to resource limitations. With DTL and Nbayes, we conducted two experiments on all 9 questions. The first experiments show the results with non-annotated data; we then repeat the experiments with annotated data. Annotation in this context is a lightweight activity, simply consisting of a domain expert highlighting the part of the answer that deserves a mark. Our idea was to make this as simple a process as possible, requiring minimal software, and being exactly analogous to what some markers do with pencil and paper. As it transpired, this was not always straightforward, and does not mean that the training data is noiseless since sometimes annotating the data accurately requires non-adjacent components to be linked: we could not take account of this.

4.1 Inductive Logic Programming

For our problem, for every question, the set of training data consists of students' answers, to that question, in a Prologised version of their textual form, with no syntactic analysis at all initially. We supplied some `background knowledge' predicates based on the work of (Junker et al. 1999). Instead of using their 3 Prolog basic predicates, however, we only defined 2, namely, *word-pos(Text,Word,Pos)* which represents words and their position in the text and *window(Pos2-Pos1,Word1,Word2)* which represents two words occurring within a *Pos2-Pos1* window distance.

After some initial experiments, we believed that a stemmed and tagged training data should give better results and that *window* should be made independent to occur in the logic rules learned by Progol. We used our POS tagger mentioned above and the Porter stemmer (Porter 1980). We set the Progol noise parameter to 10%, i.e. the rules do not have to fit the training data perfectly. They can be more general. The percentages of agreement are shown in table 2[3]. The results reported are on a 5-fold cross validation testing and the agreement is on whether an answer is marked 0 or a mark >0, i.e. pass-fail, against the human examiner scores. The baseline is the number of answers with the most common mark multiplied by 100 over the total number of answers.

Question	**Baseline**	**% of agreement**
6	**51,53**	74,87
7	**73,63**	90,50
8	**57,73**	74,30
9	**70,97**	65,77
Average	**71,15**	**77,73**

Table 2. Results using ILP.

The results of the experiment are not very promising. It seems very hard to learn the rules with ILP. Most rules state that an answer is correct if it contains a certain word, or two certain words within a predefined distance. A question such as 7, though, scores reasonably well. This is because Progol learns a rule such as *mark(Answer) only if word-pos(Answer,'shiver', Pos)* which is, according to its marking scheme, all it takes to get its full mark, 1. ILP has in effect found the single keyword that the examiners were looking for.

Recall that we only have ~200 answers for training. By training on a larger set, the learning algorithm may be able to find more structure in the answers and may come up with better results. However, the rules learned may still be basic since, with the background knowledge we have supplied the ILP learner always tries to find simple and small predicates over (stems of) keywords.

4.2 Decision Tree Learning and Bayesian Learning

In our marking problem, seen as a machine learning problem, the outcome or target attribute is well-defined. It is the mark for each question and its values are {0,1, …, full_mark}. The input attributes could vary from considering each word to be an attribute or considering deeper linguistic features like a head of a noun phrase or a verb group to be an attribute, etc. In the following experiments, each word in the answer was considered to be an attribute. Furthermore, Rennie et al. (2003)

[3] Our thanks to our internship student, Leonie IJzereef for the results in table 2.

propose simple heuristic solutions to some problems with naïve classifiers. In Weka, Complement of Naïve Bayes (CNBayes) is a refinement to the selection process that Naïve Bayes makes when faced with instances where one outcome value has more training data than another. This is true in our case. Hence, we ran our experiments using this algorithm also to see if there were any differences. The results reported are on a 10-fold cross validation testing.

4.2.1 Results on Non-Annotated data

We first considered the non-annotated data, that is, the answers given by students in their raw form. The first experiment considered the values of the marks to be {0,1, …, full_mark} for each question. The results of decision tree learning and Bayesian learning are reported in the columns titled DTL1 and NBayes/CNBayes1. The second experiment considered the values of the marks to be either 0 or >0, i.e. we considered two values only, pass and fail. The results are reported in columns DTL2 and NBayes2/CNBayes2. The baseline is calculated the same way as in the ILP case. Obviously, the result of the baseline differs in each experiment only when the sum of the answers with marks greater than 0 exceeds that of those with mark 0. This affected questions 8 and 9 in Table 3 below. Hence, we took the average of both results. It was no surprise that the results of the second experiment were better than the first on questions with the full mark >1, since the number of target features is smaller. In both experiments, the complement of Naïve Bayes did slightly better or equally well on questions with a full mark of 1, like questions 4 and 7 in the table, while it resulted in a worse performance on questions with full marks >1.

Ques.	Base-line	*DTL1*	*N/CNBayes1*	N/CNBayes2	DTL2
1	69	*73.52*	*73.52 / 66.47*	81.17 / 73.52	76.47
2	54	*62.01*	*65.92 /61.45*	73.18/ 68.15	62.56
3	46	*68.68*	*72.52 / 61.53*	93.95 / 92.85	93.4
4	58	*69.71*	*75.42 / 76*	75.42 / 76	69.71
5	54	*60.81*	*66.66 / 53.21*	73.09 / 73.09	67.25
6	51	*47.95*	*59.18 / 52.04*	81.63 /77.55	67.34
7	73	*88.05*	*88.05 / 88.05*	88.05 / 88.05	88.05
8	42	*41.75*	*43.29 / 37.62*	70.10/ 69.07	72.68
9	60	*61.82*	*67.20 / 62.36*	79.03 / 76.88	76.34
Ave.	60.05	*63.81*	*67.97/62.1*	79.51/77.3	74.86

Table 3. Results for Bayesian learning and decision tree learning on non-annotated data.

Since we were using the words as attributes, we expected that in some cases stemming the words in the answers would improve the results. Hence, we experimented with the answers of 6, 7, 8 and 9 from the list above but there was only a tiny improvement (in question 8). Stemming does not necessarily make a difference if the attributes/words that make a difference appear in a root form already. The lack of any difference or worse performance may also be due to the error rate in the stemmer.

4.2.2 Results on Annotated data

We repeated the second experiments with the annotated answers. The baseline for the new data differs and the results are shown in Table 4.

Question	Baseline	DTL	NBayes/CNBayes
1	58	74.87	86.69 / 81.28
2	56	75.89	77.43 / 73.33
3	86	90.68	95.69 / 96.77
4	62	79.08	79.59 / 82.65
5	59	81.54	86.26 / 81.97
6	69	85.88	92.19 / 93.99
7	79	88.51	91.06 / 89.78
8	78	94.47	96.31 / 93.94
9	79	85.6	87.12 / 87.87
Average	69.56	84.05	88.03 / 86.85

Table 4. Results for Bayesian learning and decision tree learning on annotated data.

As we said earlier, annotation in this context simply means highlighting the part of the answer that deserves 1 mark (if the answer has >=1 mark), so for e.g. if an answer was given a 2 mark then at least two pieces of information should be highlighted and answers with 0 mark stay the same. Obviously, the first experiments could not be conducted since with the annotated answers the mark is either 0 or 1. Bayesian learning is doing better than DTL and 88% is a promising result. Furthermore, given the results of CNBayes in Table 3, we expected that CNBayes would do better on questions 4 and 7. However, it actually did better on questions 3, 4, 6 and 9. Unfortunately, we cannot see a pattern or a reason for this.

5. Comparison of Results

IE did best on all the questions before annotating the data as it can be seen in Fig. 1. Though, the training data for the machine learning algorithms is tiny relative to what usually such algorithms consider, after annotating the data, the performance of NBayes on questions 3, 6 and 8 were better than IE. This is seen in Fig. 2. However, as we said earlier in section 2, the percentages shown for IE method are on the whole mark while the results of DTL and Nbayes, after annotation, are calculated on pass-fail.

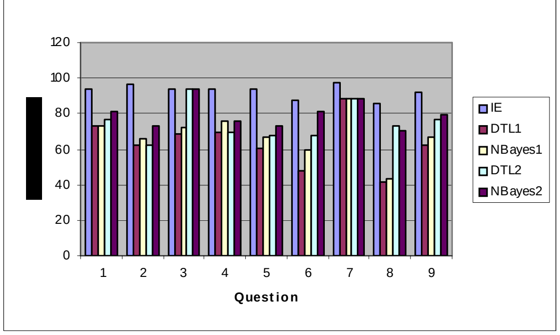


Fig. 1. IE vs DTL & Nbayes pre-annotation

In addition, in the pre-annotation experiments reported in Fig. 1, the NBayes algorithm did better than that of DTL. Post-annotation, results in Fig. 2 show, again, that NBayes is doing better than the DTL algorithm. It is worth noting that, in the annotated data, the number of answers whose marks are 0 is less than in the answers whose mark is 1, except for questions 1 and 2. This may have an effect on the results.

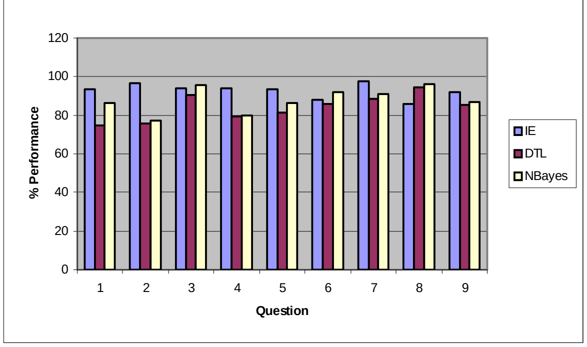


Fig.2. IE vs DTL & NBayes post-annotation

Moreover, after getting the worse performance in NBayes2 before annotation, question 8 jumps to best performance. The rest of the questions maintained the same position more or less, with question 3 always coming nearest to the top (see Fig. 3). We noted that Count(Q,1)-Count(Q,0) is highest for questions 8 and 3, where Count(Q,N) is, for question Q, the number of answers whose mark is N. Also, the improvement of performance for question 8 in relation to Count(8,1) was not surprising, since question 8 has a full-mark of 4 and the annotation's role was an attempt at a one-to-one correspondence between an answer and 1 mark.

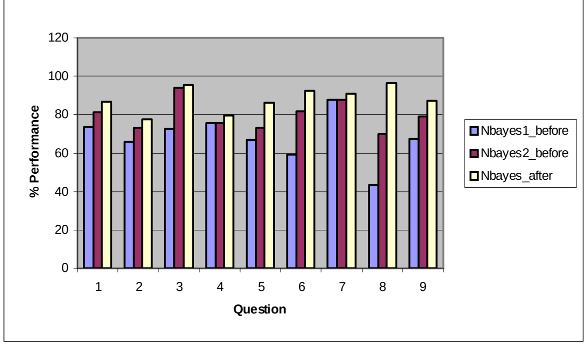


Fig. 3. NBayes before and after annotation

On the other hand, question 1 that was in seventh place in DTL2 before annotation, jumps down to the worst place after annotation. In both cases, namely, NBayes2 and DTL2 after annotation, it seems reasonable to hypothesize that P(Q1) is better than P(Q2) if Count(Q1,1)-Count(Q1,0) >> Count(Q2,1)-Count(Q2,0), where P(Q) is the percentage of agreement for question Q.

As they stand, the results of agreement with given marks are encouraging. However, the models that the algorithms are learning are very naïve in the sense that they depend on words only. Unlike the IE approach, it would not be possible to provide a reasoned justification for a student as to why they have got the mark they have. One of the advantages to the pattern-matching approach is that it is very easy, knowing which patterns have matched, to provide some simple automatic feed-back to the student as to which components of the answer were responsible for the mark awarded.

We began experimenting with machine learning methods in order to try to overcome the IE customisation bottleneck. However, our experience so far has been that in short answer marking (as opposed to essay marking) these methods are, while promising, not accurate enough at present to be a real alternative to the hand-crafted, pattern-matching approach. We should instead think of them either as aids to the pattern writing process – for example, frequently the decision trees that are learned are quite intuitive, and suggestive of useful patterns – or perhaps as complementary supporting assessment techniques to give extra confirmation.

6. Other work

Several other groups are working on this problem, and we have learned from all of them. Systems which share properties with ours are C-Rater, developed by Leacock et al. (2003) at the Educational Testing Service(ETS), the IE-based system of Mitchell et al. (2003) at Intelligent Assessment Technologies, and Rosé et al. (2003) at Carnegie Mellon University. The four systems are being developed independently, yet it seems they share similar characteristics. Commercial and resource pressures currently make it impossible to try these different systems on the same data, and so performance comparisons are meaningless: this is a real hindrance to progress in this area. The field of automatic marking really needs a MUC-style competition to be able to develop and assess these techniques and systems in a controlled and objective way.

7. Current and Future Work

The manually-engineered IE approach requires skill, much labour, and familiarity with both domain and tools. To save time and labour, various researchers have investigated machine-learning approaches to learn IE patterns (Collins et al. 1999, Riloff 1993). We are currently investigating machine learning algorithms to learn the patterns used in IE (an initial skeleton-like algorithm can be found in Sukkarieh et al. 2004).

We are also in the process of evaluating our system along two dimensions: firstly, how long it takes, and how difficult it is, to customise to new questions; and secondly, how easy it is for students to use this kind of system for formative assessment. In the first trial, a domain expert (someone other than us) is annotating some new training data for us. Then we will measure how long it takes us (as computational linguists familiar with the system) to write IE patterns for this data, compared to the time taken by a computer scientist who is familiar with the domain and with general concepts of pattern matching but with no computational linguistics expertise. We will also assess the performance accuracy of the resulting patterns.

For the second evaluation, we have collaborated with UCLES to build a web-based demo which will be trialled during May and June 2005 in a group of schools in the Cambridge (UK) area. Students will be given access to the system as a method of self-assessment. Inputs and other aspects of the transactions will be logged and used to improve the IE pattern accuracy. Students' reactions to the usefulness of the tool will also be recorded. Ideally, we would go on to compare the future examination performance of students with and without access to the demo, but that is some way off at present.

References

Collins, M. and Singer, Y. 1999. *Unsupervised models for named entity classification.* Proceedings Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 189-196.

Junker, M, M. Sintek & M. Rinck 1999. *Learning for Text Categorization and Information Extraction with ILP*. In: Proceedings of the 1st Workshop on Learning Language in Logic, Bled, Slovenia, 84-93.

Leacock, C. and Chodorow, M. 2003. *C-rater: Automated Scoring of Short-Answer Questions*. Computers and Humanities 37:4.

Mitchell, T. Russell, T. Broomhead, P. and Aldridge, N. 2003. *Computerized marking of short-answer free-text responses*. Paper presented at the 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK.

Muggleton, S. 1995. *Inverting Entailment and Progol*. In: New Generation Computing, 13:245-286.

Porter, M.F. 1980. *An algorithm for suffix stripping*, Program, 14(3):130-137.

Rennie, J.D.M., Shih, L., Teevan, J. and Karger, D. 2003 *Tackling the Poor Assumptions of Naïve Bayes TextClassifiers*. http://haystack.lcs.mit.edu/papers/rennie.icml03.pdf.

Riloff, E. 1993. *Automatically constructing a dictionary for information extraction tasks*. Proceedings 11th National Conference on Artificial Intelligence, pp. 811-816.

Rosé, C. P. Roque, A., Bhembe, D. and VanLehn, K. 2003. A hybrid text classification approach for analysis of student essays. In Building Educational Applications Using Natural Language Processing, pp. 68-75.

Sukkarieh, J. Z., Pulman, S. G. and Raikes N. 2003. *Auto-marking: using computational linguistics to score short, free text responses*. Paper presented at the 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK.

Sukkarieh, J. Z., Pulman, S. G. and Raikes N. 2004. *Auto-marking2: An update on the UCLES-OXFORD University research into using computational linguistics to score short, free text responses*. Paper presented at the 30th annual conference of the International Association for Educational Assessment (IAEA), Philadelphia, USA.

Witten, I. H. Eibe, F. 2000. *Data Mining.* Academic Press.

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. ©Association for Computational Linguistics, 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/`

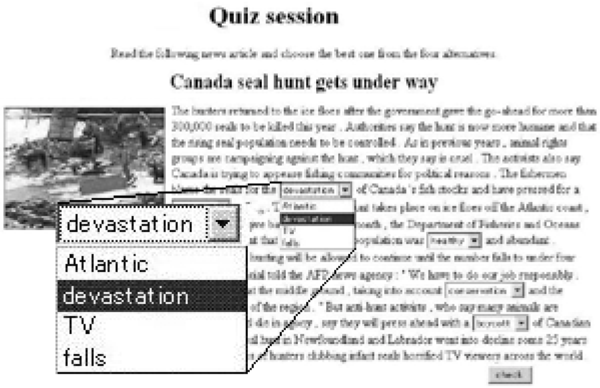
data, and the instances are classified into *true* or *false*. The positions of the blanks are decided according to the certainty of the classification so the blanks (i.e. questions) are generated as many as the user has specified.

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.



Quiz session

Read the following news article and choose the best one from the four alternatives.

Canada seal hunt gets under way

The hunters returned to the ice floes after the government gave the go-ahead for more than 300,000 seals to be killed this year . Authorities say the hunt is now more humane and that the rising seal population needs to be controlled . As in previous years , animal rights groups are campaigning against the hunt , which they say is cruel . The activists also say Canada is trying to appease fishing communities for political reasons . The fishermen blame the seals for the [devastation ▾] of Canada 's fish stocks and have pressed for a [illegible] that takes place on ice floes off the Atlantic coast , [illegible] month , the Department of Fisheries and Oceans [illegible] population was [healthy ▾] and abundant . [illegible] hunting will be allowed to continue until the number falls to under four [illegible] told the AFP news agency : ' We have to do our job responsibly . [illegible] the middle ground , taking into account [conservation ▾] and the [illegible] of the region . ' But anti-hunt activists , who say many animals are [illegible] die in agony , say they will press ahead with a [boycott ▾] of Canadian [illegible] hunt in Newfoundland and Labrador went into decline some 25 years [illegible] hunters clubbing infant seals horrified TV viewers across the world

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.

Predicting Learning in Tutoring with the Landscape Model of Memory

Arthur Ward
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, Pa., 15260, USA
artward@cs.pitt.edu

Diane Litman
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, Pa., 15260, USA
litman@cs.pitt.edu

Abstract

A Landscape Model analysis, adopted from the text processing literature, was run on transcripts of tutoring sessions, and a technique developed to count the occurrence of key physics points in the resulting connection matrices. This point-count measure was found to be well correlated with learning.

1 Introduction

Human one-to-one tutoring often yields significantly higher learning gains than classroom instruction (Bloom, 1984). This difference motivates natural language tutoring research, which hopes to discover which aspects of tutorial dialogs correlate with learning. Much of this research focuses on various dialog characteristics. For example, (Graesser et al., 1995) argue that the important components of tutoring include question answering and explanatory reasoning. In other work (Litman et al., 2004) examine dialog characteristics that can be identified automatically, such as ratio of student to tutor words, and average turn length.

In this paper, rather than look at characteristics of the tutoring dialog itself, we feed the dialog into a computational model of student memory, in which we then find a measure correlated with learning. This “Landscape Model” (van den Broek et al., 1996) proves useful for predicting how much students remember from tutoring sessions, as measured by their learning gains.

We will first briefly describe the Landscape Model. Then we will describe the tutoring experiments from which we draw a corpus of dialogs, and how the model was applied to this corpus. Finally, we cover the model’s success in predicting learning.

2 The Landscape Model

The Landscape Model was designed by van den Broek et al. (1996) to simulate human reading comprehension. In this model, readers process a text sentence-by-sentence. Each sentence contains explicitly mentioned concepts which are added into working memory. In addition, the reader may re-instantiate concepts from earlier reading cycles or from world knowledge in an effort to maintain a coherent representation. Concepts are entered into working memory with initial activation values, which then decay over subsequent reading cycles.

After concepts are entered, the model calculates connection strengths between them. Two concepts that are active in working memory at the same time will be given a link. The higher the levels of concept activation, the stronger the link will be. Van den Broek et al. (1996) give this formula for calculating link strengths: $\sum_{i=1}^{l} A_{xi} A_{yi}$

This defines the strength of the connection between concepts x and y as the product of their activations (A) at each cycle i, summed over all reading cycles.

Two matrices result from these calculations. The first is a matrix of activation strengths, showing all the active concepts and their values for each reading cycle. The second is a square matrix of link values showing the strength of the connection between

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 21–24, Ann Arbor, June 2005.

each pair of concepts. Van den Broek et al. (1996) demonstrate a method for extracting a list of individual concepts from these matrices in order of their link strengths, starting with the strongest concept. They show a correlation between this sequence and the order in which subjects name concepts in a free-recall task.

In van den Broek's original implementation, this model was run on short stories. In the current work, the model is extended to cover a corpus of transcripts of physics tutoring dialogs. In the next section we describe this corpus.

3 Corpus of Tutoring Transcripts

Our corpus was taken from transcripts collected for the ITSPOKE intelligent tutoring system project (Litman and Silliman, 2004). This project has collected tutoring dialogs with both human and computer tutors. In this paper, we describe results using the human tutor corpus.

Students being tutored are first given a pre-test to gauge their physics knowledge. After reading instructional materials about physics, they are given a qualitative physics problem and asked to write an essay describing its solution. The tutor (in our case, a human tutor), examines this essay, identifies points of the argument that are missing or wrong, and engages the student in a dialog to remediate those flaws. When the tutor is satisfied that the student has produced the correct argument, the student is allowed to read an "ideal" essay which demonstrates the correct physics argument. After all problems have been completed, the student is given a post-test to measure overall learning gains. Fourteen students did up to ten problems each. The final data set contained 101,181 student and tutor turns, taken from 128 dialogs.

4 Landscape Model & Tutoring Corpus

Next we generated a list of the physics concepts necessary to represent the main ideas in the target solutions. Relevant concepts were chosen by examining the "ideal" essays, representing the complete argument for each problem. One hundred and twelve such concepts were identified among the 10 physics problems. Simple keyword matching was used to identify these concepts as they appeared in each line of the dialog. A small sample of these concepts and their keywords is shown in Table 1.

Concept Name	Keywords
above	above, over
acceleration	acceleration,accelerating
action	action, reaction
affect	experience,experienced
after	after, subsequent
air friction	air resistance, wind resistance
average	mean
ball	balls, sphere
before	before, previous
beside	beside, next to

Table 1: Examples of concepts and keywords

Each concept found was entered into the working memory model with an initial activation level, which was made to decay on subsequent turns using a formula modeled on van den Broek (1996). Concept strengths are assumed to decay by 50% every turn for three turns, after which they go to zero. A sample portion of a transcript showing concepts being identified, entering and decaying is shown in Table 2. Connections between concepts were then calculated as described in section two. A portion of a resulting concept link matrix is shown in Table 3.

It should be noted that the Landscape model has some disadvantages in common with other bag-of-words methods. For example, it loses information about word order, and does not handle negation well.

As mentioned in section two, van den Broek et al. created a measure that predicted the order in which individual concepts would be recalled. For our task, however, such a measure is less appropriate. We are less interested, for example, in the specific order in which a student remembers the concepts "car" and "heavier," than we are in whether the student remembers the whole idea that a heavier car accelerates less. To measure these constellations of concepts, we created a new measure of idea strength.

5 Measuring Idea Strength

The connection strength matrices described above encode data about which concepts are present in each dialog, and how they are connected. To extract useful information from these matrices, we used the idea of a "point." Working from the ideal essays, we identified a set of key points important for the solution of each physics problem. These key points

Turn	Text	Concepts			
		car	heavier	acceleration	cause
Student	I don't know how to answer this it's got to be slower, cause, it's the car is heavier but	5	5	0	0
Tutor	yeah, just write whatever you think is appropriate	2.5	2.5	0	0
Student	ok,	1.25	1.25	0	0
Essay	The rate of acceleration will decrease if the first car is towing a second, because even though the force of the car's engine is the same, the weight of the car is double	5	0.625	5	5
Student	ok	2.5	0	2.5	2.5
Tutor	qualitatively,um, what you say is right, you have correctly recognized that the force, uh, exerted will be the same in both cases,uh, now, uh, how is force related to acceleration?	1.25	0	5	1.25

Table 2: Portion of a transcript, showing activation strengths per turn

	car	heavier	acceleration	cause	decelerates	decrease
car	0	35.9375	115.234375	102.34375	33.203125	33.2
heavier	0	0	3.125	3.125	3.125	3.13
acceleration	0	0	0	107.8125	42.1875	42.19
cause	0	0	0	0	33.203125	33.2
decelerates	0	0	0	0	0	66.41
decrease	0	0	0	0	0	0

Table 3: Portion of link value table, showing connection strengths between concepts

are modeled after the points the tutor looks for in the student's essay and dialog. For example, in the "accelerating car" problem, one key point might be that the car's acceleration would decrease as the car got heavier. The component concepts of this point would be "car," "acceleration," "decrease," and "heavier." If this point were expressed in the dialog or essay, we would expect these concepts to have higher-than-average connection strengths between them. If this point were not expressed, or only partially expressed, we would expect lower connection strengths among its constituent concepts.

The strength of a point, then, was defined as the sum of strengths of all the links between its component concepts. Call the point in the example above "p_i." point p_i has n = 4 constituent concepts, and to find its strength we would sum the link strengths between their pairs: "car-acceleration," "car-decrease," "car-heavier," "acceleration-decrease,", "acceleration-heavier," and "decrease-heavier." Using values from Table 3, the total strength for the point would therefore be:

$$pointStr_{p_i n} = 115.23 + 33.2 + 35.94 + \\ 42.19 + 3.13 + 3.13 = 232.81.$$

For each point, we determined if its connections were significantly stronger than the average. We generate a reference average $AvgStr_n$ by taking 500 random sets of n concepts from the same dialog and averaging their link weights, where n is the number of concepts in the target point [1]. If the target point was found to have a significantly (p $<$ .05 in a t-test) larger value than the mean of this random sample, that point was above threshold, and considered to be present in the dialog.

The number of above-threshold points was added up over all dialogs for each student. The total point-count for student S is therefore:

$$pointCount_S = \sum_{i=1}^{P} T(pointStr_{p_i n}, AvgStr_n)$$

Where P is the total number of points in all dialogs, and T is a threshold function which returns 1 if $pointStr_{p_i n} > AvgStr_n$, and 0 otherwise.

Fifty-seven key points were identified among the ten problems, with each point containing between two and five concepts. The next section describes how well this point-count relates to learning.

[1] 500 was chosen as the largest feasible sample size given runtime limitations

6 Results: Point Counts & Learning

We first define “concept-count” to be the number of times physics concepts were added to the activation strength matrix. This corresponds to each “5” in Table 2. Now we look at a linear model with post-test score as the dependant variable, and pre-test score and concept-count as independent variables. In this model pre-test score is significant, with a p-value of .029, but concept-count is not, with a p-value of .270. The adjusted R squared for the model is .396

Similarly, in a linear model with pre-test score and point-count as independent variables, pre-test score is significant with a p-value of .010 and point-count is not, having a p-value of .300. The adjusted R squared for this model is .387.

However, the situation changes in a linear model with pre-test score, concept-count and point-count as independent variables, and post-test score as the dependent variable. Pre-test is again significant with a p-value of .002. Concept-count and point-count are now both significant with p-values of .016 and .017, respectively. The adjusted R-squared for this model rises to .631.

These results indicate that our measure of points, as highly associated constellations of concepts, adds predictive power over simply counting the occurrence of concepts alone. The number of concept mentions does not predict learning, but the extent to which these concepts are linked into relevant points in the Landscape memory model is correlated with learning.

7 Discussion

Several features of the resulting model are worth mentioning. First, the Landscape Model is a model of memory, and our measurements can be interpreted as a measure of what the student is remembering from the tutoring session taken as a whole.

Second, the point-counts are taken from the entire dialog, rather than from either the tutor or student’s contributions. Other results suggest that it would be interesting to investigate the extent to which these points are produced by the student, the tutor, or both...and what effect their origin might have on their correlation with learning. For example, (Chi et al., 2001) investigated student-centered, tutor-centered and interactive hypotheses of tutoring and found that students learned just as effectively when tutor feedback was suppressed. They suggest, among other things, that students self-construction of knowledge was encouraging deep learning.

8 Summary and Future Work

We have shown that the Landscape Model yields a measure significantly correlated with learning in our human-human tutoring corpus. We hope to continue this work by investigating the use of well researched NLP methods in creating the input matrix. In addition, machine learning methods could be used to optimize the various parameters in the model, such as the decay rate, initial activation value, and point strength threshold.

9 Acknowledgments

We thank Tessa Warren, Chuck Perfetti and Franz Schmalhofer for early advice on this project. This research is supported by ONR (N00014-04-1-0108).

References

B. Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13:4–16.

M. Chi, S. Siler, H. Jeong, T. Yamauchi, and R. Hausman. 2001. Learning from human tutoring. *Cognitive Science*, 25:471–533.

A. Graesser, N. Person, and J. Magliano. 1995. Collaborative dialog patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology*, 9:359–387.

D. Litman and S. Silliman. 2004. ITSPOKE: An intelligent tutoring spoken dialogue system. In *Companion Proc. of the Human Language Technology Conf: 4th Meeting of the North American Chap. of the Assoc. for Computational Linguistics*.

Diane J. Litman, Carolyn P. Rosé, Kate Forbes-Riley, Kurt VanLehn, Dumisizwe Bhembe, and Scott Silliman. 2004. Spoken versus typed human and computer dialogue tutoring. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems(ITS)*. Maceio, Brazil.

P. van den Broek, K. Risden, C.R. Fletcher, and R. Thurlow. 1996. A landscape view of reading: Fluctuating patterns of activation and the construction of a stable memory representation. In B.K. Britton and A.C. Graesser, editors, *Models of understanding text*, pages 165–187. Mahweh, NJ: Lawrence Erlbaum Associates.

Towards Intelligent Search Assistance for Inquiry-Based Learning

Weijian Xuan
MHRI
University of Michigan
Ann Arbor, MI 48109
wxuan@umich.edu

Meilan Zhang
School of Education
University of Michigan
Ann Arbor, MI 48109
meilanz@umich.edu

Abstract

In Online Inquiry-Based Learning (OIBL) learners search for information to answer driving questions. While learners conduct sequential related searches, the search engines interpret each query in isolation, and thus are unable to utilize task context. Consequently, learners usually get less relevant search results. We are developing a NLP-based search agent to bridge the gap between learners and search engines. Our algorithms utilize contextual features to provide user with search term suggestions and results re-ranking. Our pilot study indicates that our method can effectively enhance the quality of OIBL.

1 Introduction

Major science education standards call on students to engage in Online Inquiry-Based Learning where they pose scientific Driving Questions (DQ), plan their search, collect and analyze online information, and synthesize their findings into an argument. In collaboration with National Science Digital Library (NSDL), we are developing an integrated Online Inquiry-Based Learning Environment (OIBLE), called IdeaKeeper (Quintana and Zhang, 2004), to help learners fulfill the promise of OIBL. IdeaKeeper is among the first reported OIBLE that integrates various online search engines with support for inquiry planning, information search, analysis and synthesis.

Our observation reveals that searching is one of the bottlenecks impeding students' learning experience. Students demonstrate various problems in search. First, they repeatedly search for very similar keywords on search engines. Second, they are usually unable to develop effective search terms. Many search keywords students generate are either too broad or too narrow. Although learners have specific search purposes, many times they are unable to express the purposes in keyword-based queries. In fact, by analyzing the search logs, we found that the average query length is only about 2 words. In such typical cases in OIBL, informative contexts are not presented in queries, and thus the requests become ambiguous. As a result, the search engines may not interpret the query as the learners intended to. Therefore, the results are usually not satisfactory. Given the self-regulated nature of OIBL and limited self-control skills of K-12 students, the problem is even more serious, as students may shift their focus off the task if they constantly fail to find relevant information for their DQ.

2 Related Work

In Information Retrieval field, many algorithms based on relevance feedback are proposed (Buckley, et al., 1994; Salton and Buckley, 1990). However, current general web search engines are still unable to interactively improve research results. In NLP domain, there are considerable efforts on Question Answering systems that attempt to answer a question by returning concise facts. While some QA systems are promising (Harabagiu, et al., 2000; Ravichandran and Hovy, 2002), they can only handle factual questions as in TREC (Voorhees, 2001), and the context for the whole task is largely not considered. There are proposals on using context in search. Huang et al (2001) proposed a term suggestion method for interactive web search. More existing systems that utilize contextual information in search are reviewed by Lawrence (2000). However, one problem is that "context" is defined differently in each

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 25–28, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

study. Few attempts target at inquiry-based learning, which has some unique features, e.g., DQ/SQ.

We are developing an OnLine Inquiry Search Assistance (OLISA). OLISA applies Natural Language Processing (NLP) and Information Retrieval (IR) techniques to provide students query term suggestions and re-rank results returned from search engines by the relevance to the current query as well as to the DQ. OLISA is not a built-in component of IdeaKeeper, but can be very easily plugged into IdeaKeeper or other OIBL systems as a value-added search agent. The main advantage of OLISA is that it utilizes the context of the whole learning task. Our pilot study demonstrated that it is a simple and effective initiative toward automatically improving the quality of web search in OIBLE.

3 Method

3.1 Utilizing Learning Context

OLISA acquires search context by parsing OIBL logs and by monitoring search history. For example, in the planning phase of a learning task, IdeaKeeper asks students to input DQ, Sub-Questions (SQs), potential keywords, and to answer some questions such as “what do I know”, “what do I want to know”, etc.

The context information is represented as bag-of-words feature vectors. To calculate the vectors, we first remove common terms. We compiled a corpus of 30 million words from 6700 full-length documents collected from diverse resources. Word frequencies are calculated for 168K unique words in the corpus. A word is considered common if it is in the 1000 most frequent word list. Remaining words are stemmed using Porter’s algorithm (Porter, 1980).

All contextual information are combined to form a main feature vector ($W_1^{(c)}, W_2^{(c)}, \cdots, W_n^{(c)}$), where $W_i^{(c)}$ is the weight of the *ith* term in combined context. It’s defined by product of term frequency (*tf)* and inverse document frequency (*idf*).

Comparing with traditional tf measure, we do not assign a uniform weight to all words in context. Rather, we consider DQ/SQ and the current query more important than the rest of context. We define their tf differently from other context.

$$tf_i^{(dq)} = (1 + \ln(\#wordInContext / \#wordInDQ)) * tf_i^{(dq)} \quad (1)$$

The $tf_i^{(sq)}$ is calculated similarly. For the term frequency of current query $tf_i^{(q)}$, we assign it a larger weight as it represents the current information needs:

$$tf_i^{(q)} = (\#wordInContext / \#wordInQuery) * tf_i^{(q)} \quad (2)$$

Therefore,

$$tf_i^{(c)} = tf_i^{(q)} + tf_i^{(dq)} + tf_i^{(sq)} + tf_i^{(other)} \quad (3)$$

The inverse document frequency is defined by:

$$idf_i^{(c)} = \ln(N / n_i) \quad (4)$$

where N is total number of documents in the corpus, and n_i is the number of documents containing *ith* term. The term weight is defined by:

$$W_i^{(c)} = \frac{\ln(1 + tf_i^{(c)}) \times idf_i^{(c)}}{\sqrt{\sum \ln^2(1 + tf_i^{(c)}) \times \sum idf_i^{(c)2}}} \quad (5)$$

These context feature vectors are calculated for later use in re-ranking search results.

Meanwhile, we use Brill's tagger (Brill, 1995) to determine parts of speech (POS) of words in DQ/SQ. Heuristic rules (Zhang and Xuan, 2005) based on POS are used to extract noun phrases.

Noun phrases containing words with high term weight are considered as keyphrases. The keyphrase weight is defined by:

$$W_{P_i}^{(c)} = \sum_j W_j^{(c)} \quad (where\, W_j^{(c)} \in Phrase\ P_i) \quad (6)$$

3.2 Term Suggestion

When a user commits a query, OLISA will first search it on selected search engines (Google as default). If the total hit exceeds certain threshold (2 million as default), we consider the query potentially too general. In addition to the original query, we will call term suggestion component to narrow down the search concept by expanding the query. WordNet (Fellbaum, 1998) is used during the expansion. Below is the outline of our heuristic algorithm in generating term suggestion.

```
for each keyword in original query do
    if the keyword is part of a keyphrase then
        form queries by merging each phrase with the original query
    if multiple keyphrases are involved then
        select up to #maxPhrase keyphrases with highest weights
if #queries>0 then return queries
for each keyword that has hyponyms in WordNet do
    if some hyponym occur at least once in learning context then
        form queries by merging the hyponym with the original query
    else form suggestions by merging the hyponym with the original query
if #queries>0 or #suggestions> 0 then return queries and suggestions
for each keyword in original query that has synonyms in WordNet do
    if some synonym is part of a keyphrase then
        form suggestions by merging keywords in phrase with original query
    if multiple keyphrases are involved then
        select up to #maxPhrase keyphrases with highest weights
return suggestions
```

On the other hand, if the total hit is below certain threshold, the query is potentially too specific. Thus term suggestion component is called to generalize the query. The procedure is similar to the algorithm above, but will be done in the reverse direction. For example, keywords will replace phrases and hypernyms will replace hyponyms. Since there are cases where learners desire specific search terms, both original and expanded queries will be submitted, and results for the former will be presented at the top of the returned list.

If no new queries are constructed, OLISA will return the results from original query along with suggestions. Otherwise, OLISA will send requests for each expanded query to selected search engines. Since by default we return up to R_T*=100* search engine results to user, we will extract the top R_Q*=*R_T*/(#newQuery+1)* entries from results of each new query and original query. These results will be re-ranked by an algorithm that we will describe later. Then the combined results will be presented to the user in IdeaKeeper along with a list of expanded queries and suggestions.

3.3 Query Reformulation

From our observation, in OIBLE students often submit questions in natural language. However, most of the time, such type of queries does not return desirable results. Therefore, we loosely follow Kwok (2001) to reformulate queries. We apply Link Grammar Parser (Sleator and Temperley, 1993) to parse sentence structure. For example, one student asked "What is fat good for". The parser generates the following linkage:

```
    +----------------Xp----------------+
    |       +----------Bsw---------+   |
    |       |    +----Paf----+     |   |
    +---Wq--+    +-SIs+      +-MVp-+   |
    |       |    |    |      |     |   |
LEFT-WALL what is.v fat.n good.a for.p ?
```

where "SI" is used in subject-verb inversion. By getting this linkage, we are able to reformulate the query as "fat is good for". Meanwhile, regular expressions are developed to eliminate interrogative words, e.g. "what" and "where".

Search engines may return very different results for the original query and the reformulated queries. For example, for the example above, Google returned 620 hits for the reformulated query, but only 2 hits for the quoted original question. By sending request in both original and reformulated forms, we can significantly improve recall ratio without losing much precision.

3.4 Integrating Multiple Search Engines

We enhanced the searching component of IdeaKeeper by integrating multiple search engines (e.g. Google, AskJeeves, NSDL, etc.). IdeaKeeper will parse and transform search results and present users with a uniform format of results from different search engines. A spelling check function for search keywords is built in OLISA, which combined spelling check results from Google as well as suggestions from our own program based on a local frequency-based dictionary.

3.5 Search Results Re-Ranking

After query reformulation OLISA will send requests to selected search engines. For performance issue, we only retrieve a total of 100 snippets (R_Q snippets from each query) from web search engines. Feature vector is calculated for each snippet in the measure similar to (5), except that tf is actual frequency without assigning additional weight.

The similarity between learning context C and each document D (i.e. snippet) is calculated as:

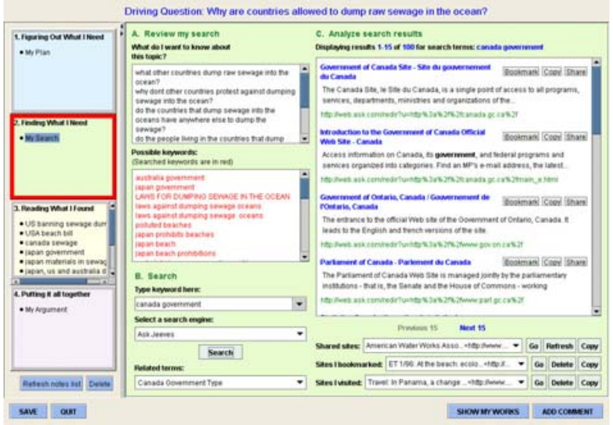
$$Similarity(C,D) = \frac{\sum_i^n W_i^{(c)} W_i^{(d)}}{\sqrt{\sum_i^n W_i^{(c)2} \sum_i^n W_i^{(d)2}}} \tag{7}$$

The higher the similarity score, the more relevant it will be to user's query as well as to the overall learning context.

OLISA re-ranks snippets by similarity scores. To avoid confusion to learners, the snippets from the original query and the expanded queries are reranked independently. R_Q re-ranked results from original query appear at the top as default, followed by other re-ranked results with signs indicating corresponding queries. The expanded queries and further search term suggestions are shown in a dropdown list in IdeaKeeper.

4 Preliminary Results and Discussion

OLISA is under development. While thorough evaluation is needed, our preliminary results demonstrate its effectiveness. We conducted field studies with middle school students for OIBL projects using IdeaKeeper. Fig.1 shows a case of using OLISA search function in IdeaKeeper. By video taping some students' search session, we found that enhanced search functions of OLISA significantly saved students' effort and improve their experience on search. The term suggestions were frequently used in these sessions.



Driving Question: Why are countries allowed to dump raw sewage in the ocean?

3. Reading What I Found
- US banning sewage dum
- USA beach bill
- canada sewage
- japan government
- japan materials in sewag
- japan, us and australia d

What do I want to know about this topic?

what other countries dump raw sewage into the ocean?
why dont other countries protest against dumping sewage into the ocean?
do the countries that dump sewage into the oceans have anywhere else to dump the sewage?
do the people living in the countries that dump

Possible keywords:
(Searched keywords are in red)

australia government
japan government
LAWS FOR DUMPING SEWAGE IN THE OCEAN
laws against dumping sewage oceans
laws against dumping sewage oceans
polluted beaches
japan prohibits beaches
japan beach
japan beach prohibitions

Type keyword here: canada government

Select a search engine: Ask Jeeves

Related terms: Canada Government Type

Displaying results 1-15 of 100 for search terms: canada government

Government of Canada Site - Site du gouvernement du Canada
The Canada Site, le Site du Canada, is a single point of access to all programs, services, departments, ministries and organizations of the...
http://web.ask.com/redir?u=http%3a%2f%2fcanada.gc.ca%2f

Introduction to the Government of Canada Official Web Site - Canada
Access information on Canada, its **government**, and federal programs and services organized into categories. Find an MP's e-mail address, the latest...
http://web.ask.com/redir?u=http%3a%2f%2fcanada.gc.ca%2fmain_e.html

Government of Ontario, Canada / Gouvernement de l'Ontario, Canada
The entrance to the official Web site of the Government of Ontario, Canada. It leads to the English and french versions of the site.
http://web.ask.com/redir?u=http%3a%2f%2fwww.gov.on.ca%2f

Parliament of Canada - Parlement du Canada
The Parliament of Canada Web Site is managed jointly by the parliamentary institutions - that is, the Senate and the House of Commons - working
http://web.ask.com/redir?u=http%3a%2f%2fwww.parl.gc.ca%2f

Shared sites: American Water Works Asso...<http://www...
Sites I bookmarked: ET 1/96: At the beach: ecolo...<http://...
Sites I visited: Travel: In Panama, a change ...<http://www...

Fig. 1 Using OLISA function in IdeaKeeper

Our initials results also demonstrate that calculation on the snippets returned by search engines is simple and efficient. Therefore, we don't need to retrieve each full document behind. We want to point out that in our feature vector calculation each past query is combined into previous context. So the learning context is interactively changing.

Previous research has found that in OIBL projects, students often spend considerable time searching for sites due to their limited search skills. Consequently, students have little time on higher-order cognitive and metacognitive activities, such as evaluation, sense making, synthesis, and reflection. By supporting students' search, OLISA helps student focus more on higher-order activities, which provide rich opportunities for deep learning to occur.

Our future work includes fine-tuning the parameters in our algorithms and conducting more evaluation of each component of OLISA. We are also considering taking into account the snippets or documents users selected, because they also represent user feedback. How to determine the relative weight of words in selected documents, and how to disambiguate polysemies using WordNet or other resources are topics of future research.

References

Brill, E. (1995). *Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging*, Computational Linguistics, 21, 543-565.

Buckley, C., Salton, G. and Allan, J. (1994). *The Effect of Adding Relevance Information in a Relevance Feedback Environment*. Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. Dublin, Ireland, 292-300.

Fellbaum, C., Ed. (1998) *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Harabagiu, S.M., Pasca, M.A. and Maiorano, S.J. (2000) *Experiments with Open-Domain Textual Question Answering*. Proceedings of the 17th conference on Computational linguistics. Saarbrucken, Germany, 292-298.

Huang, C.-K., Oyang, Y.-J. and Chien, L.-F. (2001) *A Contextual Term Suggestion Mechanism for Interactive Web Search*. Web Intelligence, 272-281

Kwok, C., Etzioni, O. and Weld, D. (2001) *Scaling Question answering to the Web*. ACM Transactions on Information Systems, 19, 242-262.

Lawrence, S. (2000) *Context in Web Search*, IEEE Data Engineering Bulletin, 23, 25–32.

Porter, M.F. (1980) *An algorithm for suffix stripping*. Program, 14, 130-137.

Quintana, C. and Zhang, M. (2004) *The Digital IdeaKeeper: Integrating Digital Libraries with a Scaffolded Environment for Online Inquiry*. JCDL'04. Tuscon, AZ, 388-388.

Ravichandran, D. and Hovy, E. (2002) *Learning Surface Text Patterns for a Question Answering System*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, 41-47.

Salton, G. and Buckley, C. (1990) *Improving retrieval performance by relevance feedback*. Journal of the American Society for Information Science, 41, 288-297.

Sleator, D. and Temperley, D. (1993) *Parsing English with a Link Grammar*. Proceedings of the Third International Workshop on Parsing Technologies.

Voorhees, E. (2001) *Overview of the TREC 2001 Question Answering Track*. Proceedings of the 10th Text Retrieval Conference (TREC10). Gaithersburg, MD, 157-165.

Zhang, M. and Xuan, W. (2005) *Towards Discovering Linguistic Features from Scientific Abstracts*. Proceedings of the 26th ICAME and the 6th AAACL conference. Ann Arbor, MI.

Automatic Essay Grading with Probabilistic Latent Semantic Analysis

Tuomo Kakkonen, Niko Myller, Jari Timonen, and Erkki Sutinen
Department of Computer Science, University of Joensuu
P.O. Box 111, FI-80101 Joensuu, FINLAND
firstname.lastname@cs.joensuu.fi

Abstract

Probabilistic Latent Semantic Analysis (PLSA) is an information retrieval technique proposed to improve the problems found in Latent Semantic Analysis (LSA). We have applied both LSA and PLSA in our system for grading essays written in Finnish, called Automatic Essay Assessor (AEA). We report the results comparing PLSA and LSA with three essay sets from various subjects. The methods were found to be almost equal in the accuracy measured by Spearman correlation between the grades given by the system and a human. Furthermore, we propose methods for improving the usage of PLSA in essay grading.

1 Introduction

The main motivations behind developing automated essay assessment systems are to decrease the time in which students get feedback for their writings, and to reduce the costs of grading. The assumption in most of the systems is that the grades given by the human assessors describe the true quality of an essay. Thus, the aim of the systems is to “simulate” the grading process of a human grader and a system is usable only if it is able to perform the grading as accurately as human raters. An automated assessment system is not affected by errors caused by lack of consistency, fatigue or bias, thus it can help achieving better accuracy and objectivity of assessment (Page and Petersen, 1995).

There has been research on automatic essay grading since the 1960s. The earliest systems, such as PEG (Page and Petersen, 1995), based their grading on the surface information from the essay. For example, the number of words and commas were counted in order to determine the quality of the essays (Page, 1966). Although these kinds of systems performed considerably well, they also received heavy criticism (Page and Petersen, 1995). Some researchers consider the use of natural language as a feature for human intelligence (Hearst et al., 2000) and writing as a method to express the intelligence. Based on that assumption, taking the surface information into account and ignoring the meanings of the content is insufficient. Recent systems and studies, such as e-rater (Burstein, 2003) and approaches based on LSA (Landauer et al., 1998), have focused on developing the methods which determine the quality of the essays with more analytic measures such as syntactic and semantic structure of the essays. At the same time in the 1990s, the progress of natural language processing and information retrieval techniques have given the opportunity to take also the meanings into account.

LSA has produced promising results in content analysis of essays (Landauer et al., 1997; Foltz et al., 1999b). Intelligent Essay Assessor (Foltz et al., 1999b) and Select-a-Kibitzer (Wiemer-Hastings and Graesser, 2000) apply LSA for assessing essays written in English. In Apex (Lemaire and Dessus, 2001), LSA is applied to essays written in French. In addition to the essay assessment, LSA is applied to other educational applications. An intelligent tutoring system for providing help for students (Wiemer-

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 29–36, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

Hastings et al., 1999) and Summary Street (Steinhart, 2000), which is a system for assessing summaries, are some examples of other applications of LSA. To our knowledge, there is no system utilizing PLSA (Hofmann, 2001) for automated essay assessment or related tasks.

We have developed an essay grading system, *Automatic Essay Assessor* (AEA), to be used to analyze essay answers written in Finnish, although the system is designed in a way that it is not limited to only one language. It applies both course materials, such as passages from lecture notes and course textbooks covering the assignment-specific knowledge, and essays graded by humans to build the model for assessment. In this study, we employ both LSA and PLSA methods to determine the similarities between the essays and the comparison materials in order to determine the grades. We compare the accuracy of these methods by using the Spearman correlation between computer and human assigned grades.

The paper is organized as follows. Section 2 explains the architecture of AEA and the used grading methods. The experiment and results are discussed in Section 3. Conclusions and future work based on the experiment are presented in Section 4.

2 AEA System

We have developed a system for automated assessment of essays (Kakkonen et al., 2004; Kakkonen and Sutinen, 2004). In this section, we explain the basic architecture of the system and describe the methods used to analyze essays.

2.1 Architecture of AEA

There are two approaches commonly used in the essay grading systems to determine the grade for the essay:

1. The essay to be graded is compared to the human-graded essays and the grade is based on the most similar essays' grades; or

2. The essay to be graded is compared to the essay topic related materials (e.g. textbook or model essays) and the grade is given based on the similarity to these materials.

In our system, AEA (Kakkonen and Sutinen, 2004), we have combined these two approaches. The relevant parts of the learning materials, such as chapters of a textbook, are used to train the system with assignment-specific knowledge. The approaches based on the comparison between the essays to be graded and the textbook have been introduced in (Landauer et al., 1997; Foltz et al., 1999a; Lemaire and Dessus, 2001; Hearst et al., 2000), but have been usually found less accurate than the methods based on comparison to prescored essays. Our method attempts to overcome this by combining the use of course content and prescored essays. The essays to be graded are not directly compared to the prescored essays with for instance k-nearest neighbors method, but prescored essays are used to determine the similarity threshold values for grade categories as discussed below. Prescored essays can also be used to determine the optimal dimension for the reduced matrix in LSA as discussed in Kakkonen et al. (2005).

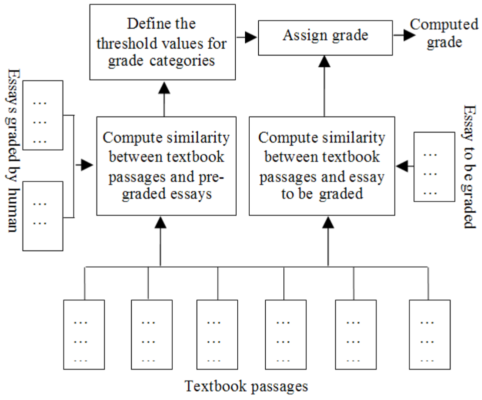

Define the threshold values for grade categories
Assign grade
Computed grade
Essay s graded by human
Compute similarity between textbook passages and pre-graded essays
Compute similarity between textbook passages and essay to be graded
Essay to be graded
Textbook passages

Figure 1: The grading process of AEA.

Figure 1 illustrates the grading process of our system. The texts to be analyzed are added into *word-by-context matrix* (WCM), representing the number of occurrences of each unique word in each of the contexts (e.g. documents, paragraphs or sentences). In WCM M, cell M_{ij} contains the count of the word i occurrences in the context j. As the first step in analyzing the essays and course materials, the lemma of each word form occurring in the texts must be found. We have so far applied AEA only to essays written in Finnish. Finnish is morphologically more complex than English, and word forms are formed by adding suffixes into base forms. Because of that, base forms have to be used instead of inflectional forms when building the WCM, especially if a relatively small corpus is utilized. Furthermore, several words can become synonyms when suffixes are added to them, thus making the word sense disambiguation necessary. Hence, instead of just stripping suffixes, we apply a more sophisticated method, a morphological parser and disambiguator, namely Constraint Grammar parser for Finnish (FINCG) to produce the lemmas for each word (Lingsoft, 2005). In addition, the most commonly occurring words (stopwords) are not included in the matrix, and only the words that appear in at least two contexts are added into the WCM (Landauer et al., 1998). We also apply entropy-based term weighting in order to give higher values to words that are more important for the content and lower values to words with less importance.

First, the comparison materials based on the relevant textbook passages or other course materials are modified into machine readable form with the method described in the previous paragraph. The vector for each context in the comparison materials is marked with Y_i. This WCM is used to create the model with LSA, PLSA or another information retrieval method. To compare the similarity of an essay to the course materials, a query vector X_j of the same form as the vectors in the WCM is constructed. The query vector X_j representing an essay is added or *folded in* into the model build with WCM with the method specific way discussed later. This folded-in query $\tilde{X}_j$ is then compared to the model of each text passage $\tilde{Y}_i$ in the comparison material by using a similarity measure to determine the similarity value. We have used the cosine of the angle between ($\tilde{X}_j$, $\tilde{Y}_i$), to measure the similarity of two documents. The *similarity score* for an essay is calculated as the sum of the similarities between the essay and each of the textbook passages.

The document vectors of manually graded essays are compared to the textbook passages, in order to determine the similarity scores between the essays and the course materials. Based on these measures, threshold values for the grade categories are defined as follows: the grade categories, $g_1, g_2, \ldots, g_C$, are associated with similarity value limits, $l_1, l_2, \ldots, l_{C+1}$, where C is the number of grades, and $l_{C+1} = \infty$ and normally $l_1 = 0$ or $-\infty$. Other category limits $l_i, 2 \leq i \leq C$, are defined as weighted averages of the similarity scores for essays belonging to grade categories g_i and g_{i-1}. Other kinds of formulas to define the grade category limits can be also used.

The grade for each essay to be graded is then determined by calculating the similarity score between the essay and the textbook passages and comparing the similarity score to the threshold values defined in the previous phase. The similarity score S_i of an essay d_i is matched to the grade categories according to their limits in order to determine the correct grade category as follows: `For each` i, $1 \leq i \leq C$, `if` $l_i < S_i \leq l_{i+1}$ `then` $d_i \in g_i$ `and break`.

2.2 Latent Semantic Analysis

Latent Semantic Analysis (LSA) (Landauer et al., 1998) is a corpus-based method used in information retrieval with vector space models. It provides a means of comparing the semantic similarity between the source and target texts. LSA has been successfully applied to automate giving grades and feedback on free-text responses in several systems as discussed in Section 1. The basic assumption behind LSA is that there is a close relationship between the meaning of a text and the words in that text. The power of LSA lies in the fact that it is able to map the essays with similar wordings closer to each other in the vector space. The LSA method is able to strengthen the similarity between two texts even when they do not contain common words. We describe briefly the technical details of the method.

The essence of LSA is dimension reduction based on the singular value decomposition (SVD), an algebraic technique. SVD is a form of factor analysis, which reduces the dimensionality of the original WCM and thereby increases the dependency between contexts and words (Landauer et al., 1998). SVD is defined as $X = T_0 S_0 {D_0}^T$, where X is the preprocessed WCM and T_0 and D_0 are orthonormal matrices representing the words and the contexts. S_0 is a diagonal matrix with singular values. In the dimension reduction, the k highest singular values in S_0 are selected and the rest are ignored. With this operation, an approximation matrix $\tilde{X}$ of the original matrix X is acquired. The aim of the dimension reduction is to reduce “noise” or unimportant details and to allow the underlying semantic structure to become evident (Deerwester et al., 1990).

In information retrieval and essay grading, the queries or essays have to be folded in into the model in order to calculate the similarities between the documents in the model and the query. In LSA, the folding in can be achieved with a simple matrix multiplication: $\tilde{X}_q = X_q^T T_0 S_0^{-1}$, where X_q is the term vector constructed from the query document with preprocessing, and T_0 and S_0 are the matrices from the SVD of the model after dimension reduction. The resulting vector $\tilde{X}_q$ is in the same format as the documents in the model.

The features that make LSA suitable for automated grading of essays can be summarized as follows. First, the method focuses on the content of the essay, not on the surface features or keyword-based content analysis. The second advantage is that LSA-based scoring can be performed with relatively low amount of human graded essays. Other methods, such as PEG and e-rater typically need several hundred essays to be able to form an assignment-specific model (Shermis et al., 2001; Burstein and Marcu, 2000) whereas LSA-based IEA system has sometimes been calibrated with as few as 20 essays, though it typically needs more essays (Hearst et al., 2000).

Although LSA has been successfully applied in information retrieval and related fields, it has also received criticism (Hofmann, 2001; Blei et al., 2003). The objective function determining the optimal decomposition in LSA is the Frobenius norm. This corresponds to an implicit additive Gaussian noise assumption on the counts and may be inadequate. This seems to be acceptable with small document collections but with large document collections it might have a negative effect. LSA does not define a properly normalized probability distribution and, even worse, the approximation matrix may contain negative entries meaning that a document contains negative number of certain words after the dimension reduction. Hence, it is impossible to treat LSA as a generative language model and moreover, the use of different similarity measures is limited. Furthermore, there is no obvious interpretation of the directions in the latent semantic space. This might have an effect if also feedback is given. Choosing the number of dimensions in LSA is typically based on an ad hoc heuristics. However, there is research done aiming to resolve the problem of dimension selection in LSA, especially in the essay grading domain (Kakkonen et al., 2005).

2.3 Probabilistic Latent Semantic Analysis

Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 2001) is based on a statistical model which has been called the *aspect model*. The aspect model is a latent variable model for co-occurrence data, which associates unobserved class variables z_k, $k \in \{1, 2, \ldots, K\}$ with each observation. In our settings, the observation is an occurrence of a word w_j, $j \in \{1, 2, \ldots, M\}$, in a particular context d_i, $i \in \{1, 2, \ldots, N\}$. The probabilities related to this model are defined as follows:

- $P(d_i)$ denotes the probability that a word occurrence will be observed in a particular context d_i;
- $P(w_j|z_k)$ denotes the class-conditional probability of a specific word conditioned on the unobserved class variable z_k; and
- $P(z_k|d_i)$ denotes a context specific probability distribution over the latent variable space.

When using PLSA in essay grading or information retrieval, the first goal is to build up the model. In other words, approximate the probability mass functions with machine learning from the training data, in our case the comparison material consisting of assignment specific texts.

Expectation Maximization (EM) algorithm can be used in the model building with maximum likelihood formulation of the learning task (Dempster et al., 1977). In EM, the algorithm alternates between two steps: (i) an *expectation (E)* step where posterior probabilities are computed for the latent variables, based on the current estimates of the parameters, (ii) a *maximization (M)* step, where parameters are updated based on the loglikelihood which depends on the posterior probabilities computed in the E-step. The standard E-step is defined in equation (1).

$$P(z_k|d_i, w_j) = \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_{l=1}^{K} P(w_j|z_l)P(z_l|d_i)} \qquad (1)$$

The M-step is formulated in equations (2) and (3) as derived by Hofmann (2001). These two steps are alternated until a termination condition is met, in this case, when the maximum likelihood function has converged.

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j) P(z_k|d_i, w_j)}{\sum_{m=1}^{M} \sum_{i=1}^{N} n(d_i, w_m) P(z_k|d_i, w_m)} \quad (2)$$

$$P(z_k|d_i) = \frac{\sum_{j=1}^{M} n(d_i, w_j) P(z_k|d_i, w_j)}{\sum_{m=1}^{M} n(d_i, w_m)} \quad (3)$$

Although standard EM algorithm can lead to good results, it may also overfit the model to the training data and perform poorly with unseen data. Furthermore, the algorithm is iterative and converges slowly, which can increase the runtime seriously. Hence, Hofmann (2001) proposes another approach called *Tempered EM* (TEM), which is a derivation of standard EM algorithm. In TEM, the M-step is the same as in EM, but a dampening parameter is introduced into the E-step as shown in equation (4). The parameter β will dampen the posterior probabilities closer to uniform distribution, when $\beta < 1$ and form the standard E-step when $\beta = 1$.

$$P(z_k|d_i, w_j) = \frac{(P(w_j|z_k)P(z_k|d_i))^{\beta}}{\left(\sum_{l=1}^{K} P(w_j|z_l)P(z_l|d_i)\right)^{\beta}} \quad (4)$$

Hofmann (2001) defines the TEM algorithm as follows:

1. Set $\beta := 1$ and perform the standard EM with early stopping.
2. Set $\beta := \eta\beta$ (with $\eta < 1$).
3. Repeat the E- and M-steps until the performance on hold-out data deteriorates, otherwise go to step 2.
4. Stop the iteration when decreasing β does not improve performance on hold-out data.

Early stopping means that the optimization is not done until the model converges, but the iteration is stopped already once the performance on hold-out data degenerates. Hofmann (2001) proposes to use the *perplexity* to measure the generalization performance of the model and the stopping condition for the early stopping. The perplexity is defined as the log-averaged inverse probability on unseen data calculated as in equation (5).

$$\mathcal{P} = \exp\left(-\frac{\sum_{i,j} n'(d_i, w_j) \log P(w_j|d_i)}{\sum_{i,j} n'(d_i, w_j)}\right), \quad (5)$$

where $n'(d_i, w_j)$ is the count on hold-out or training data.

In PLSA, the folding in is done by using TEM as well. The only difference when folding in a new document or query q outside the model is that just the probabilities $P(z_k|q)$ are updated during the M-step and the $P(w_j|z_k)$ are kept as they are. The similarities between a document d_i in the model and a query q folded in to the model can be calculated with the cosine of the angle between the vectors containing the probability distributions $(P(z_k|q))_{k=1}^{K}$ and $(P(z_k|d_i))_{k=1}^{K}$ (Hofmann, 2001).

PLSA, unlike LSA, defines proper probability distributions to the documents and has its basis in Statistics. It belongs to a framework called Latent Dirichlet Allocations (Girolami and Kabán, 2003; Blei et al., 2003), which gives a better grounding for this method. For instance, several probabilistic similarity measures can be used. PLSA is interpretable with its generative model, latent classes and illustrations in N-dimensional space (Hofmann, 2001). The latent classes or topics can be used to determine which part of the comparison materials the student has answered and which ones not.

In empirical research conducted by Hofmann (2001), PLSA yielded equal or better results compared to LSA in the contexts of information retrieval. It was also shown that the accuracy of PLSA can increase when the number of latent variables is increased. Furthermore, the combination of several similarity scores (e.g. cosines of angles between two documents) from models with different number of latent variables also increases the overall accuracy. Therefore, the selection of the dimension is not as crucial as in LSA. The problem with PLSA is that the algorithm used to computate the model, EM or its variant, is probabilistic and can converge to a local maximum. However, according to Hofmann (2001), this is not a problem since the differences between separate runs are small. Flaws in the generative model and the overfitting problem

Set No.	Field	Training essays	Test essays	Grading scale	Course materials	Comp. mat. division type	No. Passages	No. Words
1	Education	70	73	0–6	Textbook	Paragraphs	26	2397
2	Education	70	73	0–6	Textbook	Sentences	147	2397
3	Communications	42	45	0–4	Textbook	Paragraphs	45	1583
4	Communications	42	45	0–4	Textbook	Sentences	139	1583
5	Soft. Eng.	26	27	0–10	*)	Paragraphs	27	965
6	Soft. Eng.	26	27	0–10	*)	Sentences	105	965

Table 1: The essay sets used in the experiment. *) Comparison materials were constructed from the course handout with teacher's comments included and transparencies represented to the students.

have been discussed in Blei et al. (2003).

3 Experiment

3.1 Procedure and Materials

To analyze the performance of LSA and PLSA in the essay assessment, we performed an experiment using three essay sets collected from courses on education, marketing and software engineering. The information about the essay collections is shown in Table 1. Comparison materials were taken either from the course book or other course materials and selected by the lecturer of the course. Furthermore, the comparison materials used in each of these sets were divided with two methods, either into paragraphs or sentences. Thus, we run the experiment in total with six different configurations of materials.

We used our implementations of LSA and PLSA methods as described in Section 2. With LSA, all the possible dimensions (i.e. from two to the number of passages in the comparison materials) were searched in order to find the dimension achieving the highest accuracy of scoring, measured as the correlation between the grades given by the system and the human assessor. There is no upper limit for the number of latent variables in PLSA models as there is for the dimensions in LSA. Thus, we applied the same range for the best dimension search to be fair in the comparison. Furthermore, a linear combination of similarity values from PLSA models (PLSA-C) with predefined numbers of latent variables $K \in \{16, 32, 48, 64, 80, 96, 112, 128\}$ was used just to analyze the proposed potential of the method as discussed in Section 2.3 and in (Hofmann, 2001). When building up all the PLSA models with TEM, we used 20 essays from the training set of the essay collections to determine the early stopping condition with perplexity of the model on unseen data as proposed by Hofmann (2001).

3.2 Results and Discussion

The results of the experiment for all the three methods, LSA, PLSA and PLSA-C are shown in Table 2. It contains the most accurate dimension (column *dim.*) measured by machine-human correlation in grading, the percentage of the same (*same*) and adjacent grades (*adj.*) compared to the human grader and the Spearman correlation (*cor.*) between the grades given by the human assessor and the system.

The results indicate that LSA outperforms both methods using PLSA. This is opposite to the results obtained by Hofmann (2001) in information retrieval. We believe this is due to the size of the document collection used to build up the model. In the experiments of Hofmann (2001), it was much larger, 1000 to 3000 documents, while in our case the number of documents was between 25 and 150. However, the differences are quite small when using the comparison materials divided into sentences. Although all methods seem to be more accurate when the comparison materials are divided into sentences, PLSA based methods seem to gain more than LSA.

In most cases, PLSA with the most accurate dimension and PLSA-C perform almost equally. This is also in contrast with the findings of Hofmann (2001) because in his experiments PLSA-C performed better than PLSA. This is probably also due to the small document sets used. Nevertheless, this means that finding the most accurate dimension is unnecessary, but it is enough to com-

Set No.	LSA dim.	LSA same	LSA adj.	LSA cor.	PLSA dim.	PLSA same	PLSA adj.	PLSA cor.	PLSA-C same	PLSA-C adj.	PLSA-C cor.
1	14	39.7	43.9	0.78	9	31.5	32.9	0.66	34.2	35.6	0.70
2	124	35.6	49.3	0.80	83	37.0	37.0	0.76	35.6	41.1	0.73
3	8	31.1	28.9	0.54	38	24.4	35.6	0.41	17.7	24.4	0.12
4	5	24.4	42.3	0.57	92	35.6	31.1	0.59	22.2	35.6	0.47
5	6	29.6	48.2	0.88	16	18.5	18.5	0.78	11.1	40.1	0.68
6	6	44.4	37.1	0.90	55	33.3	44.4	0.88	14.8	40.7	0.79

Table 2: The results of the grading process with different methods.

bine several dimensions' similarity values. In our case, it seems that linear combination of the similarity values is not the best option because the similarity values between essays and comparison materials decrease when the number of latent variables increases. A topic for a further study would be to analyze techniques to combine the similarity values in PLSA-C to obtain higher accuracy in essay grading. Furthermore, it seems that the best combination of dimensions in PLSA-C depends on the features of the document collection (e.g. number of passages in comparison materials or number of essays) used. Another topic of further research is how the combination of dimensions can be optimized for each essay set by using the collection specific features without the validation procedure proposed in Kakkonen et al. (2005).

Currently, we have not implemented a version of LSA that combines scores from several models but we will analyze the possibilities for that in future research. Nevertheless, LSA representations for different dimensions form a nested sequence because of the number of singular values taken to approximate the original matrix. This will make the model combination less effective with LSA. This is not true for statistical models, such as PLSA, because they can capture a larger variety of the possible decompositions and thus several models can actually complement each other (Hofmann, 2001).

4 Future Work and Conclusion

We have implemented a system to assess essays written in Finnish. In this paper, we report a new extension to the system for analyzing the essays with PLSA method. We have compared LSA and PLSA as methods for essay grading. When our results are compared to the correlations between human and system grades reported in literature, we have achieved promising results with all methods. LSA was slightly better when compared to PLSA-based methods. As future research, we are going to analyze if there are better methods to combine the similarity scores from several models in the context of essay grading to increase the accuracy (Hofmann, 2001). Another interesting topic is to combine LSA and PLSA to compliment each other.

We used the cosine of the angle between the probability vectors as a measure of similarity in LSA and PLSA. Other methods are proposed to determine the similarities between probability distributions produced by PLSA (Girolami and Kabán, 2003; Blei et al., 2003). The effects of using these techniques will be compared in the future experiments.

If the PLSA models with different numbers of latent variables are not highly dependent on each other, this would allow us to analyze the reliability of the grades given by the system. This is not possible with LSA based methods as they are normally highly dependent on each other. However, this will need further work to examine all the potentials.

Our future aim is to develop a semi-automatic essay assessment system (Kakkonen et al., 2004). For determining the grades or giving feedback to the student, the system needs a method for comparing similarities between the texts. LSA and PLSA offer a feasible solution for the purpose. In order to achieve even more accurate grading, we can use some of the results and techniques developed for LSA and develop them further for both methods. We are currently working with an extension to our LSA model that uses standard validation methods for reducing automatically the irrelevant content information in LSA-based essay grading (Kakkonen et al., 2005). In addition, we plan to continue the work with PLSA, since it, being a probabilistic model, introduces new possibilities, for instance, in similarity comparison and feedback giving.

References

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent Dirichlet Allocation. *J. of Machine Learning Research*, 3:993–1022.

J. Burstein and D. Marcu. 2000. Benefits of modularity in an automated scoring system. In *Proc. of the Workshop on Using Toolsets and Architectures to Build NLP Systems, 18th Int'l Conference on Computational Linguistics*, Luxembourg.

J. Burstein. 2003. The e-rater scoring engine: Automated essay scoring with natural language processing. In M. D. Shermis and J. Burstein, editors, *Automated essay scoring: A cross-disciplinary perspective*. Lawrence Erlbaum Associates, Hillsdale, NJ.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing By Latent Semantic Analysis. *J. of the American Society for Information Science*, 41:391–407.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *J. of the Royal Statistical Society*, 39:1–38.

P. W. Foltz, D. Laham, and T. K. Landauer. 1999a. Automated Essay Scoring: Applications to Educational Technology. In *Proc. of Wolrd Conf. Educational Multimedia, Hypermedia & Telecommunications*, Seattle, USA.

P. W. Foltz, D. Laham, and T. K. Landauer. 1999b. The Intelligent Essay Assessor: Applications to Educational Technology. *Interactive Multimedia Electronic J. of Computer-Enhanced Learning*, 1. `http://imej.wfu.edu/articles/1999/2/04/index.asp` (Accessed 3.4.2005).

M. Girolami and A. Kabán. 2003. On an Equivalence between PLSI and LDA. In *Proc. of the 26th Annual Int'l ACM SIGIR Conf. on Research and Development in Informaion Retrieval*, pages 433–434, Toronto, Canada. ACM Press.

M. Hearst, K. Kukich, M. Light, L. Hirschman, J. Burger, E. Breck, L. Ferro, T. K. Landauer, D. Laham, P. W. Foltz, and R. Calfee. 2000. The Debate on Automated Essay Grading. *IEEE Intelligent Systems*, 15:22–37.

T. Hofmann. 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196.

T. Kakkonen and E. Sutinen. 2004. Automatic Assessment of the Content of Essays Based on Course Materials. In *Proc. of the Int'l Conf. on Information Technology: Research and Education*, pages 126–130, London, UK.

T. Kakkonen, N. Myller, and E. Sutinen. 2004. Semi-Automatic Evaluation Features in Computer-Assisted Essay Assessment. In *Proc. of the 7th IASTED Int'l Conf. on Computers and Advanced Technology in Education*, pages 456–461, Kauai, Hawaii, USA.

T. Kakkonen, N. Myller, E. Sutinen, and J. Timonen. 2005. Comparison of Dimension Reduction Methods for Automated Essay Grading. Submitted.

T. K. Landauer, D. Laham, B. Rehder, and M. E. Schreiner. 1997. How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In *Proc. of the 19th Annual Meeting of the Cognitive Science Society*, Mawhwah, NJ. Erlbaum.

T. K. Landauer, P. W. Foltz, and D. Laham. 1998. Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

B. Lemaire and P. Dessus. 2001. A System to Assess the Semantic Content of Student Essays. *J. of Educational Computing Research*, 24:305–320.

Lingsoft. 2005. `http://www.lingsoft.fi/` (Accessed 3.4.2005).

E. B. Page and N. S. Petersen. 1995. The computer moves into essay grading. *Phi Delta Kappan*, 76:561–565.

E. B. Page. 1966. The imminence of grading essays by computer. *Phi Delta Kappan*, 47:238–243.

M. D. Shermis, H. R. Mzumara, J. Olson, and S. Harrington. 2001. On-line Grading of Student Essays: PEG goes on the World Wide Web. *Assessment & Evaluation in Higher Education*, 26:247.

D. Steinhart. 2000. *Summary Street: an LSA Based Intelligent Tutoring System for Writing and Revising Summaries*. Ph.D. thesis, University of Colorado, Boulder, Colorado.

P. Wiemer-Hastings and A. Graesser. 2000. Select-a-Kibitzer: A computer tool that gives meaningful feedback on student compositions. *Interactive Learning Environments*, 8:149–169.

P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser. 1999. Approximate natural language understanding for an intelligent tutor. In *Proc. of the 12th Int'l Artificial Intelligence Research Symposium*, pages 172–176, Menlo Park, CA, USA.

Using Syntactic Information to Identify Plagiarism

Özlem Uzuner, Boris Katz, and Thade Nahnsen
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139
ozlem,boris,tnahnsen@csail.mit.edu

Abstract

Using keyword overlaps to identify plagiarism can result in many false negatives and positives: substitution of synonyms for each other reduces the similarity between works, making it difficult to recognize plagiarism; overlap in ambiguous keywords can falsely inflate the similarity of works that are in fact different in content. Plagiarism detection based on verbatim similarity of works can be rendered ineffective when works are paraphrased even in superficial and immaterial ways. Considering linguistic information related to creative aspects of writing can improve identification of plagiarism by adding a crucial dimension to evaluation of similarity: documents that share linguistic elements in addition to content are more likely to be copied from each other. In this paper, we present a set of low-level syntactic structures that capture creative aspects of writing and show that information about linguistic similarities of works improves recognition of plagiarism (over tfidf-weighted keywords alone) when combined with similarity measurements based on tfidf-weighted keywords.

1 Introduction

To plagiarize is "to steal and pass off (the ideas or words of another) as one's own; [to] use (another's production) without crediting the source; [or] to commit literary theft [by] presenting as new and original an idea or product derived from an existing source".[1] Plagiarism is frequently encountered in academic settings. According to turnitin.com, a 2001 survey of 4500 high school students revealed that "15% [of students] had submitted a paper obtained in large part from a term paper mill or website". Increased rate of plagiarism hurts quality of education received by students; facilitating recognition of plagiarism can help teachers control this damage.

To facilitate recognition of plagiarism, in the recent years many commercial and academic products have been developed. Most of these approaches identify verbatim plagiarism[2] and can fail when works are paraphrased. To recognize plagiarism in paraphrased works, we need to capture similarities that go beyond keywords and verbatim overlaps. Two works that exhibit similarity both in their conceptual content (as indicated by keywords) and in their expression of this content should be considered more similar than two works that are similar only in content. In this context, *content* refers to the story or the information; *expression* refers to the linguistic choices of authors used in presenting the content, i.e., creative elements of writing, such as whether authors tend toward passive or active voice, whether they prefer complex sentences with embedded clauses or simple sentences with independent clauses, as well as combinations of such choices.

Linguistic information can be a source of power for measuring similarity between works based on

[1] www.webster.com

[2] www.turnitin.com

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP, pages 37–44, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

their expression of content. In this paper, we use linguistic information related to the creative aspects of writing to improve recognition of paraphrased documents as a first step towards plagiarism detection. To identify a set of features that relate to the linguistic choices of authors, we rely on different syntactic expressions of the same content. After identifying the relevant features (which we call *syntactic elements of expression*), we rely on patterns in the use of these features to recognize paraphrases of works.

In the absence of real-life plagiarism data, in this paper, we use a corpus of parallel translations of novels as surrogate for plagiarism data. Translations of *titles*, i.e., original works, into English by different people provide us with *books* that are paraphrases of the same content. We use these paraphrases to automatically identify:

1. Titles even when they are paraphrased, and
2. Pairs of book chapters that are paraphrases of each other.

Our first experiment shows that syntactic elements of expression outperform all baselines in recognizing titles even when they are paraphrased, providing a way of recognizing copies of works based on the similarities in their expression of content. Our second experiment shows that similarity measurements based on the combination of tfidf-weighted keywords and syntactic elements of expression outperform the weighted keywords in recognizing pairs of book chapters that are paraphrases of each other.

2 Related Work

We define expression as "the linguistic choices of authors in presenting a particular content" (Uzuner, 2005; Uzuner and Katz, 2005). Linguistic similarity between works has been studied in the text classification literature for identifying the style of an author. However, it is important to differentiate expression from style. Style refers to the *linguistic elements that, independently of content, persist over the works* of an author and has been widely studied in authorship attribution. Expression involves the *linguistic elements that relate to how an author phrases particular content* and can be used to identify potential copyright infringement or plagiarism. Similarities in the expression of similar content in two different works signal potential copying. We hypothesize that syntax plays a role in capturing expression of content. Our approach to recognizing paraphrased works is based on phrase structure of sentences in general, and structure of verb phrases in particular.

Most approaches to similarity detection use computationally cheap but linguistically less informed features (Peng and Hengartner, 2002; Sichel, 1974; Williams, 1975) such as keywords, function words, word lengths, and sentence lengths; approaches that include deeper linguistic information, such as syntactic information, usually incur significant computational costs (Uzuner et al., 2004). Our approach identifies useful linguistic information without incurring the computational cost of full text parsing; it uses context-free grammars to perform high-level syntactic analysis of part-of-speech tagged text (Brill, 1992). It turns out that such a level of analysis is sufficient to capture syntactic information related to creative aspects of writing; this in turn helps improve recognition of paraphrased documents. The results presented here show that extraction of useful linguistic information for text classification purposes does not have to be computationally prohibitively expensive, and that despite the tradeoff between the accuracy of features and computational efficiency, we can extract linguistically-informed features without full parsing.

3 Identifying Creative Aspects of Writing

In this paper, we first identify linguistic elements of expression and then study patterns in the use of these elements to recognize a work even when it is paraphrased. Translated literary works provide examples of linguistic elements that differ in expression but convey similar content. These works provide insight into the linguistic elements that capture expression. For example, consider the following semantically equivalent excerpts from three different translations of *Madame Bovary* by Gustave Flaubert.

> Excerpt 1: "Now Emma would often take it into her head to write him during the day. Through her window she would signal to Justin, and he would whip off his apron and fly to la huchette. And when Rodolphe arrived in response to her summons, it was to hear that she was miserable, that her husband was odious, that her life was a torment." (Translated by Unknown1.)

> Excerpt 2: “Often, even in the middle of the day, Emma suddenly wrote to him, then from the window made a sign to Justin, who, taking his apron off, quickly ran to la huchette. Rodolphe would come; she had sent for him to tell him that she was bored, that her husband was odious, her life frightful.” (Translated by Aveling.)

> Excerpt 3: “Often, in the middle of the day, Emma would take up a pen and write to him. Then she would beckon across to Justin, who would off with his apron in an instant and fly away with the letter to la huchette. And Rodolphe would come. She wanted to tell him that life was a burden to her, that she could not endure her husband and that things were unbearable.” (Translated by Unknown2.)

Inspired by syntactic differences displayed in such parallel translations, we identified a novel set of syntactic features that relate to how people convey content.

3.1 Syntactic Elements of Expression

We hypothesize that given particular content, authors choose from a set of semantically equivalent syntactic constructs to express this content. To paraphrase a work without changing content, people try to interchange semantically equivalent syntactic constructs; patterns in the use of various syntactic constructs can be sufficient to indicate copying.

Our observations of the particular expressive choices of authors in a corpus of parallel translations led us to define syntactic elements of expression in terms of sentence-initial and -final phrase structures, semantic classes and argument structures of verb phrases, and syntactic classes of verb phrases.

3.1.1 Sentence-initial and -final phrase structures

The order of phrases in a sentence can shift the emphasis of a sentence, can attract attention to particular pieces of information and can be used as an expressive tool.

1 (a) Martha can finally put some money in the bank.
(b) Martha can put some money in the bank, finally.
(c) Finally, Martha can put some money in the bank.

2 (a) Martha put some money in the bank on Friday.
(b) On Friday, Martha put some money in the bank.
(c) Some money *is what* Martha put in the bank on Friday.
(d) In the bank *is where* Martha put some money on Friday.

The result of such expressive changes affect the distributions of various phrase types in sentence-initial and -final positions; studying these distributions can help us capture some elements of expression. Despite its inability to detect the structural changes that do not affect the sentence-initial and -final phrase types, this approach captures some of the phrase-level expressive differences between semantically equivalent content; it also captures different sentential structures, including question constructs, imperatives, and coordinating and subordinating conjuncts.

3.1.2 Semantic Classes of Verbs

Levin (1993) observed that verbs that exhibit similar syntactic behavior are also related semantically. Based on this observation, she sorted 3024 verbs into 49 high-level semantic classes. Verbs of “sending and carrying”, such as `convey`, `deliver`, `move`, `roll`, `bring`, `carry`, `shuttle`, and `wire`, for example, are collected under this semantic class and can be further broken down into five semantically coherent lower-level classes which include “drive verbs”, “carry verbs”, “bring and take verbs”, “slide verbs”, and “send verbs”. Each of these lower-level classes represents a group of verbs that have similarities both in semantics and in syntactic behavior, i.e., they can grammatically undergo similar syntactic alternations. For example, “send verbs” can be seen in the following alternations (Levin, 1993):

1. **Base Form**
   - Nora sent the book to Peter.
   - NP + V + NP + PP.
2. **Dative Alternation**
   - Nora sent Peter the book.
   - NP + V + NP + NP.

Semantics of verbs in general, and Levin’s verb classes in particular, have previously been used for evaluating content and genre similarity (Hatzivassiloglou et al., 1999). In addition, similar semantic classes of verbs were used in natural language processing applications: START was the first natural language question answering system to use such verb classes (Katz and Levin, 1988). We use

Levin's semantic verb classes to describe the expression of an author in a particular work. We assume that semantically similar verbs are often used in semantically similar syntactic alternations; we describe part of an author's expression in a particular work in terms of the semantic classes of verbs she uses and the particular argument structures, e.g., NP + V + NP + PP, she prefers for them. As many verbs belong to multiple semantic classes, to capture the dominant semantic verb classes in each document we credit all semantic classes of all observed verbs. We extract the argument structures from part of speech tagged text, using context-free grammars (Uzuner, 2005).

3.1.3 Syntactic Classes of Verbs

Levin's verb classes include exclusively "non-embedding verbs", i.e., verbs that do not take clausal arguments, and need to be supplemented by classes of "embedding verbs" that do take such arguments. Alexander and Kunz (1964) identified syntactic classes of embedding verbs, collected a comprehensive set of verbs for each class, and described the identified verb classes with formulae written in terms of phrasal and clausal elements, such as verb phrase heads (Vh), participial phrases (Partcp.), infinitive phrases (Inf.), indicative clauses (IS), and subjunctives (Subjunct.). We used 29 of the more frequent embedding verb classes and identified their distributions in different works. Examples of these verb classes are shown in Table 1. Further examples can be found in (Uzuner, 2005; Uzuner and Katz, 2005).

Syntactic Formula	Example
NP + Vh + NP + from + Partcp.	The belt kept him from dying.
NP + Vh + that + IS	He admitted that he was guilty.
NP + Vh + that + Subjunct.	I request that she go alone.
NP + Vh + to + Inf.	My father wanted to travel.
NP + Vh + wh + IS	He asked if they were alone.
NP + pass. + Partcp.	He was seen stealing.

Table 1: Sample syntactic formulae and examples of embedding verb classes.

We study the syntax of embedding verbs by identifying their syntactic class and the structure of their observed embedded arguments. After identifying syntactic and semantic characteristics of verb phrases, we combine these features to create further elements of expression, e.g., syntactic classes of embedding verbs and the classes of semantic non-embedding verbs they co-occur with.

4 Evaluation

We tested sentence-initial and -final phrase structures, semantic and syntactic classes of verbs, and structure of verb arguments, i.e., syntactic elements of expression, in paraphrase recognition and in plagiarism detection in two ways:

- Recognizing titles even when they are paraphrased, and
- Recognizing pairs of book chapters that are paraphrases of each other.

For our experiments, we split books into chapters, extracted all relevant features from each chapter, and normalized them by the length of the chapter.

4.1 Recognizing Titles

Frequently, people paraphrase parts of rather than complete works. For example, they may paraphrase chapters or paragraphs from a work rather than the whole work. We tested the effectiveness of our features on recognizing paraphrased components of works by focusing on chapter-level excerpts (smaller components than chapters have very sparse vectors given our sentence-level features and will be the foci of future research) and using boosted decision trees (Witten and Frank, 2000).

Our goal was to recognize chapters from the *titles* in our corpus even when some *titles* were paraphrased into multiple *books*; in this context, titles are original works and paraphrased books are translations of these titles. For this, we assumed the existence of one legitimate book from each title. We used this book to train a model that captured the syntactic elements of expression used in this title. We used the remaining paraphrases of the title (i.e., the remaining books paraphrasing the title) as the test set—these paraphrases are considered to be plagiarized copies and should be identified as such given the model for the title.

4.1.1 Data

Real life plagiarism data is difficult to obtain. However, English translations of foreign titles exist and can be obtained relatively easily. Titles that have been translated on different occasions by different translators and that have multiple translations provide us with examples of books that paraphrase the same content and serve as our surrogate for plagiarism data.

To evaluate syntactic elements of expression on recognizing paraphrased chapters from titles, we compared the performance of these features with tfidf-weighted keywords on a 45-way classification task. The corpus used for this experiment included 49 books from 45 titles. Of the 45 titles, 3 were paraphrased into a total of 7 books (3 books paraphrased the title *Madame Bovary*, 2 books paraphrased *20000 Leagues*, and 2 books paraphrased *The Kreutzer Sonata*). The remaining titles included works from J. Austen (1775-1817), C. Dickens (1812-1870), F. Dostoyevski (1821-1881), A. Doyle (1859-1887), G. Eliot (1819-1880), G. Flaubert (1821-1880), T. Hardy (1840-1928), V. Hugo (1802-1885), W. Irving (1789-1859), J. London (1876-1916), W. M. Thackeray (1811-1863), L. Tolstoy (1828-1910), I. Turgenev (1818-1883), M. Twain (1835-1910), and J. Verne (1828-1905).

4.1.2 Baseline Features

The task described in this section focuses on recognizing paraphrases of works based on the way they are written. Given the focus of authorship attribution literature on “the way people write”, to evaluate the syntactic elements of expression on recognizing paraphrased chapters of a work, we compared these features against features frequently used in authorship attribution as well as features used in content recognition.

Tfidf-weighted Keywords: Keywords, i.e., content words, are frequently used in content-based text classification and constitute one of our baselines.

Function Words: In studies of authorship attribution, many researchers have taken advantage of the differences in the way authors use function words (Mosteller and Wallace, 1963; Peng and Hengartner, 2002). In our studies, we used a set of 506 function words (Uzuner, 2005).

Distributions of Word Lengths and Sentence Lengths: Distributions of word lengths and sentence lengths have been used in the literature for authorship attribution (Mendenhall, 1887; Williams, 1975; Holmes, 1994). We include these features in our sets of baselines along with information about means and standard deviations of sentence lengths (Holmes, 1994).

Baseline Linguistic Features: Sets of surface, syntactic, and semantic features have been found to be useful for authorship attribution and have been adopted here as baseline features. These features included: the number of words and the number of sentences in the document; type–token ratio; average and standard deviation of the lengths of words (in characters) and of the lengths of sentences (in words) in the document; frequencies of declarative sentences, interrogatives, imperatives, and fragmental sentences; frequencies of active voice sentences, be-passives and get-passives; frequencies of ’s-genitives, of-genitives and of phrases that lack genitives; frequency of overt negations, e.g., “not”, “no”, etc.; and frequency of uncertainty markers, e.g., “could”, “possibly”, etc.

4.1.3 Experiment

To recognize chapters from the *titles* in our corpus even when some *titles* were paraphrased into multiple *books*, we randomly selected 40–50 chapters from each title. We used 60% of the selected chapters from each title for training and the remaining 40% for testing. For paraphrased titles, we selected training chapters from one of the paraphrases and testing chapters from the remaining paraphrases. We repeated this experiment three times; at each round, a different paraphrase was chosen for training and the rest were used for testing.

Our results show that, on average, syntactic elements of expression accurately recognized components of titles 73% of the time and significantly outperformed all baselines[3] (see middle column in Table 2).[4]

[3] The tfidf-weighted keywords used in this experiment do not include proper nouns. These words are unique to each title and can be easily replaced without changing content or expression in order to trick a plagiarism detection system that would rely on proper nouns.

[4] For the corpora used in this paper, a difference of 4% or more is statistically significant with $\alpha = 0.05$.

Feature Set	Avg. accuracy (complete corpus)	Avg. accuracy (para-phrases) only
Syntactic elements of expression	73%	95%
Function words	53%	34%
Tfidf-weighted keywords	47%	38%
Baseline linguistic	40%	67%
Dist. of word length	18%	54%
Dist. of sentence length	12%	17%

Table 2: Classification results (on the test set) for recognizing titles in the corpus even when some titles are paraphrased (middle column) and classification results only on the paraphrased titles (right column). In either case, random chance would recognize a paraphrased title 2% of the time.

The right column in Table 2 shows that the syntactic elements of expression accurately recognized on average 95% of the chapters taken from paraphrased titles. This finding implies that some of our elements of expression are common to books that are derived from the same title. This commonality could be due to the similarity of their content or due to the underlying expression of the original author.

4.2 Recognizing Pairs of Paraphrased Chapters

Experiments in Section 4.1 show that we can use syntactic elements of expression to recognize titles and their components based on the way they are written even when some works are paraphrased. In this section, our goal is to identify pairs of chapters that paraphrase the same content, i.e., chapter 1 of translation 1 of *Madame Bovary* and chapter 1 of translation 2 of *Madame Bovary*. For this evaluation, we used a similar approach to that presented by Nahnsen et al. (2005).

4.2.1 Data

Our data for this experiment included 47 chapters from each of two translations of *20000 Leagues under the Sea* (Verne), 35 chapters from each of 3 translations of *Madame Bovary* (Flaubert), 28 chapters from each of two translations of *The Kreutzer Sonata* (Tolstoy), and 365 chapters from each of 2 translations of *War and Peace* (Tolstoy). Pairing up the chapters from these titles provided us with more than 1,000,000 chapter pairs, of which approximately 1080 were paraphrases of each other.[5]

4.2.2 Experiment

For experiments on finding pairwise matches, we used similarity of vectors of tfidf-weighted keywords;[6] and the multiplicative combination of the similarity of vectors of tfidf-weighed keywords of works with the similarity of vectors of syntactic elements of expression of these works. We used cosine to evaluate the similarity of the vectors of works. We omitted the remaining baseline features from this experiment—they are features that are common to majority of the chapters from each book, they do not relate to the task of finding pairs of chapters that could be paraphrases of each other.

We ranked all chapter pairs in the corpus based on their similarity. From this ranked list, we identified the top *n* most similar pairs and predicted that they are paraphrases of each other. We evaluated our methods with precision, recall, and f-measure.[7]

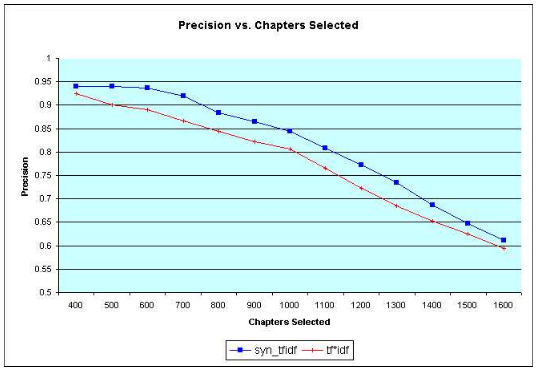


Figure 1: Precision.

Figures 1, 2, and 3 show that syntactic elements of expression improve the performance of tfidf-weighted keywords in recognizing pairs of paraphrased chapters significantly in terms of precision, recall, and f-measure for all *n*; in all of these figures, the blue line marked *syn_tfidf* represents the performance of tfidf-weighted keywords enhanced with

[5] Note that this number double-counts the paraphrased pairs; however, this fact is immaterial for our discussion.

[6] In this experiment, proper nouns are included in the weighted keywords.

[7] The ground truth marks only the same chapter from two different translations of the same title as similar, i.e., chapter x of translation 1 of *Madame Bovary* and chapter y of translation 2 of *Madame Bovary* are similar only when $x = y$.

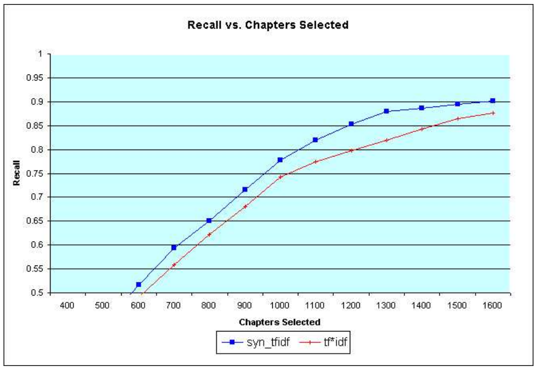


Figure 2: Recall.

syntactic elements of expression. More specifically, the peak f-measure for tfidf-weighted keywords is approximately 0.77 without contribution from syntactic elements of expression. Adding information about similarity of syntactic features to cosine similarity of tfidf-weighted keywords boosts peak f-measure value to approximately 0.82.[8] Although the f-measure of both representations degrade when $n > 1100$, this degradation is an artifact of the evaluation metric: the corpus includes only 1080 similar pairs, at $n > 1100$, recall is very close to 1, and therefore increasing *n* hurts overall performance.

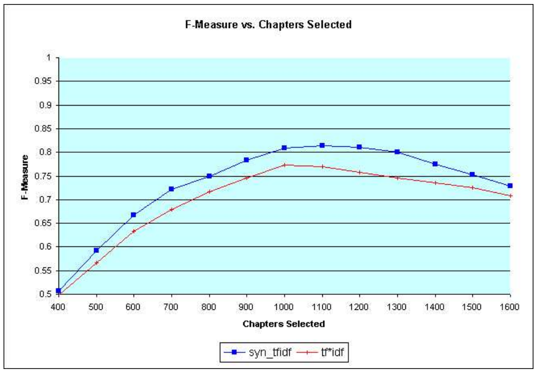


Figure 3: F-measure.

5 Conclusion

Plagiarism is a problem at all levels of education. Increased availability of digital versions of works makes it easier to plagiarize others' work and the large volumes of information available on the web makes it difficult to identify cases of plagiarism. To identify plagiarism even when works are paraphrased, we propose studying the use of particular syntactic constructs as well as keywords in documents.

This paper shows that syntactic information can help recognize works based on the way they are written. Syntactic elements of expression that focus on the changes in the phrase structure of works help identify paraphrased components of a title. The same features help improve identification of pairs of chapters that are paraphrases of each other, despite the content these chapters share with the rest of the chapters taken from the same title. The results presented in this paper are based on experiments that use translated novels as surrogate for plagiarism data. Our future work will extend our study to real life plagiarism data.

6 Acknowledgements

The authors would like to thank Sue Felshin for her insightful comments. This work is supported in part by the Advanced Research and Development Activity as part of the AQUAINT research program.

References

D. Alexander and W. J. Kunz. 1964. Some classes of verbs in English. In *Linguistics Research Project*. Indiana University, June.

E. Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*.

V. Hatzivassiloglou, J. Klavans, and E. Eskin. 1999. Detecting similarity by applying learning over indicators. In *Proceedings of the 37th Annual Meeting of the ACL*.

D. I. Holmes. 1994. Authorship attribution. *Computers and the Humanities*, 28.

B. Katz and B. Levin. 1988. Exploiting lexical regularities in designing natural language systems. In *Proceedings of the 12th Int'l Conference on Computational Linguistics (COLING '88)*.

B. Levin. 1993. *English Verb Classes and Alternations. A Preliminary Investigation*. University of Chicago Press.

T. C. Mendenhall. 1887. Characteristic curves of composition. *Science*, 11.

[8] The difference is statistically significant at $\alpha = 0.05$.

F. Mosteller and D. L. Wallace. 1963. Inference in an authorship problem. *Journal of the American Statistical Association*, 58(302).

T. Nahnsen, Ö. Uzuner, and B. Katz. 2005. Lexical chains and sliding locality windows in content-based text similarity detection. *CSAIL Memo*, AIM-2005-017.

R. D. Peng and H. Hengartner. 2002. Quantitative analysis of literary styles. *The American Statistician*, 56(3).

H. S. Sichel. 1974. On a distribution representing sentence-length in written prose. *Journal of the Royal Statistical Society (A)*, 137.

Ö. Uzuner and B. Katz. 2005. Capturing expression using linguistic information. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*.

Ö. Uzuner, R. Davis, and B. Katz. 2004. Using empirical methods for evaluating expression and content similarity. In *Proceedings of the 37th Hawaiian International Conference on System Sciences (HICSS-37)*. IEEE Computer Society.

Ö. Uzuner. 2005. *Identifying Expression Fingerprints Using Linguistic Information*. Ph.D. thesis, Massachusetts Institute of Technology.

C. B. Williams. 1975. Mendenhall's studies of word-length distribution in the works of Shakespeare and Bacon. *Biometrika*, 62(1).

I. H. Witten and E. Frank. 2000. *Data Mining: Practical Machine Learning Tools with Java Implementations*. Morgan Kaufmann, San Francisco.

Towards a Prototyping Tool for Behavior Oriented Authoring of Conversational Agents for Educational Applications

Gahgene Gweon, Jaime Arguello, Carol Pai, Regan Carey, Zachary Zaiss, Carolyn Rosé

Human-Computer Interaction Institute/ Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213 USA
Ggweon,jarguell,cpai,rcarey,zzaiss,cp3a@andrew.cmu.edu

Abstract

Our goal is to develop tools for facilitating the authoring of conversational agents for educational applications, and in particular to enable non-computational linguists to accomplish this task efficiently. Such a tool would benefit both learning researchers, allowing them to study dialogue in new ways, and educational technology researchers, allowing them to quickly build dialogue based help systems for tutoring systems. We argue in favor of a user-centered design methodology. We present our *work-in-progress* design for authoring, which is motivated by our previous tool development experiences and preliminary contextual interviews and then refined through user testing and iterative design.

1 Introduction

This paper reports work in progress towards developing TuTalk, an authoring environment developed with the long term goal of enabling the authoring of effective tutorial dialogue agents. It was designed for developers without expertise in knowledge representation, artificial intelligence, or computational linguistics. In our previous work we have reported progress towards the development of authoring tools specifically focusing on robust language understanding capabilities (Rosé et al., 2003; Rosé & Hall, 2004; Rosé, et al., 2005). In this paper, we explore issues related to authoring both at the dialogue and sentence level, as well as the interaction between these two levels of authoring. Some preliminary work on the underlying architecture is reported in (Jordan, Rosé, & VanLehn, 2001; Aleven & Rosé, 2004; Rosé & Torrey, 2004). In this paper we focus on the problem of making this computational linguistics technology accessible to our target user population.

We are developing the TuTalk authoring environment in connection with a number of existing local research projects related to educational technology in general and tutorial dialogue in particular. It is being developed primarily for use within the Pittsburgh Sciences of Learning Center (PSLC) data shop, which includes development efforts for a suite of authoring tools to be used for building the infrastructure for 7 different computer enhanced courses designated as LearnLab courses. These LearnLab courses, which are conducted within local secondary schools as well as universities, and which include Chinese, French, English as a Second Language, Physics, Algebra, Geometry, and Chemistry, involve heavy use of technology both for the purpose of supporting learning as well as for the purpose of conducting learning research in a classroom setting. Other local projects related to calculus and thermodynamics

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 45–52, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

tutoring also have plans to use TuTalk. In this paper we will discuss specifically how we have used corpora related to ESL, physics, thermodynamics, and calculus in our development effort.

To support this multi-domain effort, it is essential that the technology we develop be domain independent and usable by a non-technical user population, or at least a user population not possessing expertise in knowledge representation, artificial intelligence, or computational linguistics. Thus, we are employing a corpus based methodology that bootstraps domain specific authoring using examples of desired conversational behavior for the domain.

2 A Historical Perspective

While a focus on design based on standards and practices from human-computer interaction community have not received a great deal of attention in previously published tool development efforts known to the computational linguistics community, our experience tells us that insufficient attention to these details leads to the development of tools that are unusable, particularly to the user population that we target with our work.

Some desiderata related to the design of our system are obvious based on our target user population. Currently, many educational technology oriented research groups do not have computational linguists on their staff with the expertise required to author domain specific knowledge sources for use with sophisticated state-of-the-art understanding systems, such as CARMEL (Rosé, 2000) or TRIPS (Allen et al., 2001). However, previous studies have shown that, while scaffolding and guidance is required to support the authoring process, non-computational linguists possess many of the basic skills required to author conversational interfaces (Rosé, Pai, & Arguello, 2005). Because the main barrier of entry to such sophisticated tools are expertise in understanding the underlying data structures and linguistically motivated representation, our tools should have an interface that masks the unnecessary details and provides intuitive widgets that manipulate the data in ways that are consistent with the mental models the users bring with them to the authoring process. In order to be maximally accessible to developers of educational technology, the system should involve minimal programming.

The design of Carmel-Tools (Rosé et al., 2003; Rosé & Hall, 2004), the first generation of our authoring tools, was based on these obvious desiderata and not on any in-depth analysis of data collected from our target user population. While an evaluation of the underlying computational linguistics technology showed promise (Rosé & Hall, 2004), the results from actual authoring use were tremendously disappointing.

A formal study reported in (Rosé, et al., 2005) demonstrates that even individuals with expertise in computational linguistics have difficulty predicting the coverage of knowledge sources that would be generated automatically from example texts annotated with desired representations. Informal user studies involving actual use of Carmel-Tools then showed that a consequence of this lack of ability is that authors were left without a clear strategy for moving through their corpus. As a result, time was lost from annotating examples that did not yield the maximum amount of new knowledge in the generated knowledge sources. Furthermore, since authors tended not to test the generated knowledge sources as they were annotating examples, errors were difficult for them to track later, despite facilities designed to help them with that task.

Another finding from our user studies was that although the interface prevented authors from violating the constraints they designed into their predicate language, it did not keep authors from annotating similar texts with very different representations, thus introducing a great deal of spurious ambiguity. Thus, they did not naturally maintain consistency in their application of their own designed meaning representation languages across example texts. An additional problem was that authors sometimes decomposed examples in ways that lead to overly general rules, which then lead to incorrect analyses when these rules matched inappropriate examples.

These disappointing results convinced us of the importance of taking a user-centered design approach to our authoring interface redesign process.

3 Preliminary Design Intents from Contextual Interviews

The core essence of the user-centered design approach is designing from data rather than from preconceived notions of what will be useful and what will work well. Expert blind spots often lead to designs based on intuitions that overlook needs or overly emphasize issues that are not centrally important (Koedinger & Nathan, 2004; Nathan & Koedinger, 2000). Contextual inquiry is used at an early stage in the user-centered design process to collect the foundational data on which to build a design (Beyer and Holtzbatt, 2000). Contextual Inquiry is a popular method developed within the Human Computer Interaction community where the design team gathers data from end users while watching what the users do in context of their work. Contextual interviews are used to illuminate these observations by engaging end-users in interviews in which they show specific instances within their work life that are relevant for the design process. These methods help define requirements as well as plan and prioritize important aspects of functionality. At the same time, the system designers get a chance to gain insights about the users' environment, tasks, cultural influences and difficulties in the current processes.

Many aspects of the Tutalk tool were designed based on contextual inquiry (CI) data. The design team conducted five CIs with users who have experience in using existing authoring tools such as Carmel-Tools (Rosé & Hall, 2004). The design team leader also spent one week observing novice tool users working with the current set of tools at an Intelligent Tutoring Summer School. Here we will discuss some findings from those CIs and observations and how they motivated some general design intents, which we flesh out later in the paper.

A common pattern we observed in our CIs was that having different floating windows for different tasks fills up the computer screen relatively quickly and confuses authors as to where they are in the process of authoring. The TuTalk design addresses this observed problem by anchoring the main window and switching only the components of the window as needed. A standard logic for layout and view switching helps authors know what to expect in different contexts. Placement of buttons in TuTalk is consistently near the textboxes that they control, and a bounding box is drawn around related sets of controls so that the user does not get lost trying to figure out where the buttons are or what they are for.

We observed that authors needed to refer to cheat sheets and user documentation to use their current tools effectively and that different users did not employ the same terminology to refer to similar functionality, which made communication difficult. Furthermore, their current suites of tools were not designed as one integrated environment. Thus, a lot of shuffling of files from one directory to another was required in order to complete the authoring process. Users without Unix operating system experience found this especially confusing. Our goal is to require only very minimal documentation that can be obtained on-line in the context of use.

TuTalk is a single, integrated environment that makes use of GUI widgets for actions rather then requiring any text-based commands or file system activity. In this way we hope to avoid requiring the users to use a manual or a "cheat-sheet" reference for the commands they forget. As is common practice, TuTalk also uses consistent labels throughout the interface to promote understandability and communication with tool developers as well as other dialogue system developers.

4 Exploring the User's Mental Model through User Studies

As an additional way of gaining insights into what sort of interface would make the process of authoring conversational interfaces accessible, we conducted a small, exploratory user study in which we examined how members of our target user population think about the structure of language.

Two groups of college-level participants with no deep linguistics training were asked to read three transcribed conversations about ordering from a menu at a restaurant from our English as a Second Language corpus. The three specific restaurant dialogues were chosen because of their breadth of topic coverage and richness in linguistic expression. Participants were asked to perform tasks with these dialogues to mimic three levels of conversational interface authoring:

Macro Organization Tasks (dialogue level)

Level 1. How authors understand, segment, and organize dialogue topics

Level 2. How authors generalize across dialogues as part of constructing a "model" script

Micro Organization Task (sentence level)

Level 3. How authors categorize and decompose sentences within these dialogues

The first group (Group A, five participants) was asked to perform *Macro Organization Tasks* before processing sentences for the *Micro Organization Tasks*. The second group (Group B, four participants) was asked to perform these sets of tasks in the opposite order.

Our findings for the *Macro Organization Tasks* showed that participants effectively broke down dialogues into segments that reflected intuitive breaks in the conversation. These topics were then organized into semantically related categories. Although participants were not explicitly instructed on how to organize the topics, every participant used spatial proximity as a representation for semantic relatedness. Another finding was the presence of primacy effects in the "model" restaurant scripts they were asked to construct. These scripts were heavily influenced by the first dialogue read. As a result, important topics that surfaced in the other two dialogues were omitted from the model scripts.

Furthermore, we found that participants in Group B took much longer in completing the *Micro Organization Task* (35-40 minutes as opposed to 25-30 minutes) without performing the *Macro Organization Tasks* first. In general, we found that participants clustered sentences based on surface characteristics rather than creating ontologically similar classes that would be more useful from a system development perspective. In a follow-up study we are exploring ways of guiding users to cluster sentences in ways that are more useful from a system building perspective.

Our preliminary findings show that getting an overall sense of the corpus facilitates micro-level organization. This is hindered by two factors: First, primacy effects interfere with macro-level comprehension. Second, system developers struggle to strategically select portions of their corpus on which to focus their initial efforts.

5 Stage One: Corpus Organization

While existing tools from our previous work required authors to organize their corpus data prior to their interaction with the tools, both our contextual research and user studies indicated that support for organizing corpus data prior to authoring is important.

In light of this concern, the TuTalk authoring process consists of three main stages. Corpus collection, corpus data organization through what we call the InfoMagnet interface, and authoring propper. First, a corpus is collected by asking users to engage in conversation using either a typed or spoken chat interface. In the case of spoken input, the speech is then transcribed into textual form. Second, the raw corpus data is automatically preprocessed for display and interactive organization using the InfoMagnet interface. As part of the preprocessing, dialogue protocols are segmented automatically at topic boundaries, which can be adjusted by hand later during authoring propper. The topic oriented segments are then clustered semi-automatically into topic based classes. The output from this stage is an XML file where dialogue segments are reassembled into their original dialogue contexts, with each utterance labeled by topic. This XML file is finally passed onto the authoring environment propper, which is then used for finer grained processing, such as shifting topic segment boundaries and labeling more detailed utterance functionality.

Our design is for knowledge sources that are runable from our dialogue system engine to be generated directly from the knowledge base created during the fine-grained authoring process as in Carmel-Tools (Rosé & Hall, 2004), however currently our focus is on iterative development of a prototype of the authoring interaction design. Thus, more work is required to create the final end-to-end implementation. In this section we focus on the design of the corpus collection and organization part of the authoring process.

5.1 Corpus Collection

An important part of our mission is developing technology that can use collected and automatically pre-processed corpus data to guide and streamline the authoring process. Prior to the arduous process of organizing and extracting meaningful data, a corpus must be collected.

As part of the PSLC and other local tutorial dialogue efforts we have collected corpus data from multiple domains that we have made use of in our development process. In particular, we have been working with data collected in connection with the PSLC Physics and English as a Second Language LearnLab courses as well as local Calculus and Thermodynamics tutoring projects. Currently we have physics tutoring data primarily from one physics tutor (interactions with 40 students), thermodynamics data from four different tutors (interactions with 27 students), Calculus data from four different tutors (84 dialogues), and ESL dialogues collected from 15 pairs of students (30 dialogues altogether).

While we have drawn upon data from all of these domains for testing the underlying language processing technology for our development effort, for our user studies we have so far mainly drawn upon our ESL corpus, which includes conversations between students about every-day tasks such as ordering from a restaurant or about their pets. We chose the language ESL data for our initial user tests because we expected it to be easy for a general population to relate to, but we plan to begin using calculus data as well.

5.2 InfoMagnets Interface

As mentioned previously, once the raw dialogue corpus is collected, the next step is to sift through this data and assign utterances (or groups of utterances) to classes conceptualized by the author. Clustering is a natural step in this kind of exploratory data analysis, as it promotes learning by grouping and generalizing from what we know about some of the objects in a cluster. For this purpose we have designed the InfoMagnets interface, which introduces a non-technical metaphor to the task of iterative document clustering. The InfoMagnets interface was designed to address the problems identified in the user study discussed above in Section 4. Specifically, we expected that those problems could be addressed with an interface that:

1. Divides dialogues into topic based segments and automatically clusters them into conceptually similar classes
2. Eliminates primacy effects of sequential dialogue consumption by creating an inclusive compilation of all dialogue topics
3. Makes the topic similarity of documents easily accessible to the user

The InfoMagnets interface is displayed in Figure 1. The larger circles (InfoMagnets) correspond to cluster centroids and the smaller ones (particles) correspond to actual spans of text. Lexical cohesion in the vector space translates into attraction in the InfoMagnet space. The attraction from each particle to each InfoMagnet is evident from the particle's position with respect to all InfoMagnets and its reaction-time when an InfoMagnet is moved by the user, which causes the documents that have some attraction with it to redistribute themselves in the InfoMagnet space.

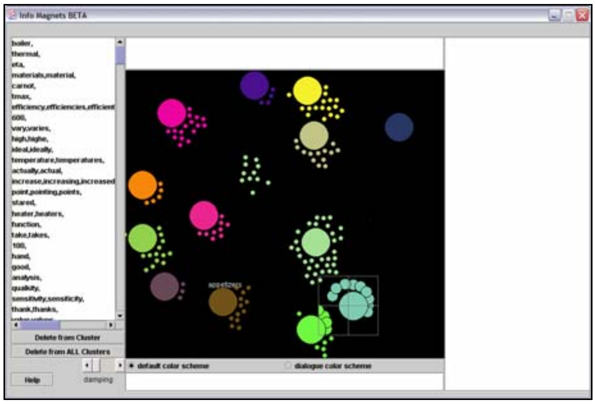


Figure 1 InfoMagnets Interface

Being an unsupervised learning method, clustering often requires human-intervention for fine-tuning (e.g. removing semantically-weak discriminators, culling meaningless clusters, or deleting/splitting clusters too fine/coarse for the author's purpose). The InfoMagnets interface provides all this functionality, while shielding the author from the computational details inherent in these tasks

Initially, the corpus is clustered using the Bisecting K-means Algorithm described in (Kumar et al., 1998). Although this is a hard clustering algorithm, the InfoMagnet interface shows the particles association with all clusters, given by the position of the particle. Using a cross-hair lens, the author is able to view the contents of each cluster centroid and each particle. The author is able to select a group of particles and view the common features between these particles and any InfoMagnet in the space. The interface allows the editing of InfoMagnets by adding and removing features, splitting InfoMagnets, and removing InfoMagnets. When the user edits an InfoMagnets, the effect in the particle distribution is shown immediately and in an animated way.

5.3 XML format

The data collected from the conversations in .txt format are reformatted into XML format before being displayed with InfoMagnet tool. The basic XML file contains a transcription of the conversational data and has the following structure: Under the top root tag, there is <dialogue> tag which designates the conversion about a topic. It has an "id" attribute so that we can keep track of each separate conversation. Then each sentence has a <sentence> tag with two attributes "uid" and "agent". "uid" is a universal id and "agent" tells who was speaking. Additionally, sentences are grouped into segments, marked off with a <subtopic> tag.

The user's interaction with the InfoMagnet interface adds a "subtopic-name" attribute to the subtopic tag. Then, the authoring interface proper, described below, allows for further adjustments and additions to the xml tags. The final knowledge sources will be generated from this XML based representation.

6 Authoring

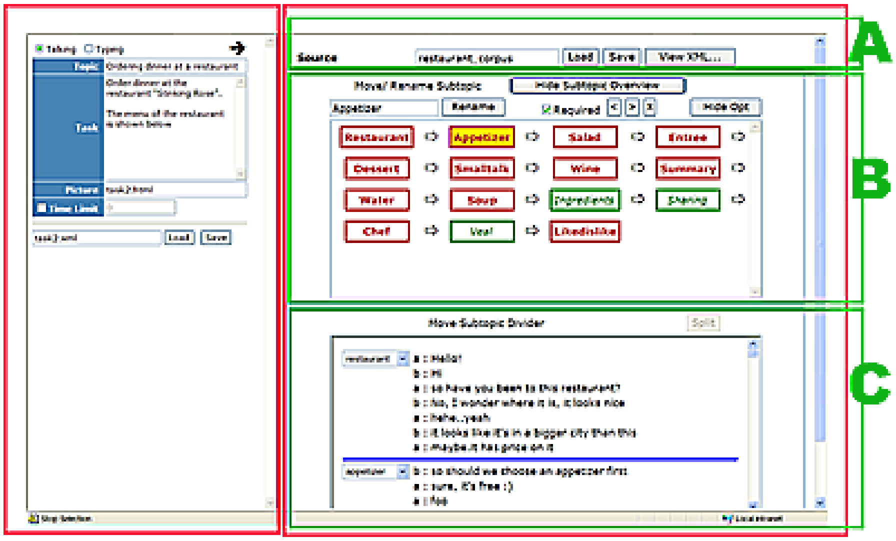
The authoring environment proper consists of two main views, namely the authoring view and tutoring view. The authoring view is where the author designs the behavior of the conversational agent. The authoring view has two levels; the topic level and the subtopic level. The tutoring view is what a student will be looking at when interacting with the conversational agent. Our focus here is on the Authoring view.

Authoring View: Topic Level

The Topic level of the authoring view allows for manipulating the relationship between subtopics as well as the definition of the subtopic. Figure 2 shows the topic level authoring view, which consists of two panels. In the left, the author inputs the description of the task that the student will engage in with the agent. The author can specify whether the student will be typing or talking, the title of the topic, the task description, an optional picture that aids with the task (such as a menu or a map of a city), and a time limit.

In the right panel of the topic level authoring view, the structure imposed on the data by interaction with the InfoMagnets interface is displayed in sequential form. The top section of the interface (figure 2, section A) has a textbox for specifying an xml file to read. The next section (figure 2, section B), "Move / Rename Subtopic" displays the subtopics. The order of the subtopics displayed in this section acts as a guideline for the agent to follow during the conversation. Double-clicking on a subtopic will display a subtopic view on the right panel. This view acts as a reference for the agent's conversation within the subtopic and is explained in the next section. The author can also rearrange the order of subtopics by selecting a subtopic and using the ">" and "<" buttons to move the subtopic right or left respectively. "x" is used to delete the subtopic. The author can also specify whether the discussion of a subtopic is required (displayed in red) or optional (in green) using the checkbox that is labeled "required". Clicking on the "Hide Opt" button will only display the required subtopics.

The last section of the right panel in topic level authoring view (figure 2, section C) is titled "move subtopic divider". A blue line denotes the border of the subtopic. The author can move the line up or down to move the boundary of the subtopics automatically inserted by the InfoMagnets interface. The author can also click on any part of conversation and press the "split" button to split the subtopic in two sections. In addition, she can change the label of the subtopic segment using the drop down list.



○ Talking ○ Typing

Topic: Ordering dinner at a restaurant

Task: Order dinner at the restaurant "Stinking Rose". The menu of the restaurant is shown below.

Picture: task1.html

☐ Time Limit

task1.xml

Source: restaurant_corpus

Move/ Rename Subtopic

Appetizer ☑ Required

Move Subtopic Divider

restaurant
a : Hello!
b : Hi
a : so have you been to this restaurant?
b : No, I wonder where it is, it looks nice
a : hehe..yeah
b : it looks like it's in a bigger city than this
a : maybe it has price on it

appetizer
b : so should we choose an appetizer first
a : sure, it's free :)
a : hee

Figure 2: Topic Level Authoring View

Authoring View: Subtopic Level

While the Topic View portion of the authoring interface proper allows specification of which subtopics can occur as part of a dialogue, which are required and which are optional, and what the default ordering is, the Subtopic Level is for specification of the low level turn-by-turn details of what happens within a subtopic segment. This section reports early work on the design of this portion of the interface.

The subtopic view displays a structure that the conversational agent refers to in deciding what its next contribution should be. The building blocks from which knowledge sources for the dialogue engine will be generated are templates abstracted from example dialogue segments, similar to KCD specifications (Jordan, Rosé, & VanLehn, 2001; Rosé & Torry, 2004). As part of the process of abstracting templates, each utterance is tagged with its utterance type using a menu-based interface as in (Gweon et al., submitted). The utterance type determines what would be an appropriate form for a response. Identifying this is meant to allow the dialogue manager to maintain coherence in the emerging dialogue. Users may also trim out undesired portions of text from the actual example fragments in abstracting out templates to be used for generating knowledge sources.

Each utterance type has sets of template response types associated with them. The full set of utterance types includes Open questions, Closed questions, Understanding check questions, Assertions, Commands/Requests, Acknowledgements, Acceptances, and Rejections. The templates will not be used in their authored form. Instead, they will be used to generate knowledge sources in the form required by the backend dialogue system as in (Rosé & Hall, 2004), although this is still work in progress. Each template is composed of one or more exchanges during which the speaker who initiated the segment maintains conversational control. If control shifts to the other speakers, a new template is used to guide the conversation. After each of the controlling speaker's turns within the segment are listed a number of prototypical responses. One of these responses is a default response that signals that the dialogue should proceed to the next turn in the template. The other prototypical responses are associated with subgoals that are in turn associated with other templates. Thus, the dialogue takes on a hierarchical structure.

Mixed initiative interaction is meant to emerge from the underlying template-based structure by means of the multi-threaded discourse management approach discussed in (Rosé & Torrey, 2004). To this end, templates are meant to be used in two ways. The first way is when the dialogue system has conversational control. In this case, conversations can be managed as in (Rosé et al., 2001). The second way in which templates are used is for determining how to respond when user's have conversational control. Provided that the user's utterances match what is expected of the conversational participant who is in control based on the current template, then the system can simply pick one of the expected responses. Otherwise if at some point the user's response does not match, the system should check whether the user is initiating yet a different segment. If not, then the system should take conversational control.

7 Future Plans

In this paper we have discussed our user research and design process to date for the development of TuTalk, an authoring environment for conversational agents for educational purposes. We are continuing our user research and design iteration with the plan of end-to-end system testing in actual use starting this summer.

Acknowledgements

This work was supported in part by Office of Naval Research, Cognitive and Neural Sciences Division Grant N00014-05-1-0043 and NSF Grant SBE0354420.

References

Aleven , V. and Rosé, C. P. 2004. Towards Easier Creation of Tutorial Dialogue Systems: Integration of Authoring Environments for Tutoring and Dialogue Systems, *Proceedings of the ITS Workshop on Tutorial Dialogue Systems*

Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., & Stent, A. 2000. An Architecture for a Generic Dialogue Shell. *NLENG: Natural Language Engineering*, Cambridge University Press, 6 (3), 1-16.

Beyer, H. & Holtzblatt, K. (1998). *Contextual Design*, Morgan Kaufmann Publishers.

Gweon, G., Rosé, C., Wittwer, J., Nueckles, M. (submitted). Supporting Efficient and Reliable Content Analysis with Automatic Text Processing Technology, Submitted to INTERACT '05.

Jordan, P., Rosé, C. P., & VanLehn, K. (2001). Tools for Authoring Tutorial Dialogue Knowledge. In J. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.), *Proceedings of AI-ED 2001* (pp. 222-233). Amsterdam, IOS Press.

Koedinger, K. R. & Nathan, M. J. (2004). The real story behind story problems: Effects of representations on quantitative reasoning. *The Journal of the Learning Sciences*, 13(2).

Nathan, M. J. & Koedinger, K. R. (2000). Moving beyond teachers' intuitive beliefs about algebra learning. *Mathematics Teacher*, 93, 218-223.

Porter, M. 1980. An Algorithm for Suffix Stripping, *Program* 14 {3}:130 – 137.

Robertson, S. and Walker, S., 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval *Proceedings of SIGIR-94.*

Rosé, C. P., and Torrey, C. (2004). ,DRESDEN: Towards a Trainable Tutorial Dialogue Manager to Support Negotiation Dialogues for Learning and Reflection, *Proceedings of the Intelligent Tutoring Systems Conference*.

Rosé, C. P. and Hall, B. (2004). A Little Goes a Long Way: Quick Authoring of Semantic Knowledge Sources for Interpretation, *Proceedings of SCaNaLu '04*.

Rosé, C. P. 2000. A framework for robust semantic interpretation. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 311–318.

Rosé, C. P., Pai, C., Arguello, J. 2005. Enabling Non-Linguists to Author Advanced Conversational Interfaces Easily. *Proceedings of FLAIRS 2005.*

Steinbach, Kepis, and Kumar, *A Comparison of Document Clustering Techniques*, pg. 8. http://lucene.apache.org

Direkt Profil: A System for Evaluating Texts of Second Language Learners of French Based on Developmental Sequences

Jonas Granfeldt[1] **Pierre Nugues**[2] **Emil Persson**[1] **Lisa Persson**[2]
Fabian Kostadinov[3] **Malin Ågren**[1] **Suzanne Schlyter**[1]

[1]Dept. of Romance Languages
Lund University
Box 201, 221 00 Lund, Sweden

[2]Dept. of Computer Science
Lund University
Box 118, 221 00 Lund, Sweden

[3]Dept. of Computer Science
University of Zurich
CH-8057 Zurich, Switzerland

{Jonas.Granfeldt, Malin.Agren, Suzanne.Schlyter}@rom.lu.se
emil.person@telia.com nossrespasil@hotmail.com
Pierre.Nugues@cs.lth.se fabian.kostadinov@access.unizh.ch

Abstract

Direkt Profil is an automatic analyzer of texts written in French as a second language. Its objective is to produce an evaluation of the developmental stage of students under the form of a grammatical learner profile. *Direkt Profil* carries out a sentence analysis based on developmental sequences, i.e. local morphosyntactic phenomena linked to a development in the acquisition of French.

The paper presents the corpus that we use to develop the system and briefly, the developmental sequences. Then, it describes the annotation that we have defined, the parser, and the user interface. We conclude by the results obtained so far: on the test corpus the systems obtains a recall of 83% and a precision of 83%.

1 Introduction

With few exceptions, systems for evaluating language proficiency and for Computer-Assisted Language Learning (CALL) do not use Natural Language Processing (NLP) techniques. Typically, existing commercial and non-commercial programs apply some sort of pattern-matching techniques to analyze texts. These techniques not only reduce the quality and the nature of the feedback but also limit the range of possible CALL applications.

In this paper, we present a system that implements an automatic analysis of texts freely written by learners. Research on Second Language Acquisition (SLA) has shown that writing your own text in a communicative and meaningful situation with a feedback and/or an evaluation of its quality and its form constitutes an excellent exercise to develop second language skills.

The aim of the program, called *Direkt Profil*, is to evaluate the linguistic level of the learners' texts in the shape of a learner profile. To analyze sentences, the program relies on previous research on second language development in French that itemized a number of specific constructions corresponding to *developmental sequences*.

2 The CEFLE Lund Corpus

For the development and the evaluation of the system, we used the CEFLE corpus (*Corpus Écrit de Français Langue Étrangère de Lund* "Lund Written Corpus of French as a Foreign Language"). This corpus currently contains approximately 100,000 words (Ågren, 2005). The texts are narratives of varied length and levels. We elicited them by asking 85 Swedish high-school students and 22 young French to write stories evoked by a sequence of images. Figure 1 shows pictures corresponding to one of them: *Le voyage en Italie* "The journey to Italy". The goal of the system being to analyze French as a foreign language, we used the texts of the French native speakers as control group.

The following narrative is an example from a be-

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 53–60, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

ginner learner:

> *Elles sont deux femmes. Elles sont a italie au une vacanse. Mais L'Auto est très petite. Elles va a Italie. Au l'hothel elles demande une chambre. Un homme a le clé. Le chambre est grande avec deux lies. Il fait chaud. C'est noir. Cette deux femmes est a une restaurang. Dans une bar cet deux hommes. Ils amour les femmes. Ils parlons dans la bar. Ils ont tres bien. Le homme et la femme participat a un sightseeing dans la Rome. Ils achetons une robe. La robe est verte. La femme et l'homme reste au un banqe. Ils c'est amour. La femme et l'homme est au une ristorante. es hommes va avec les femmes. L'auto est petite.*

This text contains a certain number of typical constructions for French as a foreign language: parataxis, very simple word order, absence of object pronouns, basic verb forms, agreement errors, spelling mistakes. Research on the acquisition of French as a foreign language has shown that these constructions (and others) appear in a certain systematic fashion according to the proficiency level of the learners. With *Direkt Profil*, we aim at detecting automatically these structures and gathering them so that they represent a grammatical learner profile. This learner profile can ultimately be used to assess learners' written production in French.

3 *Direkt Profil* and Previous Work

Direkt Profil is an analyzer of texts written in French as a foreign language. It is based on the linguistic constructions that are specific to developmental sequences. We created an annotation scheme to mark up these constructions and we used it to describe them systematically and detect them automatically. The analyzer parses the text of a learner, annotates the constructions, and counts the number of occurrences of each phenomenon. The result is a text profile based on these criteria and, possibly, an indication of the level of the text. A graphical user interface (GUI) shows the results to the user and visualizes by different colors the detected structures. It is important to stress that *Direkt Profil* is not a grammar checker.

The majority of the tools in the field can be described as writing assistants. They identify and sometimes correct spelling mistakes and grammatical errors. The line of programs leading to *PLNLP* (Jensen et al., 1993) and NLPWin (Heidorn, 2000) is one of the most notable achievements. The grammatical checker of *PLNLP* carries out a complete parse. It uses binary phrase-structure rules and takes into account some dependency relations. *PLNLP* is targeted primarily, but not exclusively, to users writing in their mother tongue. It was created for English and then applied to other languages, including French.

Other systems such as *FreeText* (Granger et al., 2001) and *Granska* (Bigert et al., 2005) are relevant to the CALL domain. *FreeText* is specifically designed to teach language and adopts a interactive approach. It uses phrase-structure rules for French. In case of parsing failure, it uses relaxed constraints to diagnose an error (agreement errors, for example). *Granska*, unlike *FreeText*, carries out a partial parsing. The authors justify this type of analysis by a robustness, which they consider superior and which makes it possible to accept more easily incorrect sentences.

4 An Analysis Based on Developmental Sequences

The current systems differ with regard to the type of analysis they carry out: complete or partial. The complete analysis of sentences and the correction of errors are difficult to apply to texts of learners with (very) low linguistic level since the number of unknown words and incorrect sentences are often extremely high.

We used a test corpus of 6,842 words to evaluate their counts. In the texts produced by learners at the lowest stage of development, Stage 1, nearly 100% of the sentences contained a grammatical error (98.9% were incorrect[1]) and 24.7% of the words were unknown.[2] At this stage of development, any complete analysis of the sentences seems very difficult to us. On the other hand, in the control group the

[1] An "incorrect sentence" was defined as a sentence containing at least one spelling, syntactic, morphological, or semantic error.

[2] An "unknown word" is a token that does not appear in the lexicon employed by the system (ABU CNAM, see below)

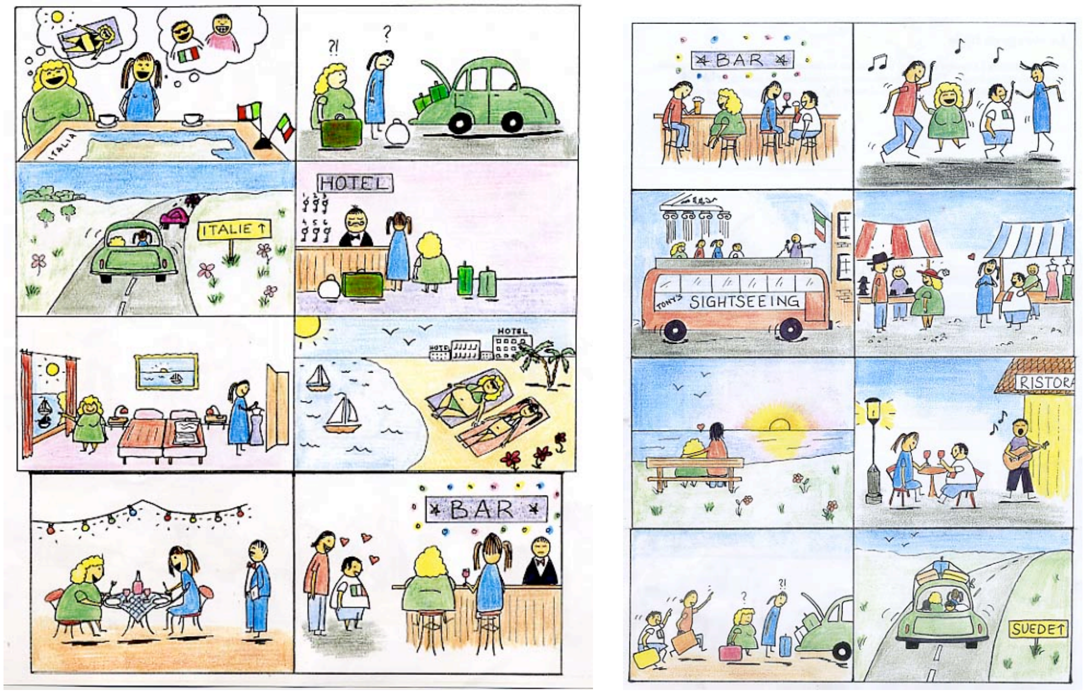


Figure 1: *Le voyage en Italie* “The journey to Italy”.

corresponding figures are 32.7% for incorrect sentences and 10.6% for unknown words. More importantly, this analysis shows that using a quantification of “unknown words” and “incorrect sentences” only is insufficient to define the linguistic level of learners’ texts. Learners at Stage 3 have in fact fewer incorrect sentences than learners from Stage 4 (70.5% vs. 80.2%). Moreover, the percentage of unknown words in the control group (the natives) is slightly higher than that of learners from the Stage 4 (10.6% vs. 10.4%). Thus, the simple count of errors is also insufficient to distinguish more advanced learners from natives. To identify properly and to define learners of various linguistic levels, we need more detailed analyses and more fine-grained measures. This is exactly the purpose of the developmental sequences and learner profiles implemented in *Direkt Profil*.

5 Developmental Sequences in French

Direkt Profil carries out an analysis of local phenomena related to a development in the acquisition of French. These phenomena are described under the form of developmental sequences. The sequences are the result of empirical observations stemming from large learner corpora of spoken language (Bartning and Schlyter, 2004). They show that certain grammatical constructions are acquired and can be produced in spontaneous spoken language in a fixed order. Clahsen and al. (1983) as well as Pienemann and Johnston, (1987) determined developmental sequences for German and spoken English. For spoken French, Schlyter (2003) and Bartning and Schlyter (2004) proposed 6 stages of development and developmental sequences covering more than 20 local phenomena. These morphosyntactic phenomena are described under the form of local structures inside the verbal or nominal domain. Table 1 shows a subset of these phenomena. It is a matter of current debate in field of SLA to what extent these developmental sequences are independent of the mother tongue.

The horizontal axis indicates the temporal development for a particular phenomenon: The developmental sequence. The vertical axis indicates the set of grammatical phenomena gathered in such way that they make up a “profile” or a stage of acquisition. To illustrate better how this works, we will compare the C (finite verb forms in finite contexts) and G (object pronouns) phenomena.

At Stage 1, the finite and infinitive forms coexist in finite contexts. As the main verb of the sentence, we find in the learners' production *je parle* (transcription of /je parl/ analyzed as a "finite form") as well as /je parle/ i.e. **je parler* or **je parlé*. The current estimation is that in Stage 1, there are between 50 and 75% of finite forms in finite contexts. At Stage 4, the percentage of finite forms has increased to 90–98%. For this morphological phenomenon, the developmental sequence describes a successive "morphologization".

The G phenomenon concerns the developmental sequence of object pronouns. The first object pronouns are placed in a postverbal position according to the scheme Subject-Verb-Object (SVO), e.g. **je vois le/la/lui* (instead of *je le/la vois*). At Stage 3, learners can produce phrases according to the SvOV scheme (Pronoun-Auxiliary-Object-Verb): *Je veux le voir* (correct) but also **j'ai le vu* (incorrect). At Stage 5, we observe *je l'ai vu*. For this syntactic phenomenon, the developmental sequence describes a change in the linear organization of the constituents.

6 Annotation

The concept of group, either noun group or verb group, correct or not, represents the essential grammatical support of our annotation. The majority of syntactic annotation standards for French takes such groups into account in one way or another. Gendner et al. (2004) is an example that reconciles a great number of annotations. These standards are however insufficient to mark up all the constructions in Table 1.

We defined a text annotation specific to *Direkt Profil* based on the inventory of the linguistic phenomena described by Bartning and Schlyter (2004) (Table 1). We represented these phenomena by decision trees whose final nodes correspond to a category of analysis.

The annotation uses the XML format and annotates the texts using 4 layers. Only the 3^{rd} layer is really grammatical:

- The first layer corresponds to the segmentation of the text in words.
- The second layer annotates prefabricated expressions or sentences (e.g. *je m'appelle*). These structures correspond to linguistic expressions learned "by heart" in a holistic fashion. It has been shown that they have a great importance in the first years of learning French.
- The third layer corresponds to a chunk annotation of the text, restricted to the phenomena to identify. This layer marks up simultaneously each word with its part-of-speech and the verb and noun groups to which they belong. The verb group incorporates subject clitic pronouns. The XML element `span` marks the groups and features an attribute to indicate their class in the table. The `tag` element annotates the words with attributes to indicate the lemma, the part-of-speech, and the grammatical features. The verb group in the sentence *Ils parlons dans la bar* extracted from the learner text above is annotated as: `<span class="p1_t1_c5131"><tag pos="pro:nom:pl:p3:mas">Ils</tag> <tag pos="ver:impre:pl:p1"> parlons </tag></span> dans la bar`. The class denoted `p1_t1_c5131` corresponds to a "finite lexical verb, no agreement".
- The fourth layer counts structures typical of an acquisition stage. It uses the `counter` XML element, `<counter id="counter.2" counter_name="passe_compose" rule_id="participe_4b" value="1"/>`.

7 Implementation

The running version of *Direkt Profil* is restricted to the analysis of the verb groups and clitic pronouns. For each category in Table 1, the program identifies the corresponding constructions in a text and counts them.

The analyzer uses manually written rules and a lexicon of inflected terms. The variety of the constructions contained in the corpus is large and in order not to multiply the number of rules, we chose a constraint reinforcement approach. Conceptually, the analyzer seeks classes of phrase structures in which all the features are removed. It gradually identifies the structures while varying the feature

Ph.	**Stages**	**1**	**2**	**3**	**4**	**5**	**6**
A.	% of sentences containing a verb (in a conversation)	20–40%	30–40%	50%	60%	70%	75%
B.	% of lexical verbs showing +/-finite opposition (types)	No opp.; % in finite forms 1–3sg	10–20% of types in opposition	About 50% in opposition	Most in opposition	All in opposition	+
C.	% of finite forms of lexical verbs in obligatory contexts (occurrences)	Finite forms 50%–75%	Finite forms 70–80%	Finite forms: 80–90%	Finite forms: 90–98%	Finite forms: 100%	+
D.	1^{st}, 2^{nd}, 3^{rd} pers. sing. (copula/aux) *est, a, va*	No opposition: *J'ai/ c'est*	Opposition *j'ai – il a je suis – il est*	Isolated errors **je va, *je a*	+	+	+
E.	% of 1^{st} pers. plural S-V agreement *nous* V-*ons* (occurrences)	–	70–80%	80–95%	Errors in complex constructions	+	+
F.	3^{rd} pers. plural S-V agreement with *viennent, veulent, prennent*	–	– *ils *prend*	Isolated cases of agreement	50% of cases with agreement	Some problems remain	+
G.	Object pronouns (placement)	–	SVO	S(v)oV	SovV appears	Productive	+ (*y, en)*
H.	% of gender agreement Article-Noun (occurrences)	55–75%	60–80%	65–85%	70–90%	75–95%	90–100%

Table 1: Developmental sequences adapted from Schlyter (2003); Bartning and Schlyter (2004).
Legend: – = no occurrences; + = acquired at a native-like level; aux = auxiliary; pers. = person; S-V = Subject-Verb

values. The recognition of the group boundaries is done by a set of closed-class words and heuristics inside the rules. It thus follows an old but robust strategy used in particular by Vergne (1999), *inter alia*, for French.

Direkt Profil applies a cascade of three sets of rules to produce the four annotation layers. The first unit segments the text in words. An intermediate unit identifies the prefabricated expressions. The third unit annotates simultaneously the parts-of-speech and the groups. Finally, the engine creates a group of results and connects them to a profile. It should be noted that the engine neither annotates all the words, nor all segments. It considers only those which are relevant for the determination of the stage. The engine applies the rules from left to right then from right to left to solve certain problems of agreement.

The rules represent partial structures and are divided into a condition part and an action part. The condition part contains the search parameters. It can be a lemma, a regular expression, or a class of inflection. The engine goes through the text and applies the rules using a decision tree. It tests the condition part to identify the sequences of contiguous words. Each rule produces a positive ("match") or negative ("no match") result. The rules are applied according to the result of the condition part and annotate the text, count the number of occurrences of the phenomenon, and connect to another rule. By traversing the nodes of the tree, the engine memorizes the rules it has passed as well as the results of the condition parts of these rules. When arriving at a final node, the engine applies the action parts of all the rules.

The engine finds the words in a dictionary of inflected terms. It does not correct the spelling mistakes except for the accents and certain stems. Learners frequently build erroneous past participles inferring a wrong generalization of stems. An example is the word **prendu* (taken) formed on the stem *prend*|*re* and of the suffix -u.

We used a lexicon available from the Association des Bibliophiles Universels' web site (http://abu.cnam.fr/) that we corrected and transposed into XML. We also enriched it with verb stems.

8 Interface

Direkt Profil merges the annotation levels in a result object. This object represents the original text, the annotation, the trace of the rule application, and the counters. The result object, which can be saved, is then transformed by the program to be presented to the user. The display uses the XHTML 1.1 specifications which can be read by any Web browser. *Direkt Profil* has a client-server architecture where the server carries out the annotation of a text and the client collects the text with an input form and interacts with the user.

Figure 2 shows a screenshot of *Direkt Profil*'s GUI displaying the analysis of the learner text above. The interface indicates to the user by different colors all the structures that the analyzer detected.

9 Results and Evaluation

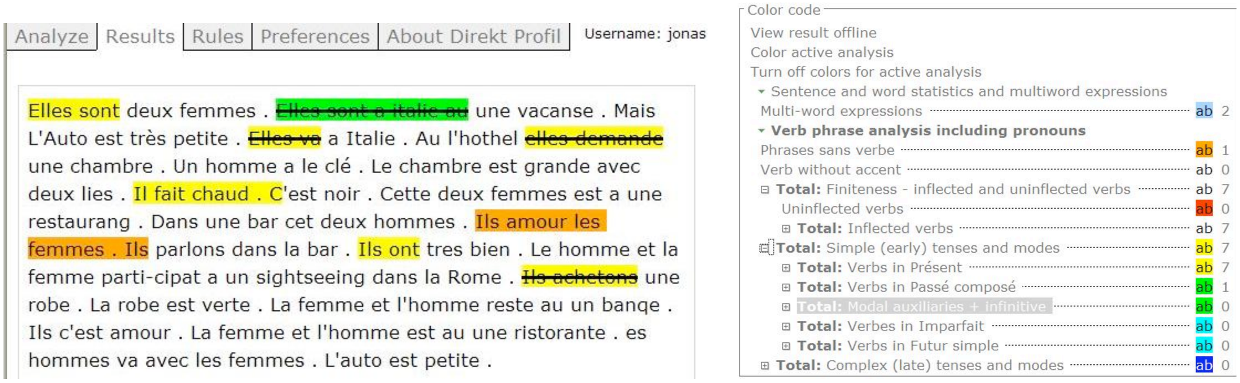
We evaluated *Direkt Profil* with a subset of the CEFLE corpus. We chose 20 texts randomly distributed on 4 learner stages. We also used 5 texts coming from the control group. In this version, we did not test the correction of the misspelled words: accent and stems. Table 2 shows some statistics on the size of the texts and Table 3 shows the results in the form of recall and precision.

The results show that *Direkt Profil* detects well the desired phenomena. It reveals also interesting differences according to the levels of the texts. The results show that *Direkt Profil* analyzes better the learner texts than the texts from the native French adolescents (control group). Without knowing exactly why, we note that it suggests that the adopted strategy, which aims at analyzing texts in French as a foreign language, seems promising.

10 Conclusion and Future Work

We presented a system carrying out a machine analysis of texts based on developmental sequences. The goal is to produce a learner profile. We built a parser and developed a set of rules to annotate the texts. *Direkt Profil* is integrated in a client-server architecture and has an interface allowing the interaction with the user.

The results show that it is possible to describe the vast majority of the local structures defined by the



Elles sont deux femmes . ~~Elles sont a italie au~~ une vacanse . Mais L'Auto est très petite . ~~Elles va~~ a Italie . Au l'hothel ~~elles demande~~ une chambre . Un homme a le clé . Le chambre est grande avec deux lies . Il fait chaud . C'est noir . Cette deux femmes est a une restaurang . Dans une bar cet deux hommes . Ils amour les femmes . Ils parlons dans la bar . Ils ont tres bien . Le homme et la femme parti-cipat a un sightseeing dans la Rome . ~~Ils achetons~~ une robe . La robe est verte . La femme et l'homme reste au un banqe . Ils c'est amour . La femme et l'homme est au une ristorante . es hommes va avec les femmes . L'auto est petite .

Color code

View result offline
Color active analysis
Turn off colors for active analysis

- Sentence and word statistics and multiword expressions
  - Multi-word expressions ab 2
- **Verb phrase analysis including pronouns**
  - Phrases sans verbe ab 1
  - Verb without accent ab 0
  - ⊟ **Total:** Finiteness - inflected and uninflected verbs ab 7
    - Uninflected verbs ab 0
    - ⊞ **Total:** Inflected verbs ab 7
  - ⊟ **Total:** Simple (early) tenses and modes ab 7
    - ⊞ **Total:** Verbs in Présent ab 7
    - ⊞ **Total:** Verbs in Passé composé ab 1
    - ⊞ **Total:** Modal auxiliaries + infinitive ab 0
    - ⊞ **Total:** Verbes in Imparfait ab 0
    - ⊞ **Total:** Verbs in Futur simple ab 0
  - ⊞ **Total:** Complex (late) tenses and modes ab 0

Figure 2: The graphical user interface.

	Stage 1	Stage 2	Stage 3	Stage 4	Control	Total
Number of analyzed texts	5	5	5	5	5	25
Word count	740	1233	1571	1672	1626	6842
Sentence count	85	155	166	126	107	639
Average text length (in words)	148	247	314	334	325	274
Average length of sentences (in words)	8.7	7.9	9.5	13.3	15.2	10.9

Table 2: Test corpus.

	Stage 1	Stage 2	Stage 3	Stage 4	Control	Total
Reference structures	23	97	101	119	85	425
Detected structures	27	98	100	112	92	429
Correctly detected structures	15	81	89	96	73	354
Non detected structures	5	16	12	20	11	()64
Overdetected structures	10	17	11	17	19	()74
Recall	65%	84%	88%	81%	86%	83%
Precision	56%	83%	89%	86%	79%	83%
F-measure	0.6	0.83	0.89	0.83	0.82	0.83

Table 3: Results.

developmental sequences under the form of rules. *Direkt Profil* can then detect them and automatically analyze them. We can thus check the validity of the acquisition criteria.

In the future, we intend to test *Direkt Profil* in teaching contexts to analyze and specify, in an automatic way, the grammatical level of a learner. The program could be used by teachers to assess student texts as well as by the students themselves as a self-assessment and as a part of their learning process.

A preliminary version of *Direkt Profil* is available on line from this address `http://www.rom.lu.se:8080/profil`

References

Malin Ågren. 2005. Le marquage morphologique du nombre dans la phrase nominale. une étude sur l'acquisition du français L2 écrit. Technical report, Institut d'études romanes de Lund. Lund University.

Inge Bartning and Suzanne Schlyter. 2004. Stades et itinéraires acquisitionnels des apprenants suédophones en français l2. *Journal of French Language Studies*, 14(3):281–299.

Johnny Bigert, Viggo Kann, Ola Knutsson, and Jonas Sjöbergh. 2005. Grammar checking for Swedish second language learners. In *CALL for the Nordic Languages*, Copenhagen Studies in Language, pages 33–47. Copenhagen Business School, Samfundslitteratur.

Harald Clahsen, Jürgen M. Meisel, and Manfred Pienemann. 1983. *Deutsch als Fremdsprache. Der Spracherwerb ausländischer Arbeiter*. Narr, Tübingen.

Véronique Gendner, Anne Vilnat, Laura Monceaux, Patrick Paroubek, and Isabelle Robba. 2004. Les annotations syntaxiques de référence peas. Technical report, LIMSI, Orsay. http://www.limsi.fr/Recherche/CORVAL/easy/PEAS_reference_annotations_v1.6.html.

Sylviane Granger, Anne Vandeventer, and Marie-Josée Hamel. 2001. Analyse de corpus d'apprenants pour l'ELAO basé sur le TAL. *Traitement automatique des langues*, 42(2):609–621.

George E. Heidorn. 2000. Intelligent writing assistance. In Robert Dale, Hermann Moisl, and Harold Somers, editors, *Handbook of Natural Language Processing*. Marcel Dekker.

Karen Jensen, George E. Heidorn, and Stephen D. Richardson. 1993. *Natural Language Processing: The PLNLP Approach*. Kluwer Academic Publishers.

Manfred Pienemann and Malcolm Johnston. 1987. Factors influencing the development of second language proficiency. In David Nunan, editor, *Applying second language acquisition research*, pages 45–141. National Curriculum Resource Centre, Adelaide.

Suzanne Schlyter. 2003. Stades de développement en français L2. Technical report, Institut d'études romanes de Lund, Lund University. http://www.rom.lu.se/durs/STADES_DE_DEVELOPPEMENT_EN_FRANCAIS_L2.PDF.

Jacques Vergne. 1999. *Étude et modélisation de la syntaxe des langues à l'aide de l'ordinateur. Analyse syntaxique automatique non combinatoire. Synthèse et Résultats*. Habilitation à diriger des recherches, Université de Caen, 29 septembre.

Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions

Eiichiro SUMITA
Spoken Language Communication
Research Laboratories
ATR
Kyoto 619-0288 Japan
eiichiro.sumita@atr.jp

Fumiaki SUGAYA
Text Information Processing Laboratory
KDDI R&D Laboratories Inc.
Saitama 356-8502 Japan
fsugaya@kddilabs.jp

Seiichi Yamamoto
Department of Information Systems
Design
Doshisha University
Kyoto 610-0321 Japan
seyamamo@mail.doshisha.ac.jp
&
Spoken Language Communication
Research Laboratories
ATR

Abstract

This paper proposes the *automatic* generation of *Fill-in-the-Blank Questions* (FBQs) together with testing based on *Item Response Theory* (IRT) to measure English proficiency. First, the proposal generates an FBQ from a given sentence in English. The position of a blank in the sentence is determined, and the word at that position is considered as the correct choice. The candidates for incorrect choices for the blank are hypothesized through a thesaurus. Then, each of the candidates is verified by using the Web. Finally, the blanked sentence, the correct choice and the incorrect choices surviving the verification are together laid out to form the FBQ. Second, the proficiency of non-native speakers who took the test consisting of such FBQs is estimated through IRT.

Our experimental results suggest that: (1) the generated questions plus IRT estimate the non-native speakers' English proficiency; (2) while on the other hand, the test can be completed almost perfectly by English native speakers; and (3) the number of questions can be reduced by using *item information* in IRT.

The proposed method provides teachers and testers with a tool that reduces time and expenditure for testing English proficiency.

1 Introduction

English has spread so widely that 1,500 million people, about a quarter of the world's population, speak it, though at most about 400 million speak it as their native language (Crystal, 2003). Thus, English education for non-native speakers both now and in the near future is of great importance.

The progress of computer technology is advancing an electronic tool for language learning called *Computer-Assisted Language Learning* (CALL) and for language testing called *Computer-Based Testing* (CBT) or *Computer-Adaptive Testing* (CAT). However, no computerized support for producing a test, a collection of questions for evaluating *language proficiency*, has emerged to date. [*]

Fill-in-the-Blank Questions (FBQs) are widely used from the classroom level to far larger scales to measure peoples' proficiency at English as a second language. Examples of such tests include TOEFL (Test Of English as a Foreign Language, http://www.ets.org/toefl/) and TOEIC (Test Of English for International Communication, http://www.ets.org/toeic/).

A test comprising FBQs has merits in that (1) it is easy for test-takers to input answers, (2) computers can mark them, thus marking is invariable and objective, and (3) they are suitable for the modern testing theory, *Item Response Theory* (IRT).

Because it is regarded that writing incorrect choices that distract only the non-proficient test-taker is a highly skilled business (Alderson, 1996), FBQs have been written by human experts. Thus, test construction is time-consuming and expensive. As a result, utilizing up-to-date texts for question writing is not practical, nor is tuning in to individual students.

[*] See the detailed discussion in Section 6.

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 61–68, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

To solve the problems of time and expenditure, this paper proposes a method for generating FBQs using a corpus, a thesaurus, and the Web. Experiments have shown that the proficiency estimated through IRT with generated FBQs highly correlates with non-native speakers' real proficiency. This system not only provides us with a quick and inexpensive testing method, but it also features the following advantages:

(I) It provides "anyone" individually with up-to-date and interesting questions for self-teaching. We have implemented a program that downloads any Web page such as a news site and generates questions from it.

(II) It also enables on-demand testing at "anytime and anyplace." We have implemented a system that operates on a mobile phone. Questions are generated and pooled in the server, and upon a user's request, questions are downloaded. CAT (Wainer, 2000) is then conducted on the phone. The system for mobile phone is scheduled to be deployed in May of 2005 in Japan.

The remainder of this paper is organized as follows. Section 2 introduces a method for making FBQ, Section 3 explains how to estimate test-takers' proficiency, and Section 4 presents the experiments that demonstrate the effectiveness of the proposal. Section 5 provides some discussion, and Section 6 explains the differences between our proposal and related work, followed by concluding remarks.

2 Question Generation Method

We will review an FBQ, and then explain our method for producing it.

2.1 Fill-in-the-Blank Question (FBQ)

FBQs are the one of the most popular types of questions in testing. Figure 1 shows a typical sample consisting of a <u>partially blanked English sentence</u> and <u>four choices for filling the blank</u>. The tester ordinarily assumes that <u>exactly one choice is correct</u> (in this case, b)) and <u>the other three choices are incorrect</u>. The latter are often called *distracters*, because they fulfill a role to distract the less proficient test-takers.

Question 1 (FBQ)
I only have to _______ my head above water one more week.
a) reserve b) keep c) guarantee d) promise

N.B. the correct choice is b) keep.

Figure 1: A sample Fill-in-the-Blank Question (FBQ)

2.2 Flow of generation

Using question 1 above, the outline of generation is presented below (Figure 2).

A *seed* sentence (in this case, "I only have to keep my head above water one more week.") is input from the designated source, e.g., a corpus or a Web page such as well-known news site. *

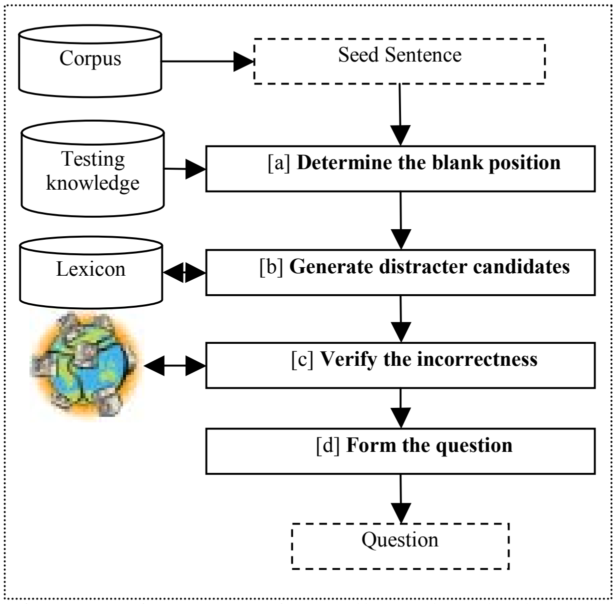

Corpus
Seed Sentence
Testing knowledge
[a] Determine the blank position
Lexicon
[b] Generate distracter candidates
[c] Verify the incorrectness
[d] Form the question
Question

Figure 2: Flow generating *Fill-In-The-Blank Question* (FBQ)

[a] The *seed* sentence is a correct English sentence that is decomposed into a sentence with a blank (*blanked sentence*) and the correct choice for the blank. After the seed

* Selection of the seed sentence (source text) is an important open problem because the *difficulty* of the seed (text) should influence the difficulty of the generated question. As for text difficulty, several measures such as Lexile by MetaMetrics (http://www.Lexile.com) have been proposed. They are known as *readability* and are usually defined as a function of sentence length and word frequency.

In this paper, we used corpora of business and travel conversations, because TOEIC itself is oriented toward business and daily conversation.

sentence is analyzed morphologically by a computer, according to the testing knowledge[*] the blank position of the sentence is determined. In this paper's experiment, the *verb* of the seed is selected, and we obtain the *blanked sentence* "I only have to ______ my head above water one more week." and the correct choice "keep."

[b] To be a good distracter, the candidates must maintain the grammatical characteristics of the correct choice, and these should be similar in *meaning*[†]. Using a *thesaurus*[‡], words similar to the correct choice are listed up as candidates, e.g., "clear," "guarantee," "promise," "reserve," and "share" for the above "keep."

[c] Verify (see Section 2.3 for details) *the incorrectness of the sentence* restored by each candidate, and if it is *not incorrect* (in this case, "clear" and "share"), the candidate is given up.

[d] If a sufficient number (in this paper, three) of candidates remain, form a question by randomizing the order of all the choices ("keep," "guarantee," "promise," and "reserve"); otherwise, another seed sentence is input and restart from step [a].

2.3 Incorrectness Verification

In FBQs, by definition, (1) the *blanked sentence* restored with the correct choice is *correct*, and (2) the *blanked sentence* restored with the distracter must be *incorrect*.

In order to generate an FBQ, the *incorrectness* of the sentence restored by each distracter candidate must be verified and if the combination is *not incorrect*, the candidate is rejected.

Zero-Hit Sentence

The Web includes all manners of language data in vast quantities, which are for everyone easy to access through a networked computer. Recently, exploitation of the Web for various natural language applications is rising (Grefenstette, 1999; Turney, 2001; Kilgarriff and Grefenstette, 2003; Tonoike et al., 2004).

We also propose a Web-based approach. We dare to assume that if there is a sentence on the Web, that sentence is considered *correct*; otherwise, the sentence is unlikely to be *correct* in that there is no sentence written on the Web despite the variety and quantity of data on it.

Figure 3 illustrates verification based on the retrieval from the Web. Here, *s* (*x*) is the blanked sentence, *s* (*w*) denotes the sentence restored by the word *w*, and *hits* (*y*) represents the number of documents retrieved from the Web for the key *y*.

Blanked sentence:

s (*x*)= "I only have to ____ my head above water one more week. "

Hits of incorrect choice candidates:

hits (*s* ("clear")) = 176 ; correct

hits (*s* ("guarantee")) = 0 ; incorrect

hits (*s* ("promise")) = 0 ; incorrect

hits (*s* ("reserve")) = 0 ; incorrect

hits (*s* ("share")) = 3 ; correct

Figure 3: Incorrectness and Hits on the Web

If *hits* (*s* (*w*)), is *small*, then the sentence restored with the word *w* is unlikely, thus the word *w* should be a good distracter. If *hits* (*s* (*w*)), is *large* then the sentence restored with the word *w* is likely, then the word *w* is *unlikely* to be a good distracter and is given up.

[*] Testing knowledge tells us what part of the seed sentence should be blanked. For example, we selected the *verb* of the seed because it is one of the basic types of blanked words in popular FBQs such as in TOEIC.

This can be a word of another *POS* (Part-Of-Speech). For this, we can use knowledge in the field of second-language education. Previous studies on errors in English usage by Japanese native speakers such as (Izumi and Isahara, 2004) unveiled patterns of errors specific to Japanese, e.g., (1) *article* selection error, which results from the fact there are no articles in Japanese; (2) *preposition* selection error, which results from the fact some Japanese counterparts have broader meaning; (3) *adjective* selection error, which results from mismatch of meaning between Japanese words and their counterpart. Such knowledge may generate questions harder for Japanese who study English.

[†] There are various aspects other than *meaning*, for example, *spelling*, *pronunciation*, and *translation* and so on. Depending on the aspect, lexical information sources other than a thesaurus should be consulted.

[‡] We used an in-house English thesaurus whose hierarchy is based on one of the off-the-shelf thesauruses for Japanese, called Ruigo-Shin-Jiten (Ohno and Hamanishi, 1984). In the above examples, the original word "keep" expresses two different concepts: (1) *possession-or-disposal*, which is shared by the words "clear" and "share," and (2) *promise*, which is shared by the words "guarantee," "promise," and "reserve." Since this depends on the thesaurus used, some may sense a slight discomfort at these concepts. If a different thesaurus is used, the distracter candidates may differ.

We used the **strongest** condition. If *hits* (*s* (*w*)) is ***zero***, then the sentence restored with the word *w* is unlikely, thus the word *w* should be a good distracter. If *hits* (*s* (*w*)), is ***not zero***, then the sentence restored with the word *w* is likely, thus the word *w* is *unlikely* to be a good distracter and is given up.

Retrieval NOT By Sentence

It is often the case that *retrieval by sentence* does not work. Instead of a sentence, a sequence of words around a blank position, beginning with a content word (or sentence head) and ending with a content word (or sentence tail) is passed to a search engine automatically. For the abovementioned sample, the sequence of words passed to the engine is "I only have to *clear* my head" and so on.

Web Search

We can use any search engine, though we have been using Google since February 2004. At that point in time, Google covered an enormous four billion pages.

The "correct" hits may come from non-native speakers' websites and contain invalid language usage. To increase reliability, we could restrict Google searches to Websites with URLs based in English-speaking countries, although we have not done so yet. There is another concern: even if sentence fragments cannot be located on the Web, it does not necessarily mean they are illegitimate. Thus, the proposed verification based on the Web is not perfect; the point, however, is that with such limitations, the generated questions are useful for estimating proficiency as demonstrated in a later section.

Setting aside the convenience provided by the off-the-shelf search engine, another search specialized for this application is possible, although the current implementation is fast enough to automate generation of FBQs, and the demand to accelerate the search is not strong. Rather, the problem of time needed for test construction has been reduced by our proposal.

The throughput depends on the text from which a seed sentence comes and the network traffic when the Web is accessed. Empirically, one FBQ is obtained in 20 seconds on average and the total number of FBQs in a day adds up to over 4,000 on a single computer.

3 Estimating Proficiency

3.1 Item Response Theory (IRT)

Item Response Theory (IRT) is the basis of modern language tests such as TOEIC, and enables *Computerized Adaptive Testing* (CAT). Here, we briefly introduce IRT. IRT, in which a question is called an *item*, calculates the test-takers' proficiency based on the answers for items of the given test (Embretson, 2000).

The basic idea is the *item response function*, which relates the probability of test-takers answering particular items correctly to their proficiency. The item response functions are modeled as *logistic curves* making an S-shape, which take the form (1) for item *i*.

$$P_i(\theta) = \frac{1}{1+\exp(-a_i(\theta - b_i))} \quad (1)$$

The *test-taker parameter*, θ, shows the proficiency of the test-taker, with higher values indicating higher performance.

Each of the *item parameters*, a_i and b_i, controls the shape of the item response function. The *a* parameter, called *discrimination*, indexes how steeply the item response function rises. The *b* parameter is called *difficulty*. Difficult items feature larger *b* values and the item response functions are shifted to the right. These item parameters are usually estimated by a maximal likelihood method. For computations including the estimation, there are many commercial programs such as BILOG (http://www.assess.com/) available.

3.2 Reducing test size by selection of effective items

It is important to estimate the proficiency of the test-taker by using as few items as possible. For this, we have proposed a method based on *item information*.

Expression (2) is the *item information* of item *i* at θ_j, the proficiency of the test-taker *j*, which indicates how much *measurement discrimination* an item provides.

The procedure is as follows.

1. Initialize *I* by the set of all generated FBQs.

2. According to Equation (3), we select the item whose contribution to *test information* is maximal.
3. We eliminate the selected item from *I* according to Equation (4).
4. If *I* is empty, we obtain the ordered list of effective items; otherwise, go back to step 2.

$$I_i(\theta_j) = a_i^{\,2} P_i(\theta_j)(1 - P_i(\theta_j)) \quad (2)$$

$$\hat{i} = \arg\max_i \left(\sum_j \sum_{i \in I} I_i\left(\theta_j\right) \right) \quad (3)$$

$$I = I - \hat{i} \quad (4)$$

4 Experiment

The FBQs for the experiment were generated in February of 2004. Seed sentences were obtained from ATR's corpus (Kikui *et al*., 2003) of the *business* and *travel* domains. The vocabulary of the corpus comprises about 30,000 words. Sentences are relatively short, with the average length being 6.47 words. For each domain 5,000 questions were generated automatically and each question consists of an English sentence with *one* blank and *four* choices.

4.1 Experiment with non-native speakers

We used the TOEIC score as the experiment's proficiency measure, and collected 100 Japanese subjects whose TOEIC scores were scattered from 400 to less than 900. The actual range for TOEIC scores is 10 to 990. Our subjects covered the dominant portion* of test-takers for TOEIC in Japan, excluding the highest and lowest extremes.†

We had the subjects answer 320 randomly selected questions from the 10,000 mentioned above. The raw marks were as follows: the average‡ mark was 235.2 (73.5%); the highest mark was 290 (90.6%); and the lowest was 158 (49.4%). This suggests that our FBQs are sensitive to test-takers' proficiency. In Figure 4, the y-axis represents estimated proficiency according to IRT (Section 3.1) and generated questions, while the x-axis is the real TOEIC score of each subject.

As the graph illustrates, the IRT-estimated proficiency (*θ*) and real TOEIC scores of subjects correlate highly with a co-efficiency of about 80%.

For comparison we refer to CASEC (http://casec.evidus.com/), an off-the-shelf test consisting of human-made questions and IRT. Its co-efficiency with real TOEIC scores is reported to be 86%.

This means the proposed automatically generated questions are promising for measuring English proficiency, achieving a nearly competitive level with human-made questions but with a few reservations: (1) whether the difference of 6% is large depends on the standpoint of possible users; (2) as for the number of questions to be answered, our proposal uses 320 questions in the experiments, while TOEIC uses 200 questions and CASEC uses only about 60 questions; (3) the proposed method uses FBQs only whereas CASEC and TOEIC use various types of questions.

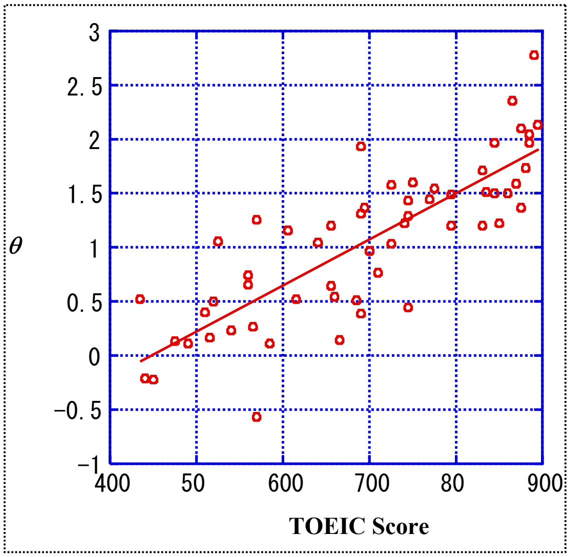


Figure 4: IRT-Estimated Proficiency (θ) vs. Real TOEIC Score

4.2 Experiment with a native speaker

To examine the quality of the generated questions, we asked a single subject§ who is a native speaker of English to answer 4,000 questions (Table 1).

The native speaker largely agreed with our generation, determining correct choices (type I). The

* Over 70% of all test-takers are covered (http://www.toeic.or.jp/toeic/data/data02.html).

† We have covered only the range of TOEIC scores from 400 to 900 due to expense of the experiment. In this restricted experiment, we do not claim that our proficiency estimation method covers the full range of TOEIC scores.

‡ The standard deviation was 29.8 (9.3%).

§ Please note that the analysis is based on a single native-speaker, thus we need further analysis by multiple subjects.

rate was 93.50%, better than 90.6%, the highest mark among the non-native speakers.

We present the problematic cases here.

- Type II is caused by the seed sentence being *incorrect* for the native speaker, and a distracter is bad because it is *correct*. Or like type III, it consists of ambiguous choices.
- Type III is caused by some generated distracters being *correct*; therefore, the choices are ambiguous.
- Type IV is caused by the seed sentence being *incorrect* and the generated distracters also being *incorrect*; therefore, the question cannot be answered.
- Type V is caused by the seed sentence being nonsense to the native speaker; the question, therefore, cannot be answered.

Table 1 Responses of a Native speaker

Type	***Explanation***		***Count***	***%***
I	**Single Selection**	**Match**	**3,740**	**93.50**
II		No match	55	1.38
III	No Selection	Ambiguous Choices	70	1.75
IV		No Correct Choice	45	1.13
V		Nonsense	90	2.25

Cases with bad seed sentences (portions of II, IV, and V) require cleaning of the corpus by a native speaker, and cases with bad distracters (portions of II and III) require refinement of the proposed generation algorithm.

Since the questions produced by this method can be flawed in ways which make them unanswerable even by native speakers (about 6.5% of the time) due to the above-mentioned reasons, it is difficult to use this method for high-stakes testing applications although it is useful for estimating proficiency as explained in the previous section.

4.3 *Proficiency θ* estimated with the reduced test and its relation to TOEIC Scores

Figure 5 shows the relationship between reduction of the test size according to the method explained in Section 3.2 and the estimated proficiency based on the reduced test. The x-axis represents the size of the reduced test in number of items, while the y-axis represents the correlation coefficient (R) between estimated proficiency and real TOEIC score.

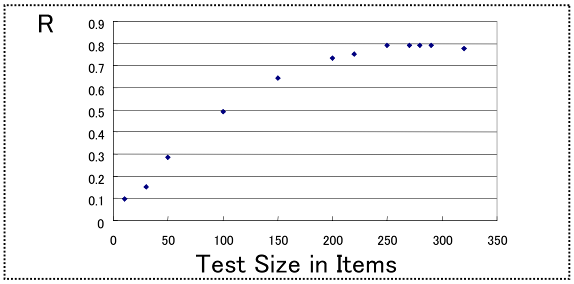


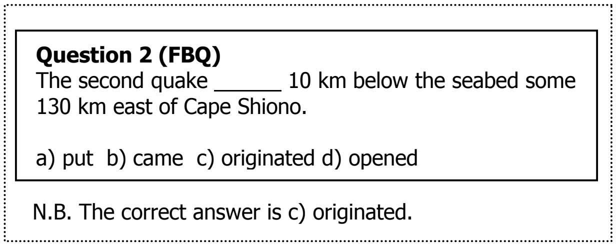
Figure 5 Correlation coefficient and Test size

5 Discussion

This section explains the on-demand generation of FBQs according to individual preference, an immediate extension and a limitation of our proposed method, and finally touches on free-format Q&A.

5.1 Effects of Automatic FBQ Construction

The method provides teachers and testers with a tool that reduces time and expenditure. Furthermore, the method can deal with any text. For example, up-to-date and interesting materials such as news articles of the day can be a source of seed sentences (Figure 6 is a sample generated from an article (http://www.japantimes.co.jp/) on an earthquake that occurred in Japan), which enables realization of a *personalized* learning environment.



Question 2 (FBQ)
The second quake ______ 10 km below the seabed some 130 km east of Cape Shiono.

a) put b) came c) originated d) opened

N.B. The correct answer is c) originated.

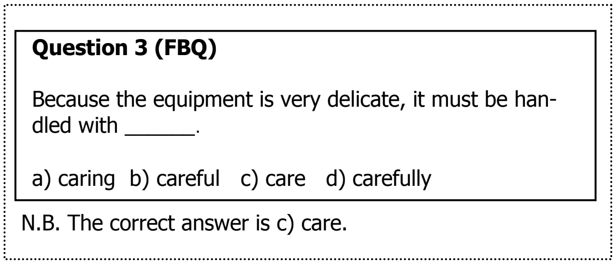
Figure 6: On-demand construction – a sample question from a Web news article in *The Japan Times* on "an earthquake"

We have generated questions from over 100 documents on various genres such as novels, speeches, academic papers and so on found in the enormous collection of e-Books provided by Project Gutenberg (http://www.gutenberg.org/).

5.2 A Variation of Fill-in-the-Blank Questions for Grammar Checking

In Section 2.2, we mentioned a constraint that a good distracter should maintain the grammatical characteristics of the correct choice originating in the seed sentence. The question checks not the grammaticality but the semantic/pragmatic correctness.

We can generate another type of FBQ by slightly modifying step [b] of the procedure in Section 2.2 to retain the stem of the original word *w* and vary the surface form of the word *w*. This modified procedure generates a question that checks the grammatical ability of the test takers. Figure 7 shows a sample of this kind of question taken from a TOEIC-test textbook (Educational Testing Service, 2002).



Question 3 (FBQ)

Because the equipment is very delicate, it must be handled with ______.

a) caring b) careful c) care d) carefully

N.B. The correct answer is c) care.

Figure 7: A variation on fill-in-the-blank questions

5.3 Limitation of the Addressed FBQs

The questions dealt with in this paper concern testing reading ability, but these questions are not suitable for testing listening ability because they are presented visually and cannot be pronounced. To test listening ability, like in TOIEC, other types of questions should be used, and automated generation of them is yet to be developed.

5.4 Free-Format Q&A

Besides measuring one's ability to receive information in a foreign language, which has been addressed so far in this paper, it is important to measure a person's ability to transmit information in a foreign language. For that purpose, tests for translating, writing, or speaking in a free format have been actively studied by many researchers (Shermis, 2003; Yasuda, 2004).

6 Related Work[*]

Here, we explain other studies on the generation of multiple-choice questions for language learning. There are a few previous studies on computer-based generation such as Mitkov (2003) and Wilson (1997).

6.1 Cloze Test

A computer can generate questions by deleting words or parts of words randomly or at every *N*-th word from text. Test-takers are requested to restore the word that has been deleted. This is called a "cloze test." The effectiveness of a "cloze test" or its derivatives is a matter of controversy among researchers of language testing such as Brown (1993) and Alderson (1996).

6.2 Tests on Facts

Mitkov (2003) proposed a computer-aided procedure for generating *multiple-choice questions* from textbooks. The differences from our proposal are that (1) Mitkov's method generates questions not about *language usage* but about *facts explicitly stated in a text*[†]; (2) Mitkov uses techniques such as term extraction, parsing, transformation of trees, which are different from our proposal; and (3) Mitkov does not use IRT while we use it.

7 Conclusion

This paper proposed the automatic construction of *Fill-in-the-Blank Questions* (FBQs). The proposed method generates FBQs using a corpus, a thesaurus, and the Web. The generated questions and *Item Response Theory* (IRT) then estimate second-language proficiency.

Experiments have shown that the proposed method is effective in that the estimated proficiency highly correlates with non-native speakers' real proficiency as represented by TOEIC scores; native-speakers can achieve higher scores than non-native speakers. It is possible to reduce the size of the test by removing non-discriminative questions with *item information* in IRT.

[*] There are many works on item generation theory (ITG) such as Irvine and Kyllonen (2002), although we do not go any further into the area. We focus only on multiple-choice questions for language learning in this paper.

[†] Based on a fact stated in a textbook like, "A prepositional phrase at the beginning of a sentence constitutes an *introductory modifier*," Mitkov generates a question such as, "*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*."

The method provides teachers, testers, and test takers with novel merits that enable low-cost testing of second-language proficiency and provides learners with up-to-date and interesting materials suitable for individuals.

Further research should be done on (1) large-scale evaluation of the proposal, (2) application to different languages such as Chinese and Korean, and (3) generation of different types of questions.

Acknowledgements

The authors' heartfelt thanks go to anonymous reviewers for providing valuable suggestions and *Kadokawa-Shoten* for providing the thesaurus named *Ruigo-Shin-Jiten*. The research reported here was supported in part by a contract with the NiCT entitled, "A study of speech dialogue translation technology based on a large corpus." It was also supported in part by the Grants-in-Aid for Scientific Research (KAKENHI), contract with MEXT numbered 16300048. The study was conducted in part as a cooperative research project by KDDI and ATR.

References

Alderson, Charles. 1996. *Do corpora have a role in language assessment?* Using Corpora for Language Research, eds. Thomas, J. and Short, M., Longman: 248—259.

Brown, J. D. 1993. *What are the characteristics of natural cloze tests?* Language Testing 10: 93—116.

Crystal, David. 2003. *English as a Global Language, (Second Edition).* Cambridge University Press: 212.

Educational Testing Service 2002. *TOEIC koushiki gaido & mondaishu.* IIBC: 249.

Embretson, Susan et al. 2000. *Item Response Theory for Psychologists.* LEA: 371.

Grefenstette, G. 1999. *The WWW as a resource for example-based MT tasks.* ASLIB "Translating and the Computer" conference.

Irvine, H. S., and Kyllonen, P. C. (2002). *Item generation for test development*. LEA: 412.

Izumi, E., and Isahara, H. (2004). *Investigation into language learners' acquisition order based on the error analysis of the learner corpus*. In Proceedings of Pacific-Asia Conference on Language, Information and Computation (PACLIC) 18 Satellite Workshop on E-Learning, Japan. (in printing)

Kikui, G., Sumita, E., Takaezawa, T. and Yamamoto, S., "Creating Corpora for Speech-to-Speech Translation," Special Session "Multilingual Speech-to-Speech Translation" of EuroSpeech, 2003.

Kilgarriff, A. and Grefenstette, G. 2003. *Special Issue on the WEB as Corpus.* Computational Linguistics 29 (3): 333—502.

Mitkov, Ruslan and Ha, Le An. 2003. *Computer-Aided Generation of Multiple-Choice Tests.* HLT-NAACL 2003 Workshop: Building Educational Applications Using Natural Language Processing: 17—22.

Ohno, S. and Hamanishi, M. 1984. *Ruigo-Shin-Jiten*, Kadokawa, Tokyo (in Japanese)

Shermis, M. D. and Burstein. J. C. 2003. *Automated Essay Scoring.* LEA: 238.

Tonoike, M., Sato, S., and Utsuro, T. 2004. *Answer Validation by Keyword Association.* IPSJ, SIGNL, 161: 53—60, (in Japanese).

Turney, P.D. 2001. *Mining the Web for synonyms: PMI-IR vs. LSA on TOEFL.* ECML 2001: 491—502.

Wainer, Howard et al. 2000. *Conputerized Adaptive Testing: A Primer, (Second Edition).* LEA: 335.

Wilson, E. 1997. *The Automatic Generation of CALL exercises from general corpora*, in eds. Wichmann, A., Fligelstone, S., McEnery, T., Knowles, G., Teaching and Language Corpora, Harlow: Longman:116-130.

Yasuda, K., Sugaya, F., Sumita, E., Takezawa, T., Kikui, G. and Yamamoto, S. 2004. *Automatic Measuring of English Language Proficiency using MT Evaluation Technology*, COLING 2004 eLearning for Computational Linguistics and Computational Linguistics for eLearning: 53-60.

Evaluating State-of-the-Art Treebank-style Parsers for Coh-Metrix and Other Learning Technology Environments

Christian F. Hempelmann, Vasile Rus, Arthur C. Graesser, and **Danielle S. McNamara**
Institute for Intelligent Systems
Departments of Computer Science and Psychology
The University of Memphis
Memphis, TN 38120, USA
{chmplmnn, vrus, a-graesser, dsmcnamr}@memphis.edu

Abstract

This paper evaluates a series of freely available, state-of-the-art parsers on a standard benchmark as well as with respect to a set of data relevant for measuring text cohesion. We outline advantages and disadvantages of existing technologies and make recommendations. Our performance report uses traditional measures based on a gold standard as well as novel dimensions for parsing evaluation. To our knowledge this is the first attempt to evaluate parsers accross genres and grade levels for the implementation in learning technology.

1 Introduction

The task of syntactic parsing is valuable to most natural language understanding applications, e.g., anaphora resolution, machine translation, or question answering. Syntactic parsing in its most general definition may be viewed as discovering the underlying syntactic structure of a sentence. The specificities include the types of elements and relations that are retrieved by the parsing process and the way in which they are represented. For example, Treebank-style parsers retrieve a bracketed form that encodes a hierarchical organization (tree) of smaller elements (called phrases), while Grammatical-Relations(GR)-style parsers explicitly output relations together with elements involved in the relation (subj(John,walk)).

The present paper presents an evaluation of parsers for the Coh-Metrix project (Graesser et al., 2004) at the Institute for Intelligent Systems of the University of Memphis. Coh-Metrix is a text-processing tool that provides new methods of automatically assessing text cohesion, readability, and difficulty. In its present form, v1.1, few cohesion measures are based on syntactic information, but its next incarnation, v2.0, will depend more heavily on hierarchical syntactic information. We are developing these measures. Thus, our current goal is to provide the most reliable parser output available for them, while still being able to process larger texts in real time. The usual trade-off between accuracy and speed has to be taken into account.

In the first part of the evaluation, we adopt a constituent-based approach for evaluation, as the output parses are all derived in one way or another from the same data and generate similar, bracketed output. The major goal is to consistently evaluate the freely available state-of-the-art parsers on a standard data set and across genre on corpora typical for learning technology environments. We report parsers' competitiveness along an array of dimensions including performance, robustness, tagging facility, stability, and length of input they can handle.

Next, we briefly address particular types of misparses and mistags in their relation to measures planned for Coh-Metrix 2.0 and assumed to be typical for learning technology applications. Coh-Metrix 2.0 measures that centrally rely on good parses include:
causal and intentional cohesion, for which the main verb and its subject must be identified;

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 69–76, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

anaphora resolution, for which the syntactic relations of pronoun and referent must be identified;
temporal cohesion, for which the main verb and its tense/aspect must be identified.

These measures require complex algorithms operating on the cleanest possible sentence parse, as a faulty parse will lead to a cascading error effect.

1.1 Parser Types

While the purpose of this work is not to propose a taxonomy of all available parsers, we consider it necessary to offer a brief overview of the various parser dimensions. Parsers can be classified according to their general approach (hand-built-grammar-based versus statistical), the way rules in parses are built (selective vs. generative), the parsing algorithm they use (LR, chart parser, etc.), type of grammar (unification-based grammars, context-free grammars, lexicalized context-free grammars, etc.), the representation of the output (bracketed, list of relations, etc.), and the type of output itself (phrases vs grammatical relations). Of particular interest to our work are Treebank-style parsers, i.e., parsers producing an output conforming to the Penn Treebank (PTB) annotation guidelines. The PTB project defined a tag set and bracketed form to represent syntactic trees that became a standard for parsers developed/trained on PTB. It also produced a treebank, a collection of hand-annotated texts with syntactic information.

Given the large number of dimensions along which parsers can be distinguished, an evaluation framework that would provide both parser-specific (to understand the strength of different technologies) and parser-independent (to be able to compare different parsers) performance figures is desirable and commonly used in the literature.

1.2 General Parser Evaluation Methods

Evaluation methods can be broadly divided into non-corpus- and corpus-based methods with the latter subdivided into unannotated and annotated corpus-based methods (Carroll et al., 1999). The non-corpus method simply lists linguistic constructions covered by the parser/grammar. It is well-suited for hand-built grammars because during the construction phase the covered cases can be recorded. However, it has problems with capturing complexities occuring from the interaction of covered cases.

The most widely used corpus-based evaluation methods are: (1) the constituent-based (phrase structure) method, and (2) the dependency/GR-based method. The former has its roots in the Grammar Evaluation Interest Group (GEIG) scheme (Grishman et al., 1992) developed to compare parsers with different underlying grammatical formalisms. It promoted the use of phrase-structure bracketed information and defined Precision, Recall, and Crossing Brackets measures. The GEIG measures were extended later to constituent information (bracketing information plus label) and have since become the standard for reporting automated syntactic parsing performance. Among the advantages of constituent-based evaluation are generality (less parser specificity) and fine grain size of the measures. On the other hand, the measures of the method are weaker than exact sentence measures (full identity), and it is not clear if they properly measure how well a parser identifies the true structure of a sentence. Many phrase boundary mismatches spawn from differences between parsers/grammars and corpus annotation schemes (Lin, 1995). Usually, treebanks are constructed with respect to informal guidelines. Annotators often interpret them differently leading to a large number of different structural configurations.

There are two major approaches to evaluate parsers using the constituent-based method. On the one hand, there is the expert-only approach in which an expert looks at the output of a parser, counts errors, and reports different measures. We use a variant of this approach for the directed parser evaluation (see next section). Using a gold standard, on the other hand, is a method that can be automated to a higher degree. It replaces the counting part of the former method with a software system that compares the output of the parser to the gold standard, highly accurate data, manually parsed – or automatically parsed and manually corrected – by human experts. The latter approach is more useful for scaling up evaluations to large collections of data while the expert-only approach is more flexible, allowing for evaluation of parsers from new perspectives and with a view to special applications, e.g., in learning technology environments.

In the first part of this work we use the gold standard approach for parser evaluation. The evaluation is done from two different points of view. First, we offer a uniform evaluation for the parsers on section 23 from the Wall Street Journal (WSJ) section of PTB, the community norm for reporting parser performance. The goal of this first evaluation is to offer a good estimation of the parsers when evaluated in identical environments (same configuration parameters for the evaluator software). We also observe the following features which are extremely important for using the parsers in large-scale text processing and to embed them as components in larger systems.

Self-tagging: whether or not the parser does tagging itself. It is advantageous to take in raw text since it eliminates the need for extra modules.

Performance: if the performance is in the mid and upper 80th percentiles.

Long sentences: the ability of the parser to handle sentences longer than 40 words.

Robustness: relates to the property of a parser to handle any type of input sentence and return a reasonable output for it and not an empty line or some other useless output.

Second, we evaluate the parsers on narrative and expository texts to study their performance across the two genres. This second evaluation step will provide additional important results for learning technology projects. We use *evalb* (http://nlp.cs.nyu.edu/evalb/) to evaluate the bracketing performance of the output of a parser against a gold standard. The software evaluator reports numerous measures of which we only report the two most important: labelled precision (LR), labelled recall (LR) which are discussed in more detail below.

1.3 Directed Parser Evaluation Method

For the third step of this evaluation we looked for specific problems that will affect Coh-Metrix 2.0, and presumably learning technology applications in general, with a view to amending them by postprocessing the parser output. The following four classes of problems in a sentence's parse were distinguished:

None: The parse is generally correct, unambiguous, poses no problem for Coh-Metrix 2.0.

One: There was one minor problem, e.g., a mislabeled terminal or a wrong scope of an adverbial or prepositional phrase (wrong attachment site) that did not affect the overall parse of the sentence, which is therefore still usable for Coh-Metrix 2.0 measures.

Two: There were two or three problems of the type one, or a problem with the tree structure that affected the overall parse of the sentence, but not in a fatal manner, e.g., a wrong phrase boundary, or a mislabelled higher constituent.

Three: There were two or more problems of the type two, or two or more of the type one as well as one or more of the type two, or another fundamental problem that made the parse of the sentence completely useless, unintelligible, e.g., an omitted sentence or a sentence split into two, because a sentence boundary was misidentified.

2 Evaluated Parsers

2.1 Apple Pie

Apple Pie (AP) (Sekine and Grishman, 1995) extracts a grammar from PTB v.2 in which S and NP are the only true non-terminals (the others are included into the right-hand side of S and NP rules). The rules extracted from the PTB have S or NP on the left-hand side and a flat structure on the right-hand side, for instance S → NP VBX JJ. Each such rule has the most common structure in the PTB associated with it, and if the parser uses the rule it will generate its corresponding structure. The parser is a chart parser and factors grammar rules with common prefixes to reduce the number of active nodes. Although the underlying model of the parser is simple, it can't handle sentences over 40 words due to the large variety of linguistic constructs in the PTB.

2.2 Charniak's Parser

Charniak presents a parser (CP) based on probabilities gathered from the WSJ part of the PTB (Charniak, 1997). It extracts the grammar and probabilities and with a standard context-free chart-parsing mechanism generates a set of possible parses for each sentence retaining the one with the highest probability (probabilities are not computed for all possible parses). The probabilities of an entire tree are computed bottom-up. In (Charniak, 2000), he proposes a generative model based on a Markov-grammar. It uses a standard bottom-up, best-first probabilistic parser to first generate possible parses before ranking them with a probabilistic model.

2.3 Collins's (Bikel's) Parser

Collins's statistical parser (CBP; (Collins, 1997)), improved by Bikel (Bikel, 2004), is based on the probabilities between head-words in parse trees. It explicitly represents the parse probabilities in terms of basic syntactic relationships of these lexical heads. Collins defines a mapping from parse trees to sets of dependencies, on which he defines his statistical model. A set of rules defines a head-child for each node in the tree. The lexical head of the head-child of each node becomes the lexical head of the parent node. Associated with each node is a set of dependencies derived in the following way. For each non-head child, a dependency is added to the set where the dependency is identified by a triplet consisting of the non-head-child non-terminal, the parent non-terminal, and the head-child non-terminal. The parser is a CYK-style dynamic programming chart parser.

2.4 Stanford Parser

The Stanford Parser (SP) is an unlexicalized parser that rivals state-of-the-art lexicalized ones (Klein and Manning, 2003). It uses a context-free grammar with state splits. The parsing algorithm is simpler, the grammar smaller and fewer parameters are needed for the estimation. It uses a CKY chart parser which exhaustively generates all possible parses for a sentence before it selects the highest probability tree. Here we used the default lexicalized version.

3 Experiments and Results

3.1 Text Corpus

We performed experiments on three data sets. First, we chose the norm for large scale parser evaluation, the 2416 sentences of WSJ section 23. Since parsers have different parameters that can be tuned leading to (slightly) different results we first report performance values on the standard data set and then use same parameter settings on the second data set for more reliable comparison.

The second experiment is on a set of three narrative and four expository texts. The gold standard for this second data set was built manually by the authors starting from CP's as well as SP's output on those texts. The four texts used initially are two expository and two narrative texts of reasonable length for detailed evaluation:

The Effects of Heat (SRA Real Science Grade 2 Elementary Science): expository; 52 sentences, 392 words: 7.53 words/sentence;

The Needs of Plants (McGraw-Hill Science): expository; 46 sentences, 458 words: 9.96 words/sentence;

Orlando (Addison Wesley Phonics Take-Home Reader Grade 2): narrative; 65 sentences, 446 words: 6.86 words/sentence;

Moving (McGraw-Hill Reading - TerraNova Test Preparation and Practice - Teachers Edition Grade 3): narrative, 33 sentences, 433 words: 13.12 words/sentence.

An additional set of three texts was chosen from the Touchstone Applied Science Associates, Inc., (TASA) corpus with an average sentence length of 13.06 (overall TASA average) or higher.

Barron17: expository; DRP=75.14 (college grade); 13 sentences, 288 words: 22.15 words/sentence;

Betty03: narrative; DRP=56.92 (5th grade); 14 sentences, 255 words: 18.21 words/sentence;

Olga91: expository; DRP=74.22 (college grade); 12 sentences, 311 words: 25.92 words/sentence.

We also tested all four parsers for speed on a corpus of four texts chosen randomly from the Metametrix corpus of school text books, across high and low grade levels and across narrative and science texts (see Section 3.2.2).
G4: 4th grade narrative text, 1,500 sentences, 18,835 words: 12.56 words/sentence;
G6: 6th grade science text, 1,500 sentences, 18,237 words: 12.16 words/sentence;
G11: 11th grade narrative text, 1,558 sentences, 18,583 words: 11.93 words/sentence;
G12: 12th grade science text, 1,520 sentences, 25,098 words: 16.51 words/sentence.

3.2 General Parser Evaluation Results

3.2.1 Accuracy

The parameters file we used for *evalb* was the standard one that comes with the package. Some parsers are not robust, meaning that for some input they do not output anything, leading to empty lines that are not handled by the evaluator. Those parses had to be "aligned" with the gold standard files so that empty lines are eliminated from the output file together with their peers in the corresponding gold standard files.

In Table 1 we report the performance values on Section 23 of WSJ. Table 2 shows the results for our own corpus. The table gives the average values of two test runs, one against the SP-based gold standard, the other against the CP-based gold standard, to counterbalance the bias of the standards. Note that CP and SP possibly still score high because of this bias. However, CBP is clearly a contender despite the bias, while AP is not.[1] The reported metrics are Labelled Precision (LP) and Labelled Recall (LR). Let us denote by a the number of correct phrases in the output from a parser for a sentence, by b the number of incorrect phrases in the output and by c the number of phrases in the gold standard for the same sentence. LP is defined as $a/(a+b)$ and LR is defined as a/c. A summary of the other dimensions of the evaluation is offered in Table 3. A stability dimension is not reported because we were not able to find a bullet-proof parser so far, but we must recognize that some parsers are significantly more stable than others, namely CP and CBP. In terms of resources needed, the parsers are comparable, except for AP which uses less memory and processing time. The LP/LR of AP is significantly lower, partly due to its outputting partial trees for longer sentences. Overall, CP offers the best performance.

Note in Table 1 that CP's tagging accuracy is worst among the three top parsers but still delivers best overall parsing results. This means that its parsing-only performance is slighstly better than the numbers in the table indicate. The numbers actually represent the tagging and parsing accuracy of the tested parsing systems. Nevertheless, this is what we would most likely want to know since one would prefer to input raw text as opposed to tagged text. If more finely grained comparisons of only the parsing aspects of the parsers are required, perfect tags extracted from PTB must be provided to measure performance.

Table 4 shows average measures for each of the parsers on the PTB and seven expository and narrative texts in the second column and for expository and narrative in the fourth column. The third and fifth columns contain standard deviations for the previous columns, respectively. Here too, CP shows the best result.

3.2.2 Speed

All parsers ran on the same Linux Debian machine: P4 at 3.4GHz with 1.0GB of RAM.[2] AP's and SP's high speeds can be explained to a large degree by their skipping longer sentences, the very ones that lead to the longer times for the other two candidates. Taking this into account, SP is clearly the fastest, but the large range of processing times need to be heeded.

3.3 Directed Parser Evaluation Results

This section reports the results of expert rating of texts for specific problems (see Section 1.3). The best results are produced by CP with an average of 88.69% output useable for Coh-Metrix 2.0 (Table 6). CP also produces good output

[1] AP's performance is reported for sentences < 40 words in length, 2,250 out of 2,416. SP is also not robust enough and the performance reported is only on 2,094 out of 2,416 sentences in section 23 of WSJ.

[2] Some of the parsers also run under Windows.

Table 1: Accuracy of Parsers.

Parser	Performance(LP/LR/Tagging - %)		
	WSJ 23	Expository	Narrative
Applie Pie	43.71/44.29/90.26	41.63/42.70	42.84/43.84
Charniak's	84.35/88.28/92.58	91.91/93.94	93.74/96.18
Collins/Bikel's	84.97/87.30/93.24	82.08/85.35	67.75/85.19
Stanford	84.41/87.00/95.05	75.38/85.12	62.65/87.56

Table 2: Performance of parsers on the narrative and expository text (average against CP-based and SP-based gold standard).

File	Performance (LR/LP - %)			
	AP	CP	CBP	SP
Heat	48.25/47.59	91.96/93.77	92.47/94.14	92.44/91.85
Plants	41.85/45.89	85.34/88.02	78.24/88.45	81.00/85.62
Orlando	45.82/49.03	85.83/91.88	65.87/93.97	57.75/90.72
Moving	37.77/41.45	88.93/92.74	53.94/91.68	76.56/84.97
Barron17	43.22/42.95	89.74/91.32	80.49/89.32	87.22/86.31
Betty03	46.53/44.67	90.77/90.74	87.95/85.21	74.53/80.91
Olga91	32.29/32.69	77.65/80.04	61.61/75.43	61.65/70.60

Table 3: Evaluation of Parsers with Respect to the Criteria Listed at the Top of Each Column.

Parser	Self-tagging	Performance	Long-sentences	Robustness
AP	Yes	No	No	No
CP	Yes	Yes	Yes	Yes
CBP	Yes	Yes	Yes	Yes
SP	Yes	Yes	No	No

Table 4: Average Performance of Parsers.

Parser	Ave. (LR/LP - %)	S.D. (%)	Ave. on	S.D. on
			Exp+Nar (LR/LP - %)	Exp+Nar (%)
AP	42.73/43.61	1.04/0.82	42.24/43.46	5.59/5.41
CP	90.00/92.80	4.98/4.07	87.17/89.79	4.85/4.66
CBP	78.27/85.95	9.22/1.17	74.36/88.31	14.24/6.51
SP	74.14/86.56	10.93/1.28	75.88/84.42	12.66/7.11

Table 5: Parser Speed in Seconds.

	G4	G6	G11	G12
#sent	619	3336	4976	2215
AP	144	89	144	242
CP	647	499	784	1406
CBP	485	1947	1418	1126
SP	449	391	724	651
Ave.	431	732	768	856

most consistently at a standard deviation over the seven texts of 8.86%. The other three candidates are clearly trailing behing, namely by between 5% (SP) and 11% (AP). The distribution of severe problems is comparable for all parsers.

Table 6: Average Performance of Parsers over all Texts (Directed Evaluation).

	Ave. (%)	S.D. (%)
AP	77.31	15.00
CP	88.69	8.86
CBP	79.82	18.94
SP	83.43	11.42

As expected, longer sentences are more problematic for all parsers, as can be seen in Table 7. No significant trends in performance differences with respect to genre difference, narrative (Orlando, Moving, Betty03) vs. expository texts (Heat, Plants, Barron17, Olga91), were detected (cf. also speed results in Table 5). But we assume that the difference in average sentence length obscures any genre differences in our small sample.

The most common non-fatal problems (type one) involved the well-documented adjunct attachment site issue, in particular for prepositional phrases ((Abney et al., 1999), (Brill and Resnik, 1994), (Collins and Brooks, 1995)) as well as adjectival phrases (Table 8)[3]. Similar misattachment issues for adjuncts are encountered with adverbial phrases, but they were rare in our corpus.

Table 7: Correlation of Average Performance per Text for all Parsers and Average Sentence Length (Directed Evaluation).

Text	perf. (%)	length (#words)
Heat	92.31	7.54
Plants	90.76	9.96
Orlando	93.46	6.86
Moving	90.91	13.12
Barron17	76.92	22.15
Betty03	71.43	18.21
Olga91	60.42	25.92

Another common problem are deverbal nouns and denominal verbs, as well as *-ing*/VBG forms. They share surface forms leading to ambiguous part of speech assignments. For many Coh-Metrix 2.0 measures, most obviously temporal cohesion, it is necessary to be able to distinguish gerunds from gerundives and deverbal adjectives and deverbal nouns.

Table 8: Specific Problems by Parser.

	PP	ADV	cNP	&X
AP	13	10	8	9
CP	15	1	2	7
CBP	10	0	0	13
SP	22	6	3	4
Sum	60	17	13	33

Problems with NP misidentification are particularly detrimental in view of the important role of NPs in Coh-Metrix 2.0 measures. This pertains in particular to the mistagging/misparsing of complex NPs and the coordination of NPs. Parses with fatal problems are expected to produce useless results for algorithms operating with them. Wrong coordination is another notorious problem of parsers (cf. (Cremers, 1993), (Grootveld, 1994)). In our corpus we found 33 instances of miscoordination, of which 23 involved NPs. Postprocessing approaches that address these issues are currently under investigation.

[3] PP = wrong attachment site for a prepositional phrase; ADV = wrong attachment site for an adverbial phrase; cNP = misparsed complex noun phrase; &X = wrong coordination

4 Conclusion

The paper presented the evaluation of freely available, Treebank-style, parsers. We offered a uniform evaluation for four parsers: Apple Pie, Charniak's, Collins/Bikel's, and the Stanford parser. A novelty of this work is the evaluation of the parsers along new dimensions such as stability and robustness and across genre, in particular narrative and expository. For the latter part we developed a gold standard for narrative and expository texts from the TASA corpus. No significant effect, not already captured by variation in sentence length, could be found here. Another novelty is the evaluation of the parsers with respect to particular error types that are anticipated to be problematic for a given use of the resulting parses. The reader is invited to have a closer look at the figures our tables provide. We lack the space in the present paper to discuss them in more detail. Overall, Charniak's parser emerged as the most succesful candidate of a parser to be integrated where learning technology requires syntactic information from real text in real time.

ACKNOWLEDGEMENTS

This research was funded by Institute for Educations Science Grant IES R3056020018-02. Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the IES. We are grateful to Philip M. McCarthy for his assistance in preparing some of our data.

References

S. Abney, R. E. Schapire, and Y. Singer. 1999. Boosting applied to tagging and pp attachment. *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 38–45.

D. M. Bikel. 2004. Intricacies of collins' parsing model. *Computational Linguistics*, 30-4:479–511.

E. Brill and P. Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the 15th International Conference on Computational Linguistics.*

J. Carroll, E. Briscoe, and A. Sanfilippo, 1999. *Parser evaluation: current practice*, pages 140–150. EC DG-XIII LRE EAGLES Document EAG-II-EWG-PR.1.

E. Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, AAAI Press/MIT Press, Menlo Park.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the North-American Chapter of Association for Computational Linguistics*, Seattle, Washington.

M. Collins and J. Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistic*, Madrid, Spain.

C. Cremers. 1993. *On Parsing Coordination Categorially.* Ph.D. thesis, Leiden University.

A. C. Graesser, D.S. McNamara, M. M. Louwerse, and Z. Cai. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, and Computers*, 36-2:193–202.

R. Grishman, C. MacLeod, and J. . Sterling. 1992. Evaluating parsing strategies using standardized parse files. *In Proceedings of the Third Conference on Applied Natural Language Processing*, pages 156–161.

M. Grootveld. 1994. *Parsing Coordination Generatively.* Ph.D. thesis, Leiden University.

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistic*, Sapporo, Japan.

D. Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1420–1427.

A. Ratnaparkhi, J. Renyar, and S. Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Workshop on Human Language Technology.*

S. Sekine and R. Grishman. 1995. A corpus-based probabilistic grammar with only two non-terminals. *Proceedings of the International Workshop on Parsing Technologies*, pages 216–223.

A Software Tool for Teaching Reading Based on Text-to-Speech Letter-to-Phoneme Rules

Marian J. Macchi **Dan Kahn**
E-Speech Corporation
Princeton, NJ 08540

mjm@espeech.com dk@espeech.com

Abstract

Native speakers of English who are good readers can "sound out" words or names from printed text, even if they have never seen them before, although they may not be conscious of the strategies they use. No tools are available today that can convey that knowledge to learners, showing them the rules that apply in English text. We have adapted the letter-to-phoneme component of a text-to-speech synthesizer to a web-based software system that can teach word decoding to non-native speakers of English, English-speaking children, and adult learners.

1 Introduction

Learning to read a language like English involves learning many different operations, including phonemic awareness, word recognition, fluency, verbal comprehension, and expression. The research in this project focuses on the pronunciation aspect of reading from the printed page: understanding how letters, or graphemes, in words are related to sounds, or phonemes.

Most people recognize that the relationship between English orthography and phonetic representation is complex and somewhat arbitrary. Although there is significant evidence that phonological information plays an important role in word reading (Kayner, Foorman, Perfetti, Pesetsky, and Seidenberg, 2001), the precise role of "phonics rules" that would allow a learner to "sound out" a printed word has been debated by educators as well as by cognitive psychologists, and many versions of phonics rules have been discussed by educators.

A classic paper by Clymer (1963) argued that most of the phonics generalizations taught in elementary school are not valid most of the time. Clymer found that for many of the rules, there were so many exceptions that the rule had little utility as a generalization for teaching learners to sound out a word of English. However, the Clymer results do not necessarily mean that phonic generalizations are not useful to readers. Since Clymer, there have been many papers that have suggested alternate formulations of the letter-to-phoneme rules for teaching reading. For example, a recent study by Johnston (2001) found one reason that Clymer considered phonics rules to be unreliable is because the rules he evaluated were too general. Today, there is no consensus on a set of rules, nor does there exist any complete, explicit rule system that "decodes" any word or proper name of English for learners.

E-Speech's letter-to-phoneme (LTP) software, developed over many years for text-to-speech and speech recognition applications, uses proprietary rules to produce pronunciations for any input text. We have adapted the LTP software into a prototype web-based, interactive online system that teaches word pronunciation by explicitly presenting rules for those words/names pronounced according to regular rules and by showing exceptions to the rules. The system allows students to view families of words that obey any given rule and to view words with the same letter patterns that obey different rules.

Our intent is to develop a system that can provide phonics training for beginning readers, either children or adults who are native speakers of En-

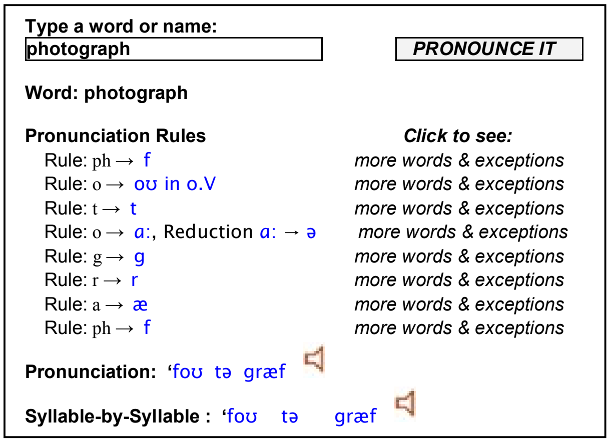
Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 77–84, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

glish, as well as for nonnative speakers of English and language-disabled learners. We envision the system either as part of an interactive dictionary or general language-teaching package or stand-alone as an instructional tool for teaching word pronunciation.

A major challenge is to identify rules that are useful for learners and to present them effectively. We have begun to test our prototype system with nonnative speakers of English who were studying English as a second or foreign language. Our preliminary results indicate that the software was (1) useful in improving nonnative speakers' pronunciation of English words; (2) effective at teaching both "basic" pronunciation rules, such as those commonly taught in phonics programs, and some novel, proprietary pronunciation rules.

2 Software Design

The *Word Pronunciation* tool allows a student to enter *any* word or name – whether it is in any dictionary or not - and see our set of rules that account for its pronunciation. The screen capture below shows the output for the word "photograph". The student can also hear the word pronounced, either normally or syllable-by-syllable.[1]



Type a word or name:
photograph

PRONOUNCE IT

Word: photograph

Pronunciation Rules | ***Click to see:***

Rule: ph → f | *more words & exceptions*
Rule: o → oʊ in o.V | *more words & exceptions*
Rule: t → t | *more words & exceptions*
Rule: o → ɑ:, Reduction ɑ: → ə | *more words & exceptions*
Rule: g → g | *more words & exceptions*
Rule: r → r | *more words & exceptions*
Rule: a → æ | *more words & exceptions*
Rule: ph → f | *more words & exceptions*

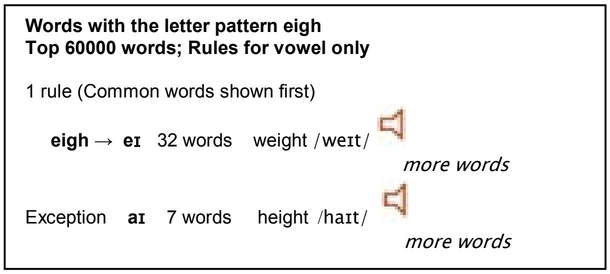
Pronunciation: ‘foʊ tə græf

Syllable-by-Syllable : ‘foʊ   tə   græf

Figure 1 : Word Pronunciation tool display

In addition, a student can click on any rule and see other words obeying the same rule as well as words that are exceptions to the rule.[2]

The *Letter Pattern* tool allows a student to enter a letter or sequence of letters (ie, a letter pattern) and see the rules that apply to that pattern and exceptions to those rules. For example, a student confused by the fact that "how" and "snow" don't rhyme can enter the letter pattern "ow" and view the various generalizations (rules) that determine the pronunciation of this letter string in different contexts, as well as words that don't follow these generalizations (exceptions). The software underlying this tool allows the user to tailor the output to his needs of the moment. For example, one can choose 1-syllable versus multisyllabic words as targets for the rules, how many sample words to output by default, and how big a vocabulary from which to draw words. A simple example of the operation of this tool is illustrated in Figure 2**.**



Words with the letter pattern eigh
Top 60000 words; Rules for vowel only

1 rule (Common words shown first)

eigh → eɪ 32 words weight /weɪt/
more words

Exception **aɪ** 7 words height /haɪt/
more words

Figure 2: Letter Pattern tool display

While we developed the Letter Pattern tool for general use by learners, we used its underlying search engine in exercises designed to diagnose and teach pronunciation rules.

We implemented a framework for a self-paced set of exercises that allows the user to work alone to diagnose his pronunciation-rule weaknesses and learn the rules necessary to correct his errors. We used this framework to assess the effectiveness of our rules and system for learners.

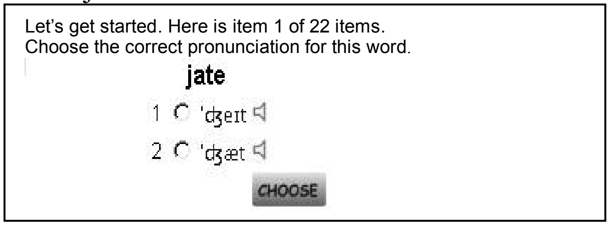
In the typical exercise interaction, the user sees a sequence of words and must choose the correct pronunciation for each. He indicates his choice of

[1] The system uses the International Phonetic Alphabet (IPA) to represent phonetic transcriptions, because most of our target population, adult foreign-born learners of English as a second language, were familiar with this alphabet, since it is used in many English learners' dictionaries.

[2] In this prototype, we worked on presenting the segmental rules, that is, rules for pronouncing phonemes. Although our letter-to-phoneme software assigns lexical stress (to indicate which syllable bears primary stress in polysyllabic words), the stress assignment algorithm is quite complicated. Although the algorithm is accurate, it is too complicated for a human to apply for learning. We also ignored the rules for morphological decomposition, such as for analyzing "walking' into "walk" plus "ing" or "snowman" into "snow" plus "man".

pronunciation by clicking on one of several options, represented in the International Phonetic Alphabet (IPA) or by clicking on a speaker-symbol, so that he can hear the options spoken. An example of a test item would be the nonsense word “doke”. If the user chooses an incorrect pronunciation for this word, he is told the correct pronunciation, as well as the relevant rule, which in this case is that an 'o' followed by a single consonant followed by a final 'e' is pronounced /oʊ/. The user can choose to see actual examples of the rule in action (“smoke”, “home”, etc.) and other rules involving 'o' (eg., the default pronunciation /ɑ:/, as in “hot”).

In some cases, the exercises tested real English words, and in others, “nonsense” words (words that do not exist in English but are possible as words, because they have letter sequences occurring in English words). Only through knowing the general rules of English pronunciation can a student correctly predict the pronunciation of words he has never seen before. Figure 3 shows the exercise for knowledge of the letter “a” in the nonsense word “jate."



Let’s get started. Here is item 1 of 22 items.
Choose the correct pronunciation for this word.

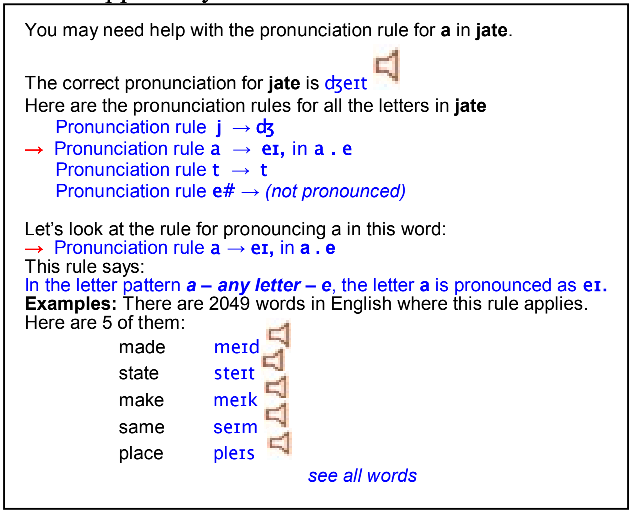
jate

1 ○ 'ʤeɪt

2 ○ 'ʤæt

Figure 3. Exercise example

A wrong answer would cause the screen in Figure 4 to appear, in an attempt to teach the student the rule he apparently hadn’t mastered.



You may need help with the pronunciation rule for **a** in **jate**.

The correct pronunciation for **jate** is ʤeɪt
Here are the pronunciation rules for all the letters in **jate**
Pronunciation rule **j** → **ʤ**
→ Pronunciation rule **a** → **eɪ,** in **a . e**
Pronunciation rule **t** → **t**
Pronunciation rule **e#** → *(not pronounced)*

Let’s look at the rule for pronouncing a in this word:
→ Pronunciation rule **a** → **eɪ,** in **a . e**
This rule says:
In the letter pattern ***a – any letter – e***, the letter **a** is pronounced as **eɪ.**
Examples: There are 2049 words in English where this rule applies.
Here are 5 of them:

made	meɪd
state	steɪt
make	meɪk
same	seɪm
place	pleɪs

see all words

Figure 4. Exercise feedback lesson example

This “lesson” screen highlights the relevant pronunciation rule in the word. Because subjects told us that our pronunciation rule syntax, derived from our LTP rules, was “too mathematical” and hard to understand, the screen also displays an English language explanation for each rule (e.g., “In the letter pattern ***a – any letter – e***, the letter **a** is pronounced as /eɪ/”). In our prototype, we developed a simple text-generation algorithm to translate from our “mathematical” rule syntax (“a → /eɪ/, in a.e”) into normal English for the rules that we tested in our evaluation. Going forward, however, we will need to produce the explanations via a more sophisticated algorithm or simply hand-prepare explanations for the rules.

A subject can click on the “See all words” link to see more English words in which that rule applies. After the “lesson” the learner is given the opportunity to try again, in order to reinforce the correct pronunciation.

The design of the prototype incorporates several features that are important to its extension to a full learning system. First, the set of exercises is table-driven, so that is relatively easy to add a new set of exercises. This feature is important since a complete system will need a large number of exercises. Second, the system is designed so that the corpus of words that serve as examples of the rules can be changed easily. This feature is important since different user groups (e.g., adult nonnative speakers, children, speakers with reading disabilities) may require different kinds of words as examples.

3 Experimental Results

In addition to developing lexical resource tools, we conducted an experiment to determine (1) if our software could be useful in teaching nonnative speakers of English how to pronounce English words, and if so, (2) if both commonly-taught pronunciation rules and pronunciation rules that are idiosyncratic to the E-Speech letter-to-phoneme system can effectively be taught.

We considered testing the lexical resource tools directly by giving students lists of words and instructing them to use the tools to learn the pronunciation rules for the words. However, we felt that a more efficient way of testing our software would be to develop a set of exercises to diagnose and teach various pronunciation rules and then to test how effectively students learned from the exercises. We developed the design of the exercises based on informal comments and results of pretests with more than 40 nonnative speakers of English.

3.1 Experimental Design

We sought to improve nonnative speakers' word pronunciation competence, aiming toward giving them the competence of native speakers of English. Therefore, we included both native and nonnative speakers as subjects. 10 nonnative speakers of English and 7 native speakers of English successfully completed the final set of exercises. Six non-native subjects were undergraduates or graduate students at Montclair State University who had been assigned to an English as a Second Language course based on their performance on an English language test administered by the university. The other four were nonnative speakers of English in Brazil, Bolivia, and Germany. Native languages of the subjects were German, Portuguese, Korean, Spanish, Polish, Bangla (Bangladesh), and Urhobo (Nigeria). The native English-speaking subjects were high school or college students who grew up in New Jersey.[3]

The subjects were assigned logins to the system and were instructed to complete a series of exercises, each of which would present different English pronunciation rules. A subject logged in to the system with a web-browser over the internet, saw a printed word and a set of possible pronunciations for the word (as described above). The student was instructed to listen to the set of choices and to choose the pronunciation that he thought was correct. Subjects were told that each exercise would consist of two parts. The first part of each exercise would identify the pronunciation rules with which a subject might need help and then teach the rule; the second part of the exercise would determine whether teaching the pronunciation rules was effective. In the teaching part of the exercise, each rule was presented several times, as it applied to different words. Subjects were allowed to repeat the first part of each exercise as many times as they wished, until they felt comfortable about proceeding to the test part of the exercise.

Our software logged the students' choice for each word in each part of each exercise and scored it as correct (1) or incorrect (0). We computed the percentage of correct choices, which we call the word pronunciation score. We also logged the number of times a student practiced with the diagnosis/lesson portion of each exercise, and the amount of time a student spent with each item.

The basic exercises were:

Basic: 1-syllable nonsense words representing "basic" rules, rules that are extremely common in English words. These are productive rules (English speakers apply them in nonsense words), and rules capturing these generalizations are commonly taught in phonics programs. Specifically the exercise teaches:

- **a** is pronounced /eɪ/ in the letter sequence **a - any letter - e** , as in **make**
- **a** is pronounced /æ/ by default, as in **cat** and analogous rules for the letters **e, i, o, u.**

The other exercises taught and tested rules from the LTP system, using English words rather than nonsense words as the material. These were:

LTP1: the basic rules for the letter **a**, plus the trisyllabic laxing rule (which we call the **3-syllable rule**), which causes underlying long vowels and diphthongs to shorten to a lax vowel in antepenultimate syllables:

- **a** is pronounced /eɪ/, in **a - any letter - e**, as in **make**
- **a** is pronounced /æ/ in **a – any letter - e** when **a** is 3 syllables from the end of a word (the "3-syllable rule"), as in **tragedy**
- **a** is pronounced /æ/,by default, as in **cat**

LTP2: rules for **a** before the letter **l**:

- **a** is pronounced /ɔ:/ in **all** at the end of the word, as in **ball**
- **a** is pronounced /ɔ:/, in **alt**, as in **salt**
- **a** is pronounced /eɪ/, in **a - any letter - e**, as in **sale** and **make**
- **a** is pronounced /æ/ by default, as in **pal** and **cat**

LTP3: rules for the letter **a** when it is followed by the letter **r**:

- **a** is pronounced /ɔ:/ in **war**
- **a** is pronounced /æ/ in **arr** followed by any vowel , as in **carry**
- **a** is pronounced /ɑ:/ in **ar** at the end of the word, as in **car,** and in **ar** followed by any consonant, as in **part**
- **a** is pronounced /eɪ/, in **a - any letter - e** , as in **care**

LTP4: rules for the letter **a** when it is preceded by the *phoneme* /**w**/:

- **a** is pronounced /ɔ:/ in **war**

[3] Nonnative subjects were told that we had developed software for teaching word pronunciation and we needed nonnative speakers to try the software and see if it helped them to improve their word pronunciation. English-speaking subjects were told that we had developed software for teaching word pronunciation to nonnative subjects, and that we needed to compare the students' performance with native speakers' performance.

- **a** is pronounced /ɑː/ in /**w/a**, as in **watch** and **quality …** *except …*
- **a** is pronounced /æ/ in /**w/a** before the phonemes **/k/, /g/, /m/, /ŋ/,** as in **wag, swam, quack, swang**
- **a** is pronounced /eɪ/ in **a - any letter - e** , as in **wake**

Since each of these exercises included only several rules, the final exercise, ***LTP-all***, recapped the other exercises, in order to assess how well students could integrate all the rules.

LTP-all: an integrated exercise***: all*** the rules for the letter **a** that were presented in the previous exercises, plus the rule

- **a** → /eɪ/ in **aste**, as in **paste**

We chose this particular set of LTP rules because they would allow us to compare the "basic" rules, common to many phonics programs, and rules in our LTP system that are not taught in phonics programs. All the non-basic rules in our experiment were pronunciation rules for the letter "a" and were chosen because they applied to many English words and represented a variety of formal types of rules. For example, some had relatively simple contexts (e.g., "alt"), and some had complicated contexts. For most of the rules, the context was specified in terms of the surrounding *letters* (for example, the letter a when followed by any letter and the letter e). For one rule, the context was specified in terms of the surrounding *phonemes*. This latter type of rule is complicated because it requires that a learner first identify the phonemes for the letters surrounding the target letter "a".

In the exercise on "basic rules", we used nonsense words to teach and test pronunciation rules. Our reasoning was that the strongest test of whether a student knows the rules is to test his pronunciation of nonsense words, since the only way he could possibly know how to pronounce a nonsense word is by applying the rules. Further, we felt there was strong evidence that the "basic rules" are productive in English. That is, native speakers of English know these rules and apply them in novel words and nonsense words. For example, English speakers pronounce the "a" in nonsense words with "a" – consonant – "e" at the end of a word, (e.g., "pake", "glape", "nade") as /eɪ/. Consequently, we felt that teaching and testing with nonsense words would help give nonnative speakers the same competence as native speakers.

However, for the other exercises we used real English words. Our reasoning was that the non-basic rules, although they apply to classes of English words, may not be productive in English. That is, native speakers of English might not apply the rules to nonsense words, even though the rule governs a class of existing English words. For example, in English, "oo" is most commonly /u:/ (as in "coo" and "cool"), but when the "oo" is followed by a final "k", the vowel is almost always pronounced /ʊ/ (e.g., "took", "book", "cook", "brook", "crook", "snook", though there are a few exceptions: "kook", "spook"). The question thus arises whether native speakers of English, who obviously know how to pronounce these words, have internalized the "ook rule" and apply it in novel words. Native speakers do not always pronounce novel "ook" words with /ʊ/; instead, they sometimes use /uː/ in nonsense words, like "mook", "dook", "vook" (see Treiman et. al., 2003). Of course, the fact that a rule is not productive does not mean that it is not useful for teaching students how to pronounce words; clearly it would be useful for students to know that "ook" is usually pronounced /ʊk/. However, since we wanted to compare nonnatives' performance to natives' performance, and we were primarily concerned with teaching nonnative speakers how to pronounce real English words, we chose to teach and test real, as opposed to nonsense, words.[4]

We did include one test of non-basic-rules that used nonsense words, anticipating that native speakers might perform differently from nonnatives, if the nonnatives, who had been explicitly taught pronunciation rules, applied them to nonsense words, even if natives did not apply them productively in nonsense words.

3.2 Experimental Results

We present our results informally, without statistical analysis of significance, primarily because we have to date collected data a relatively small number of subjects. Consequently, we interpret our results as preliminary.

[4] We attempted to choose words that have relatively low frequency-of-occurrence, to minimize the chances that a nonnative speaker would simply know the word.

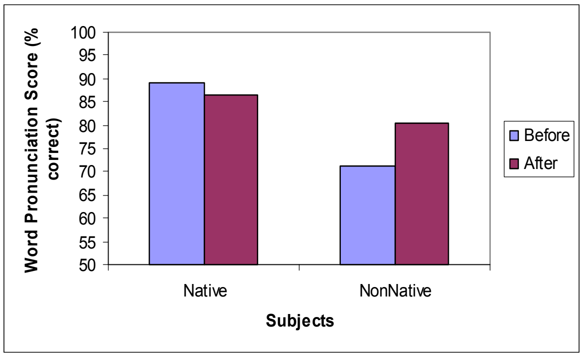


Figure 5. Overall Word Pronunciation Scores

Figure 5 shows word pronunciation scores averaged over all subjects and exercises, tabulated as "before" (word pronunciation scores before subjects were offered any lessons) and "after" (word pronunciation scores from the test parts of the exercises, after the lessons). As would be expected, native speakers had higher word pronunciation scores than nonnative speakers. Further, nonnative speakers had higher word pronunciation scores after completing the lessons than they did before the lessons, although, overall, they did not achieve native speakers' level of word pronunciation. Thus, our data suggests that, overall, nonnative speakers were able to learn aspects of word pronunciation from our system.

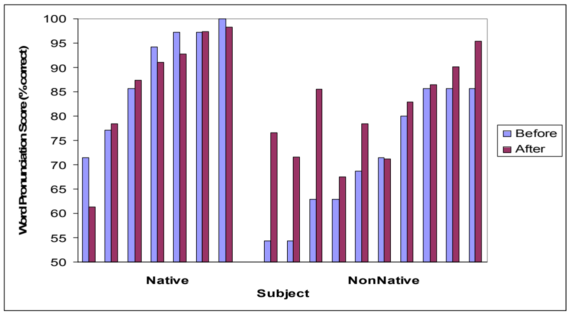


Figure 6. Word Pronunciation Scores by Subject

Figure 6 indicates that there was wide variability among the subjects. Some nonnative subjects' scores increased much more than others', and several subjects' scores did not increase or increased only slightly. Nonnative subjects with higher "before" scores, in general, did not increase as much as the nonnatives with low "before" scores, probably because their scores were high to start with.

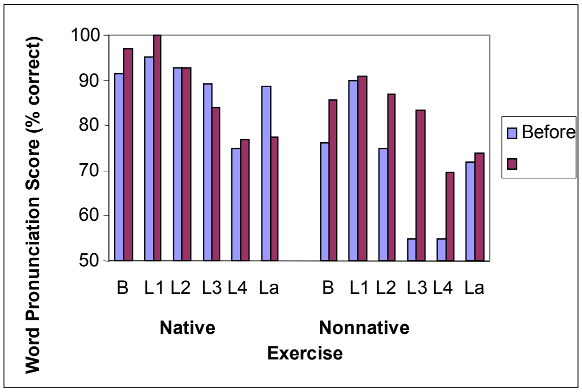


Figure 7. Word Pronunciation Scores by Exercise (B=*Basic*, L1=*LTP1*, L2=*LTP2*, L3=*LTP3*, L4=*LTP4*, La=*LTP-all*)

Figure 7 presents the same data, collapsed across subjects, for the different exercises, which represented different sets of pronunciation rules. We wanted to know whether some exercises proved more learnable than others. In general, as shown at the right side of the Figure 7, for nonnative speakers, for each exercise, the word pronunciation scores were higher after the lessons than before, although the effects were greater for some exercises than others. For native speakers, in contrast, there were no systematic differences in the before versus after scores. However, overall, scores were higher for some exercises than for other exercises even for native speakers. Examination of the native speakers' "incorrect" responses suggests that dialectal issues may have caused some native speakers to choose responses that we did not anticipate. For example, for the word "waffle", some native subjects chose the pronunciation /wɔ:fəl/, although we had assumed that the pronunciation in these subjects' dialect was /wɑ:fəl/.

Figures 8 and 9 suggest that some rules were useful to nonnative subjects. For example, the "basic" rules, in general, were effective; the nonsense words tested after the lessons elicited higher scores than those tested before any lessons. Of the other, letter-to-phoneme-based rules, the "war" rule and the 3-syllable rule seemed to be effective (the "before" bar for "war" is not displayed in Figure 9, because the before score was extremely low, only 20%).

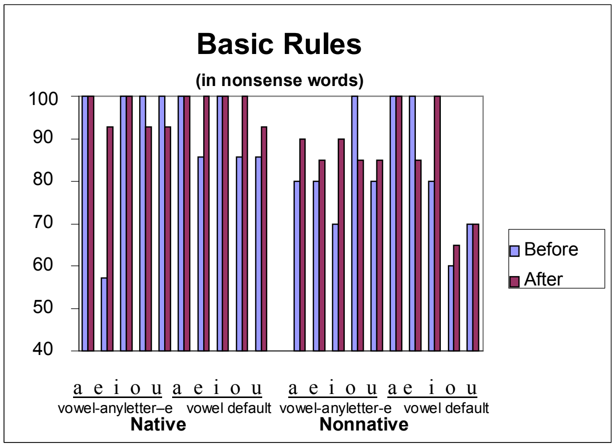


Figure 8. Word Pronunciation Scores for Basic Rules

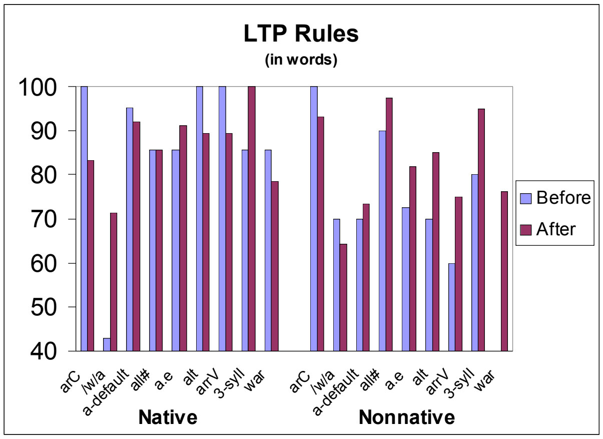


Figure 9. Word Pronunciation Scores for LTP Rules

A complicated rule, the /w/a rule (i.e., the rule that the letter "a" after the *phoneme* /w/ is pronounced /ɑ:/), appeared not to be useful to nonnative subjects. We found no evidence for the effect of teaching for another rule, the "aste" rule, because all nonnative subjects knew the pronunciation of the "aste" words before the lessons. However, there were differences between subjects. One source of this difference is probably due to differences in the nonnative subjects' pre-existing English knowledge; that is some subjects knew some word pronunciations in advance of the lessons. Consequently, for some rules we obtained data for only a few subjects.

As discussed above, since the items in the exercises testing our letter-to-phoneme rules were English words (even though there were not common words) how do we know that subjects' performance was due to our lessons; perhaps the subjects simply knew the words' pronunciation before participating in our experiments? How well do nonnative speakers apply the rules they learned to words that we can be sure they have never seen before? One of our exercises, ***LTP-all*** included a section that contained only nonsense words (e.g., "laterous", "plar", "swarg", "falt").

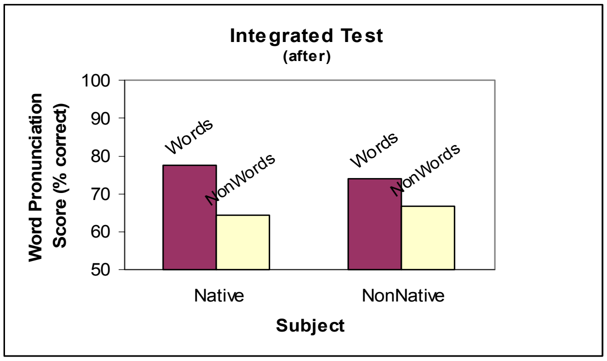


Figure 10. Word Pronunciation Scores for nonwords versus words in the Integrated Rules Test ***LTP-all*** after lessons

Figure 10 presents the results of the nonsense word portion of ***LTP-all***: pronunciation scores for real English words versus nonsense words for nonnative subjects and the analogous scores for native speakers, collapsed across subjects. Although there were between-subject differences, on average, both sets of subjects had lower scores for nonsense words than for words. If subjects based their scores for all test items – words as well as nonsense words – entirely on the word pronunciation rules that we included in our exercises, then we would expect their scores to be the same for words and nonsense words. Since words have an empirically correct pronunciation (they are given in a dictionary, for example), native speakers may be relying on a stored phonological representation for the word items. For nonsense words, however, the subjects must rely on rules or other principles. If subjects used rules or principles for pronouncing nonsense words different from the ones we expected, then the nonsense word scores would be lower than those for words. However, the pronunciation scores for the nonsense words for the nonnative subjects were *higher* than that for natives. This fact suggests that the nonnatives were, in fact, applying the pronunciation rules they had learned in our lessons to the nonwords.

In summary, our preliminary results indicate that: (1) our software was useful in teaching non-native speakers of English how to pronounce English words; (2) both "basic" pronunciation rules and some novel, proprietary pronunciation rules were useful for teaching word pronunciation.

4 Future Research

The major directions for our future research are to select and reformulate LTP rules that are useful for teaching and then to obtain more user data for nonnative speakers and for native-English-speaking children learning to read.

First, we intend to produce a complete learner's rule system for English, based on the entire set of text-to-speech LTP rules. Since our text-to-speech system contains roughly 800 LTP rules, which we believe is too large a number for learners, a significant task is to reduce the number of rules. We intend to focus on rules that apply to large numbers of words and remove rules that apply to few words, relegating the words to which they apply to the exceptions dictionary. We also intend to attempt to collapse rules that apply in similar contexts.

Second, we need to determine whether learners can, in general, understand a pronunciation system that requires rule ordering. In our system, some rules are labeled as "default" rules, meaning that they apply if no other rules apply. Consequently, learners must know *all* the rules in order to know when to apply the default rule. If the number of rules is too large, learners may need a system in which each rule is independently unambiguous.

Third, we need to recruit more nonnative speakers of English who are good candidates for improving their word pronunciation skills. Some of our subjects in this experiment had relatively high word pronunciation scores before exposure to our lessons, so that observing any effect of our lessons was inherently limited. Therefore, we would like to recruit more subjects with lower word pronunciation ability, in order to get a better picture of the effectiveness of our system. Can we predict which students will have high pronunciation scores and which will have low scores based on a student's report of his or her experience in English

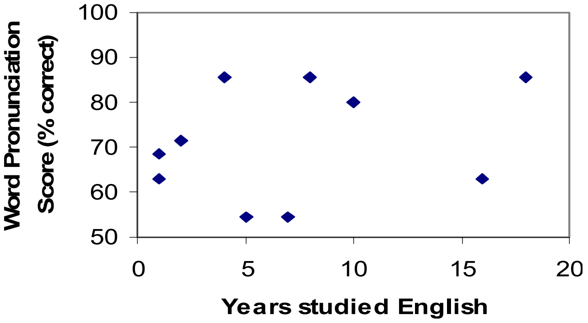


Figure 11. Relation between Word Pronunciation Score and Years Studied English

Figure 11 presents the average word pronunciation scores for each subject at the beginning of the exercises, before doing lessons. As shown, the reported length of time a student had studied English was not correlated with word pronunciation scores. Consequently, we cannot depend on a student's reported length of time studying English as predictive of his word pronunciation abilities. Instead, we will need to screen prospective subjects via pretesting.

Finally, although we have developed our system for nonnative speakers, we would like to test our system with native-English-speaking children learning to read. However, it is likely that the user interface and corpus of exemplar words will need to be different for the child population.

Acknowledgments

This work was supported in part by an SBIR grant from the National Science Foundation.

References

Clymer, T. (1963). The utility of phonic generalizations in the primary grades. *The Reading Teacher,* 50, 182-187.

Johnston, F. P. (2001). The utility of phonic generalizations: Let's take another look at Clymer's conclusions, *The Reading Teacher*, 55, 132-143

Rayner, K., Foorman, B. R., Perfetti, C. A., Pesetsky, D., and Seidenberg, M. S. 2001. How psychological science informs the teaching of reading. *Psychological science in the Public Interest*, vol. 2. 2: 31-94.

Treiman, R, Kessler, B, and Bick, S. 2003. Influence of consonantal context on the pronunciation of vowels: a comparison of human readers and computational models. *Cognition* 88: 49-78.

Situational language training for hotel receptionists

Frédérique Segond, Thibault Parmentier
Xerox Research Centre Europe
Meylan, 38240, France
segond@xrce.xerox.com

Roberta Stock, Ran Rosner
The Marathon Group
Tel Aviv, Israel
rstock@marathon-group.net

Mariola Usteran Muela
Grupo gdt.
Sevilla, Spain
mustaran@grupogdt.com

Abstract

This paper presents the lessons learned in experimenting with Thetis[1], an EC project focusing on the creation and localization of enhanced on-line pedagogical content for language learning in tourism industry. It is based on a general innovative approach to language learning that allows employees to acquire practical oral and written skills while navigating a relevant professional scenario. The approach is enabled by an underlying platform (EXILLS) that integrates virtual reality with a set of linguistic, technologies to create a new form of dynamic, extensible, goal-directed e-content.

1 Credits

The work described in this paper has been supported by the European Commission in the frame of the eContent program[2].

2 Introduction

Thetis focuses on the creation and localization of enhanced on-line pedagogical content for language learning in tourism industry. It is based on a general approach to language learning that allows employees to acquire practical oral and written skills while navigating a relevant professional scenario. The approach is enabled by an underlying platform (EXILLS[3]) that integrates virtual reality with a set of linguistic technologies to create a new form of dynamic, extensible, goal-directed e-content

Thetis has two aims:

- Test the value of linguistic technologies for on-line language learning.
- Localize and repurpose existing e-content for English and vocational blended training material that was first designed for CDROM in order to offer it on-line. The material is meant to develop oral comprehension and reading skills in order to enable end-users to communicate with English speaking customers. It will be used for the continuous vocational training of professionals in this sector through the Internet as well as on-site.

In the following sections we present the following:

- The Thetis scenario and the technologies applied
- Aspects of content adaptation
- the results of the users' evaluation
- the lessons we have learned both regarding the value of the technologies (including the linguistics technologies), and the peda-

[1] THETIS – Training for hotel employees to interact in situations (EDC- 42052). www.thetis-project.org.

[2] http://www.cordis.lu/econtent/

[3] See www.exills.com

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 85–92, Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

gogical value of such an innovative approach to language learning

3 Thetis : scenario and linguistic technologies

As the general technical architecture behind Thetis has already been described in details in (Segond and Parmentier 2004) (Brun et Al. 2002) we just give an overview of the entire system and concentrate below on the description of the Thetis scenario and on the linguistic technologies that have been integrated into the system.

Thetis integrates virtual reality and linguistic technologies in a web application in order to propose a truly e-learning solution that can be used both synchronously and asynchronously. Our motivations for applying these two components are the following:

- Virtual reality offers a cognitive context and promotes interaction.
- Linguistic technologies offer autonomy to the students by showing them concepts, giving them assistance to understand word meanings within particular contexts by presenting various examples, providing feedback on their skills (during chat sessions as well as through exercises or even free activities).

The notion of scenario is central to Thetis. The scenario allows the users to act in typical work situations such as introducing themselves, reading emails, searching for and understanding information, ordering a meal, and interacting with colleagues and customers. The lessons include traditional contents such as grammar rules, exercises, and speech acts. The users interact either via chatting or during the different activities proposed. The Thetis scenario has been explicitly designed for people working at hotels' reception desks.

3.1 Scenario

Students and tutors enter a virtual reality scene either all together at the same time (synchronously) or whenever they want (asynchronously). They are then in a virtual hotel where they are given roles that correspond to the different prototypical hotel situations listed above. All students have to play their everyday job, hotel desk receptionist. They interact with customers. The tutors can choose the role they play: customers, fellow students or tutors. The resulting system can be used either individually from any place with access to the Internet or collectively, all or some students being in the same location (open class, hotel etc.)

The customers are robots and the students are avatars that work at the reception desk.

The robots are 3D-human representations that have been programmed in advance. They can invite users to chat, react to certain stimuli such as predefined lists of words or movements of others in the 3D-scene.

The avatars are non-programmed 3D-human representations of users. The users can decide where to go in the virtual scene, with whom they want to interact, what to say.

The application is composed of several virtual rooms. Each of these rooms corresponds to a different scenario related to the receptionists work tasks.

The students' avatars interact either with the robot-customers, or among themselves. The language of the interactions is exclusively the language being learned i.e. English, (which is automatically checked by the system). In order to encourage interactions among the students and to strengthen the playful aspect of the course, the students are asked to work in groups of two or three. Each group of two or three students enter rooms at the same time, and interact individually with different robot customers. Since they are in the same room they are also able to communicate among themselves, in the language they learn, when they have difficulties. The tutor can also be present. There is a common back room where all the students can go any time to interact with the rest of the student group, be it because they need help or because the tutor asks for a meeting.

In order to call for interaction students and tutors can either type text in the chat window or, in some specific cases (previously defined in the scenario) record their answer and send the speech file to the tutor.

The figure below is a snapshot of the different activities in the different virtual rooms. Traditional linguistic exercises, cards with grammar or speech act hints are associated to the activities of each room.

In the Virtual World students always have access to a phone, a fax, a reservation book, computers, documents internal to the hotel (regulations, menu, price lists, tourist information etc.).

During the different activities, the students interact with robots that play different roles (customers, travel agents, taxi driver etc.). Dialogues between students and robots work as follows: the students type some text in the chat box, and the robots reply orally. Robot participations to dialogues are prerecorded sound files. The robots are configured to reply to stimuli like a new learner arriving in a scene asking for a private chat, etc. For instance, when a student asks a question, the robot can automatically select the most appropriate answer or action based on the dialog's steps or based on keywords found in students' written production.

Examples:

> "Could you show/give me your identity card?" would produce the action of displaying a window with the image of an identity card.
> "How long will you stay?" would produce an answer like "three nights".

When robots ask questions, the students' answers can be used (or not) to select the next move in the dialogue.

Examples:

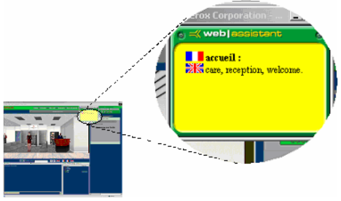
> Robot: "Do all the rooms have an Internet connection?" Student: "No" would imply a specific notice on the reservation fax to require an Internet connection.

3.2 Linguistic Technologies

In terms of tools and resources available in the virtual world, the students have access to real hotel documents (including a list of hotel services, prices and regulations, real identity documents from different countries, etc.)as well as to linguistic technologies.

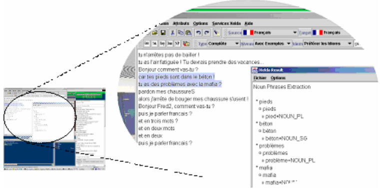
The linguistic tools include the following:

- *A comprehension help* that provides students with the most appropriate contextual translation of any word or expression. Comprehension help is crucial in speeding up both the students' comprehension and written production.



accueil :
care, reception, welcome.

- *A linguistic tool box* that provides the students with customizable services which allow them to parse and to tag their own production in order to check its correctness.

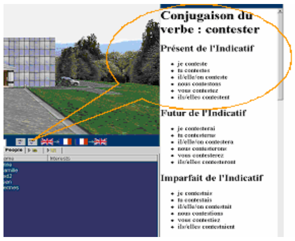


tu n'arrêtes pas de bailler !
tu as l'air fatiguée ! Tu devrais prendre des vacances...
Bonjour comment vas-tu ?
car tes pieds sont dans le béton !
tu as des problèmes avec la mafia ?
pardon mes chaussureS
alors j'arrête de bouger mes chaussure s'usent !
Bonjour Fred2, comment vas-tu ?
puis je parler francais ?
et en trois mots ?
et en deux mots
et en deux
puis je parler francais ?

Noun Phrases Extraction

- pieds
  - pieds
    - pied+NOUN_PL
- béton
  - béton
    - béton+NOUN_SG
- problèmes
  - problèmes
    - problème+NOUN_PL
- mafia
  - mafia
    - mafia+NO…

- *A morphological analyzer* that gives the students access to conjugations or declinations.



Conjugaison du verbe : contester

Présent de l'Indicatif

- je conteste
- tu contestes
- il/elle/on conteste
- nous contestons
- vous contestez
- ils/elles contestent

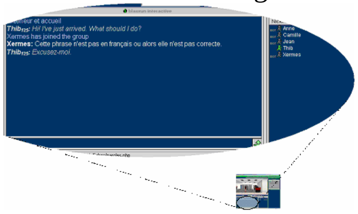
Futur de l'Indicatif

- je contesterai
- tu contesteras
- il/elle/on contestera
- nous contesterons
- vous contesterez
- ils/elles contesteront

Imparfait de l'Indicatif

- je contestais
- tu contestais
- il/elle/on contestait
- nous contestions
- vous contestiez
- ils/elles contestaient

- *A language guesser* that automatically prevents students from interacting in another language than the one they learn i.e. English.



…érieur et accueil
Thib$_{125}$: Hi! I've just arrived. What should I do?
Xermes has joined the group
Xermes: Cette phrase n'est pas en français ou alors elle n'est pas correcte.
Thib$_{125}$: Excusez-moi.

4 Good quality content

The other interest of Thetis is to create and adapt e-content to offer it on-line rather than on CD-ROM. One of the most difficult issues that e-learning faces is global delivery, especially with the challenges of localization and repackaging content for different audiences. Europe has been active in creating high quality pedagogical content for years and is genuinely concerned about languages.

Over the past few years, multimedia publishers have developed electronic content for language learning of very high pedagogical value and very good quality. With the arrival of e-learning, they see their work changing and do not always have the necessary expertise and resources to take the e-learning route.

Indeed e-learning revolutionizes the training domain as it provides personalized, real-time training and communications programs, rather than a “one size fits all” approach. To take advantage of the potential of e-learning, multimedia publishers need to develop expertise in Internet technologies in general. To be able to put their content on-line they need to work in partnership with companies that have Internet expertise and document processing technologies. On the other hand, technology providers in the area of e-learning have given very little attention to the quality of the content that will be offered, to the pedagogical aspect of e-learning as well as to the question of what needs to be changed in order to use the Internet to train people. Most of the e-learning companies have tried to sell platforms to content providers rather than trying to work in partnership with them in order to create good-quality e-learning solutions. It is certainly not enough to integrate content that has been built for another purpose (to teach in face to face meetings for example, or even for CD-ROM distribution) with technologies to make a good training module. Pedagogical aspects must be considered and technologies should support pedagogy rather than being the main objective.

4.1 eContent adaptation

Thetis particularly concentrated on defining the virtual environment and the content of the training, adapted to the needs of the employees in a hotel reception.

The main functions of receptionists are presenting a positive image of the hotel and assisting with all aspects of guest service; they act as buffers between the customers and management of the hotel.

The training needed to take into account their main duties, including the following:

Answering enquiries regarding hotel services and registration by letter, telephone and in person
Making room reservations
Registering arriving guest and assigning rooms
Responding to guests’ enquiries, requests and complaints
Using computerized or manual systems to compile and check daily record sheets, guest accounts, receipts and vouchers
Presenting statements of charges to departing guests and receiving payment
Additional duties, which may be required by small hotels.

It was also important to consider the skills needed by these kinds of workers, even when they need to communicate with customers in languages different from theirs.

Good communication skills and a neat appearance are essential for hotel receptionists. They need basic analytical skills and experience with word processing equipment and computers. Good judgment is necessary, as well as ability to solve problems in a bold way and with determination, since they may have to deal with difficult people, as well as with emergency and security problems.

But one of the best assets is the knowledge of, at least one, second language. To communicate fluently with customers in a foreign language, solve problems and deal with complaints, turn them into opportunities is the aim of this project, together with the development of key competences for employees working at hotel reception desks, such as efficiency, courtesy, initiative, capability of working in team and communication skills, among others.

What is different in Thetis, compared to "traditional CD-ROM content" building are the scenarios through which the courses will be delivered to hotel receptionists. So the content had to be appropriate to the scenarios so that it is adapted to the learning context, and it had to be localized following the target region of the training.

The pedagogical content is realized on different media like texts, audio support, and videotapes. For each medium different activities are available in order that the students can practice a specific grammar point, an idiomatic structure or phrase. There are several kinds of exercises each corresponding to a lexical area and/or speech act. This means that a student is immersed in a work situation associated with specific difficulties in grammar and vocabulary and (s)he is provided with exercises helping to develop her/his competencies.

The objective is not to find a typical situation in order to learn a specific grammar point or an idiomatic structure. The process is rather first to list the typical work situations the hotel receptionists are confronted with and afterwards to identify the language difficulties associated with each situation and, in a third time, to select the media, the content, the type of activity that is the best suited to help the students to deal with the work situation. One other difficulty is to propose a sequence of working activities that also match with the language and vocabulary level of the student.

Besides the scenario, the students are also provided with some classical content in order to help them to meet the needs they could be confronted with outside typical training situations. Such classical content is listed in the points below:

- a grammar book containing a list of all the sentence patterns and forms that are used in the program with explanations of their use and form;
- a lexicon with English headwords, examples of usage and explanation or translation into native language. Students also have the option of hearing the pronunciation of new items;
- student records which allow students to monitor their own progress,
- technical help in order to present the characteristics of the delivery web application.

This existing content has been adapted to the tourism industry

Although the basic content will be the same for all target audiences, the addition of elements that appeal directly to each target user group makes the product attractive and motivating.

In terms of language practice, it is also possible to place an emphasis on those items that cause particular problems to specific language groups.

As far as already existing content related to tourism scenarios is concerned, they had to be adapted to our purposes: some pedagogical activities had to be reversed because they had been developed in order to help a customer, a tourist in our case, to interact with other people i.e. receptionists, waiters/waitresses, passers-by. The main task was to propose mainly the same activities but for training receptionists.

5 Users' tests

The content of the course focuses on teaching English to hotel receptionists. Therefore, the users were faced with real situations that occur in hotels within the context of their everyday work environment. The objective then was to test the course within this target group in order to analyse its potential use, and possible adaptation to any socio-professional domain, as well as extension to any language.

The main elements of THETIS that needed to be tested are the following:

- Technical issues: linguistic tools, virtual reality

- Pedagogical effectiveness: quality of content, game aspects, etc.
- Work related competencies: hotel environment and situations, level of interest, relevance, etc.

There were two major groups of users involved in these tests:

- Students
- Teachers or tutors.

All students involved (18) are currently students of a Hotel Management vocational training program organised by the Andalusian Entrepreneurial Confederation (CEA).

The profile of the group was the following: 80% were between 20 and 24 years old, 20% were between 25 and 30 years old. All of them have a university degree in Tourism. 80% had previous work experience in hotels or tourism related jobs. 50% had an intermediate competency level in English, 17% low intermediate and 33% high intermediate.

A team of 3 teachers was also involved in this test; one of them was present at the location of the test activities and the other two were doing participated from another country, which shows the flexibility that distance learning offers.

Each session followed the same structure:

- Presentation of the three partners.
- Presentation of the THETIS project.
- Goals and objectives of the session.
- Explanation of the different elements of THETIS:
    - Chat
    - Virtual Reality
    - Dictionary
    - Verb conjugation
    - History
    - Notes
    - Grammar
    - Exercises

After the initial explanations, the session continued by proposing different activities in which the students had to do the following:

- **Chatting among students - learning games**. A reservation form had to be filled out. Each form contained some information of different customers. We assigned one specific guest to each student, so they had to discuss and ask each other about the missing data.
- **Chatting with the teacher - role playing exercises**. The teacher acted as a customer and started different dialogs with the students based on typical hotel situations such as making reservations or asking for directions.
- **Exercises**. The students were given time to practise four categories of exercises: reading comprehension, grammar, listening and vocabulary.
- **Interacting with the robots**. The students had to take part of the pre-set dialogs with the customer-robots at the virtual reality scenario and choose the right answer in every situation.

Once the testing activities described were finished, the students were asked to write their comments by answering a complete questionnaire

Most of the students agreed that using this program continuously as part of complete language training would be very beneficial. Some students appreciated most that it helped remember expressions and vocabulary they had learnt before, so it was good practice. Most of them acknowledged the acquisition of the specific language used in hotels and the expressions needed in situations that occur at the front desk.

Regarding the scenario itself all students considered that what happens in a hotel and the daily functions that a receptionist must carry out are very well represented. Some typical appreciations are the following:

"It is a very creative way to reflect the different situations happening in a hotel"

"Doing different things at the same time is very common at hotels and getting to learn English by having the same experience is crucial"

"It is good that we get to see the nice things that normally happen but also the not so nice ones, like dealing with upset customers"

Basically, the main advantages of THETIS appear to be how realistic the situations are, the entertaining and fun part of learning and the interaction with the teacher and the other students.

As for the weaknesses, these appear to be related to the fact that Thetis is still a pilot product and some technical and content details have yet to be finalized.

Regarding the elements the testers found most useful during the learning process, we found a broad range of answers and every tool was named. Regarding the linguistic technologies students mentioned the contextual dictionary look-up as the one they made the most use of. However, the elements that received a higher appreciation were the interaction with robots (virtual reality) and with the teacher (chat), because of the entertaining aspect they both bring, the realistic feature they portray as well as their practical application.

In order to have a complete and enriched information report, we asked for the teachers' opinion as well.

It was perceived that through this program students are given different ways to consolidate what they learn which makes it quite efficient. It is also important to say that there is a clear need to have teacher-student interactions to make it more successful pedagogically, since this element gives the students the opportunity to use free and authentic language and to be corrected in real time.

Since interactions between the teacher and the students are allowed through chat - which is something the students enjoyed a lot - it was remarked that the teacher needs to have the possibility to correct mistakes. So for this tool to be more useful and not damaging the role-playing situation, there is a need to find a way to send the students that feedback, for example, by making it appear in another color.

It is also important to prepare different possible scenarios/exercises for the teachers to develop through the chat. Therefore, it appears important to prepare a teacher's guide.

6 Conclusion: lessons learned

In general, we found that the creation of content for a Thetis like type of concept was not a trivial and easy task. Indeed Thetis provides a solution that is strongly web oriented in the sense that it insists on interaction on the web, personalization and information access. As a consequence the type of content that fits these requirements is radically different from ones that already exist for other supports such as, for instance, CD-ROMs. We did not expect this adaptation to be so time consuming. It turns out that while a "traditional" type of CD-ROM content can be integrated in the form of exercises within the Thetis solution, it is necessary to build new types of content in the form of scenarios. Indeed the notion of a scenario, which is central to Thetis as well as its strong point, means one that covers both learning a language and learning a work practice. This scenario should also be attractive and evolving enough so that it retains students' interest and motivation. The borderline is sometimes difficult: we need to be careful not only to propose games where students learn how to do their work but not so much how to speak a language. Moreover, the scenarios, since they take place on the web and in a virtual world, cannot be like the ones that are proposed in a face to face course. Interaction is different. Robots plays a central role in pushing people to interact with each other, and so does the tutor. How to make the best pedagogical use of these two aspects? What is the type of dialogues that work better in terms of pedagogical purposes?

During the testing we had had various comments on the specific features offered by the system as well as suggestions for improving it; all of them would need to be further tested. These suggestions come from different perspectives: from the tutors and from the students.

As far as the tutors are concerned, while they find the idea really interesting pedagogically, they also generally consider it difficult to interact with several students at the same time. For instance correcting several students in the chat is almost impossible. Therefore they ask for tools to help them in this process. These tools would be like the ones that already exist in current management systems courses, and they would have to be enriched with some specific functions to deal with on-line interaction. Teachers usually liked a lot the language guesser module as it forces students to interact in the language they learn and relieve them from checking that students do not interact in their own language.

On the opposite side, the students liked and used a lot the contextual on-line dictionary because they really want to play and interact and are forced to do so in the foreign language. The students definitively liked the virtual reality and the game aspect. The strongest point of Thetis is undoubtedly the interactions based on a virtual reality scenario as compared to e-mail interactions, chats or forums. Probably owing to the virtual world shy people tend to participate much more than usual. This is also the feedback that we got from tutors concerning their students.

While students enjoyed the game very much, this would need to be tested on the long run as Thetis certainly benefits from the “novelty effect”.

Both teachers and students would like to be able to speak rather than just to write. However, it is not clear how this type of interaction would be possible in virtual reality with 15 people not being able to see each other. During the project, we identified strong points as well as difficulties and gaps. The main difficulty is in the type of content that fits Thetis’ philosophy. Indeed, building such content requires more time than we first expected. While part of the content can be just repurposed and localized from already existing materials, a completely new type of content needs also to be created from scratch. Users made interesting suggestions regarding the integration of new technologies such as speech processing or tools for the tutors. These would require further testing.

References

Brun C., Parmentier T., Sandor A., Segond F., 2002, Les outils de TAL au service de la e-formation in *Multilinguisme et le traitement de l'information*, Frédérique Segond Ed., Hermès, pp. 223-251

Frase, L., R. Almond, J. Burstein, K. Kukich, R. Mislevy, K. Sheehan, L. Steinberg, and K. Singley and M. Chodorow. (2002). In H. O'Neill and R. Perez (Eds.) *Technology Applications In Education: A Learning View*. Lawrence Erlbaum

Kashny M., *Les usages des Technologies d'Information et de Communication par des enseignants dans un dispositif de formation tutorée en langues vivantes étrangères. Une approche ergonomique*. Thèse Université Pierre Mendès France, 2001, Grenoble.

Kearns M., Satinder Singh C., Litman D., and Howe.J. CobotDS, 2002: A Spoken Dialogue System for Chat. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, Edmonton, Canada.

Kindley, R., *Scenario-Based E-Learning: A Step Beyond Traditional E-Learning*, http://www.learningcircuits.com/2002/may2002/kindley.html

Kukich, K , 1992, Technique for automatically correcting words in text, *ACM Computing Surveys (CSUR)* Volume 24 , Issue 4, (377 – 439)

Paulsen, J. (2001-section 4). "Authentic Online Target Language Reference Resources." New Era Trends and Technologies in Foreign Language Learning: An Annotated Bibliography - *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, April 2001, Wake Forest University.

Puren, C. *Histoire des méthodologies de l'enseignement des langues* , Clé International, Nathan, 1988, Paris.

Segond F, Parmentier T, 2004 NLP serving the cause of language learning *Proceeding of International Workshop on eLearning for Computational Linguistics and Computational Linguistics for eLearning* COLING , Geneva

Author Index