WordGap - Automatic Generation of Gap-Filling Vocabulary Exercises for Mobile Learning

Susanne Knoop, Sabrina Wilske

University of Bremen
Bremen, Germany

susanne@informatik.uni-bremen.de, wilske@informatik.uni-bremen.de

Abstract

We present a mobile application for learners of English as a second language that instantaneously generates gap filling exercises from a given text. The app provides an opportunity for contextualized vocabulary learning, customized to the learner's interest. Part of the exercise is a multiple choice of the original gap filler plus a set of incorrect distractor items. The key problem to solve in order to automatically generate this type of exercises is the selection of suitable distractor items. For the implementation of the application, we employ strategies proposed in previous work, making use of freely available tools and resources.

Keywords: Vocabulary, ESL, NLP, CALL, cloze exercises, gap filling, Android, informal language learning, mobile learning.

1 Introduction

In recent years, the rising popularity of smartphones has led to an increase in mobile applications for vocabulary learning. Most of these applications help the user to memorize predefined word lists with the help of digital index cards. However, index cards present a word in an isolated way that does not reflect the incremental and complex process of vocabulary acquisition. Several authors therefore recommend learning from context through extensive reading as the most effective way of vocabulary acquisition (Nerbonne, 2002; Nation and Waring, 1997; Oxford and Scarcella, 1994; Nagy et al., 1985).

A widely used method to test and train word knowledge in context is the "cloze exercise". It consists of a text in which words have been replaced by a gap that has to be filled by the learner (Lee, 2008; Soudek and Soudek, 1983). In the multiple-choice version of the cloze exercise the target word is presented together with several incorrect candidates, called distractors. The quality of a multiple choice cloze exercise is highly dependent on the quality of the distractors. Good distractors should not be totally unlikely, but at the same time not too similar to the target response. If they are too implausible, the exercise would be too easy; if they are too similar, the exercise would not have a clear, unambiguous target response.

In particular if cloze tests are used for language testing, the quality of the distractors is related to (a) their capacity to distract from the correct answer and (b) their ability to discern between learners of different proficiency levels (Goodrich, 1977). Based on an empirical study with Arab learners of English Goodrich recommends words from the context of the text, antonyms, and false synonyms (i.e., synonyms that can not replace the target word in this specific context) as effective distractors for English cloze exercises.

Since it usually requires expert knowledge to select suitable distractors, the creation of individual exercises, based on up-to-date material is expensive and time-consuming (Sumita et al., 2005). In order to avoid the cost of the creation of suitable cloze exercises, there have been several attempts in recent years to automatically generate good distractors using methods from natural language processing (NLP) (Coniam, 1997; Brown et al., 2005; Sumita et al., 2005).

Using the insights of this research and readily available NLP tools, we have built a mobile application for Android smartphones that creates cloze exercises on the fly, based on texts selected by the learner. The application thus provides highly individualized exercises to support contextualized mobile vocabulary learning. Customized reading material that matches the interests of the learner increases the motivation to learn and facilitates learning progress (Heilman et al., 2010; Goto et al., 2010).

In the remainder of the paper we first summarize the strategies used in previous work for selecting suitable distractors (Section 2). We then give an overview of the application in Section 3. Section 4 describes processing steps of the application and the NLP tools that we used for the implementation. In Section 5 we give an outlook on possible further development of the application.

2 Related Work

In this section we describe previous work on generating multiple-choice cloze exercises. As we have discussed above, the challenge in multiple-choice exercises is to select appropriate distractors to the correct target response. The strategies that have been shown to be successful make use of parts of speech, frequency, and distribution of words in the text to select distractors.

One of the first approaches is described by Coniam (1997), who developed a system for the automatic generation of multiple-choice cloze exercises to assist ESL teachers of secondary schools with the preparation of tests. The program chooses distractors with the same part of speech and a similar frequency in the Bank of English Corpus as the target word.

The REAP system, implemented by Brown et al. (2005), develops an individual learner model for each user that encompasses the user's vocabulary and personal interests and chooses a reading text with a distribution of 95% known vocabulary and 5% unknown vocabulary. The knowledge of the newly learned 5% words is then trained and tested through automatically generated vocabulary tests. Afterwards, the program updates the learner model and chooses a new text. The distractors of the automatic multiple choice cloze tests have the same part of speech tag and a similar frequency as the target word and distractors that appear in the original text are preferred. The performance of learners in the automatically generated tests correlates strongly with their performance in manually created tests as well as with the results of standard vocabulary tests.

Sumita et al. (2005) use a corpus of manually created cloze exercises to determine the features of adequate target words, such as their position in the sentence and their part of speech. Distractors are synonyms of the target word that are verified with a Google search: If the sentence together with a distractor candidate yields results, it can be assumed that in this sentence, the distractor could be a valid replacement of the target word and is therefore deleted from the list. Again, the results of the automatically generated tests correlate strongly with the results of the internationally recognized TOEIC (Test of English for International Communication).

Of these examples, Sumita et al. (2005) come closest to the application we present. Like our approach, exercises can be generated based on individually selected web pages. There is also a mobile interface which allows users to download existing exercises stored on a server to a mobile phone. Unlike our application, which focuses on informal, self-paced learning, the system described by Sumita et al. is targeted primarily on language proficiency testing.

3 WordGap: Automatic cloze exercises for smartphone users

In the following, we describe WordGap, an application for the Android platform that allows learners of English to test and train their word knowledge with cloze exercises from any text file or website. The source code of the app and of the server component was published under a GNU General Public License (GPL).[1]

The target group for WordGap are adult and advanced learners of English as a second language who seek to extend their vocabulary by reading texts of personal interest, for example novels, newspaper articles or blog entries. In the context of the mobile application, even short waiting times of 5 to 10 minutes can be used for a short exercise.

The application is of best use to advanced learners because a certain amount of vocabulary has to be known to allow learning new vocabulary from context. Nation and Waring (1997) cite the number of 3,000 words that cover 95% of a text as most efficient for contextual

[1] `https://github.com/wordgap/wordgap`

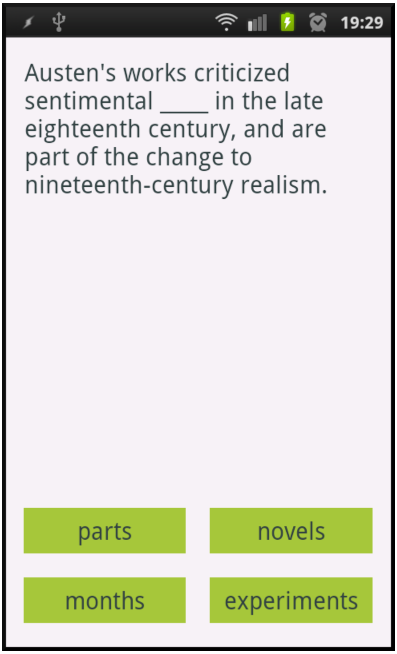

Figure 1: Screenshot of a cloze exercise with the WordGap app

learning. Less advanced learners, however, could use simplified texts, such as children's and youth literature. Unknown vocabulary can only be guessed if it is sufficiently common, therefore, WordGap is not intendend for the acquisition of terminology, entity names or extremely rare words.

The chosen text can be loaded from a text file saved on the smartphone or from a website. In the latter case, the app will load the website's text when the user selects the app on the "Sent-to" menu of the smartphone's browser. Exercises can be generated for four different parts of speech: nouns, verbs, adjectives or prepositions. The app sends the text and the chosen part of speech to a server that generates the exercise for the learner to carry out on their phone.

WordGap displays the sentences of the exercise sequentially together with the target word and three distractors in random order (see Figure 1 for a screen shot). The user has to choose the correct answer by tapping on it. Correct and incorrect choices are logged for the user's performance statistics that will be displayed after completing or aborting the exercise.

During the exercise, unknown target words can be added to a list and after finishing the exercise, the WordNet definitions of the unknown words can be obtained from the server. This delayed access to the word definitions is motivated by the Depth-of-Processing hypothesis: Guessing the meaning of an unknown word from the context requires a deeper semantic processing than simply looking it up in the dictionary and is therefore supposed to ease long-term memorization (Nerbonne, 2002; Oxford and Scarcella, 1994; Segler et al., 2002). All exercises are saved automatically on the phone's local memory so that they can be repeated at any time even without network connection.

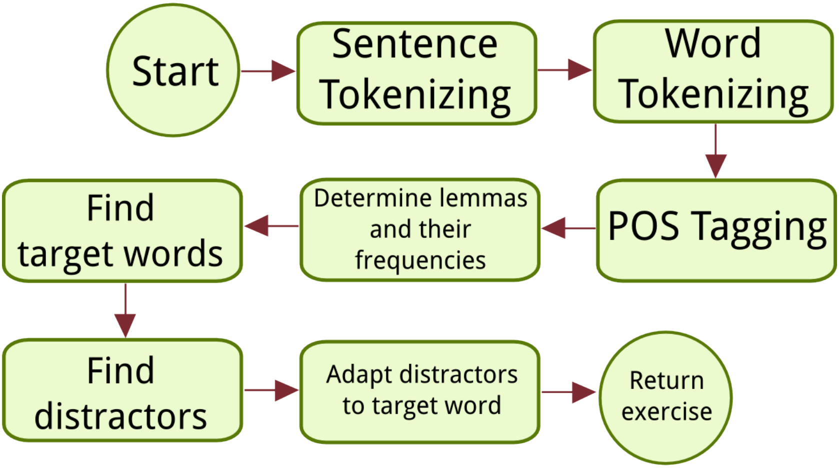

Figure 2: The NLP processing pipeline of the WordGap server

4 System architecture and NLP tools

This section describes the processing steps of the system and presents the tools and resources that we applied.

The implementation consists of a client application that runs on Android devices and a server implementation that can run on any machine. The server implemenation is based on the Django Python web framework[2]. The server and the clients communicate through the JSON data interchange format[3].

When creating an exercise, the WordGap server processes the text according to the pipeline demonstrated in Figure 2. After sentence and word tokenizing, the text is part-of-speech tagged. The tokens tagged with the part of speech chosen by the user (i.e., nouns, verbs, adjectives or prepositions) are then transformed into their base form (i. e., infinitive for verbs and singular for nouns) to determine their frequency. For each sentence, the program attempts to find a target word of the given part of speech tag. If there are multiple candidates in one sentence, the target word is chosen based on its frequency in the text – the more frequent a word, the more likely it is to be selected by a weighted random choice function.

For each target word, adequate distractors have to be determined. These can be words with the same part of speech from the text, or antonyms and false synonyms of the target word. In a post-processing step, they are adapted to the target word in their grammatical form and capitalization to avoid giving the user additional hints about the correct choice.

The following subsections describe how several Natural Language Processing resources support the process of exercise generation.

4.1 WordNet

WordNet is a semantic network that was developed by George Miller and Christiane Fellbaum to study the vocabulary acquisition of infants (Miller and Fellbaum, 2007). It organizes the

[2] www.djangoproject.com
[3] www.json.org

nouns, adjectives, adverbs and verbs of the English language into so-called “synsets” that contain lemmas that can be synonyms in certain contexts. Nouns and verbs are also ordered in a hierarchy of hypernyms and hyponyms.

WordGap uses WordNet to find distractors that are antonyms or false synonyms of the target word. For the latter, we implemented two possibilities: Words with the same hypernym as the target word or synonyms of the synonyms.

A visual and informal inspection of the distractors showed that those taken directly from the text tend to be more useful than the distractors derived from WordNet. One reason for that is that WordNet often lacks entries for antonyms and synonyms that a thesaurus would contain. Besides, many synsets contain rare words that will seem out of place because they do not fit to the text’s genre. The user could in this case just guess the right word by choosing the only word that seems familiar. Because of this shortcoming, the user is offered the option to use or not use WordNet when creating the exercise.

4.2 The Natural Language Tool Kit (NTLK)

The Natural Language Tool Kit (NTLK) is an extensive open-source library for the programming language Python that was first developed by Edward Loper and Ewan Klein at the University of Pennsylvania (Perkins, 2010; Bird et al., 2009). It contains WordNet as well as numerous corpora and dictionaries in different languages.

The WordGap server uses methods of the NLTK for sentence and word tokenizing, as well as part of speech tagging. For the latter, NLTK’s implementation of the Naive Bayes Tagger was trained on the Brown corpus. For the WordGap application we want to value precision over recall because an incorrectly tagged target word would lead to inadequate distractors. Therefore, we trained the Naive Bayes tagger with a cut-off probability of 95%, which means that no token will be tagged with a tag that has a probability of less than 95%.

4.3 Nodebox Linguistics

NodeBox Linguistics[4] is a collection of different open-source libraries for Python. First of all, it contains a more convenient interface for accessing WordNet than the NLTK does. More importantly, it provides methods for generating different grammatical forms of English lemmas. Thus it offers singular and plural forms of nouns and different tenses for verbs. WordGap uses these methods to adapt the distractors to the target word and make their number and/or tense match.

5 Conclusion and Outlook

We have presented an application for smartphones that generates instantaneous cloze exercises based on texts chosen by the user and thus provides contextualized and individualized vocabulary learning. We have shown that it is possible to create such an application with readily available NLP tools and resources.

One possible extension of WordGap would be the adaption to a target language other than English. This would require that the following resources are available in that language: A software library for sentence tokenizing, word tokenizing and part of speech tagging, a

[4] `nodebox.net/code/index.php/Linguistics`

semantic network like WordNet or a thesaurus with synonyms and antonyms, a database or software library to retrieve different grammatical forms of nouns, verbs or adjectives and a source of definitions or translations of unkown target words.

Another possible extension to the app would be multiple choice cloze exercises that focus on grammatical knowledge, similar to the work described by Meurers et al. (2010). For instance, for learning verb tenses, the distractors could be different tense forms of the same verb. For learning the use of articles, the learner would have to choose from a list of definite, indefinite and no article.

So far, the application has only been tested in terms of usability as part of the development process to identify usability issues. In future work, we would like to evaluate the app in terms of the learning gains that it enables. We also have not yet conducted a thorough assessment of the quality of the generated exercises.

References

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly & Associates, Inc., Sebastopol, CA, USA.

Brown, J., Frishkoff, G. A., and Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In *HLT/EMNLP*. The Association for Computational Linguistics.

Coniam, D. (1997). A Preliminary Inquiry Into Using Corpus Word Frequency Data in the Automatic Generation of English Language Cloze Tests. *CALICO Journal*, 14(2-4):15–33.

Goodrich, H. C. (1977). Distractor efficiency in foreign language testing. *TESOL Quarterly*, 11(1):pp. 69–78.

Goto, T., Kojiri, T., Watanabe, T., Iwata, T., and Yamada, T. (2010). Automatic generation system of multiple-choice cloze questions and its evaluation. *Knowledge Management & E-Learning : an International Journal*.

Heilman, M., Collins-Thompson, K., Callan, J., Eskenazi, M., Juffs, A., and Wilson, L. (2010). Personalization of reading passages improves vocabulary acquisition. *Int. J. Artif. Intell. Ed.*, 20(1):73–98.

Lee, S. H. (2008). Beyond reading and proficiency assessment: The rational cloze procedure as stimulus for integrated reading, writing, and vocabulary instruction and teacher–student interaction in esl. *System*, 36(4):642 – 660.

Meurers, D., Ziai, R., Amaral, L., Boyd, A., Dimitrov, A., Metcalf, V., Ott, N., and Tübingen, U. (2010). Enhancing authentic web pages for language learners.

Miller, G. A. and Fellbaum, C. (2007). Wordnet then and now. *Language Resources and Evaluation*, 41(2):pp. 209–214.

Nagy, W. E., Herman, P. A., and Anderson, R. C. (1985). Learning Words from Context. *Reading Research Quarterly*, 20(2):233–253.

Nation, P. and Waring, R. (1997). Vocabulary size, text coverage and word lists. In *Vocabulary: Description, Acquisition and Pedagogy*, pages 6–19. University Press.

Nerbonne, J. (2002). Computer-assisted language learning and natural language processing. In *Handbook of Computational Linguistics*, pages 670–698. University Press.

Oxford, R. and Scarcella, R. C. (1994). Second language vocabulary learning among adults: State of the art in vocabulary instruction. *System*, 22(2):231–243.

Perkins, J. (2010). *Python Text Processing with NLTK 2.0 Cookbook : Over 80 Practical Recipes for Using Python's NLTK Suite of Libraries to Maximize Your Natural Language Processing Capabilities*. Packt, Birmingham, UK.

Segler, T. M., Pain, H., and Sorace, A. (2002). Second Language Vocabulary Acquisition and Learning Strategies in ICALL Environments. *Computer Assisted Language Learning*, 15(4):409–422.

Soudek, M. and Soudek, L. I. (1983). Cloze after thirty years: new uses in language teaching. *ELT Journal*, 37(4):335–340.

Sumita, E., Sugaya, F., and Yamamoto, S. (2005). Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, EdAppsNLP 05, pages 61–68, Stroudsburg, PA, USA. Association for Computational Linguistics.