TextMix: using NLP and APIs to generate chunked sentence scramble tasks

Brendon Albertson[1]

Abstract. A Computer-Assisted Language Learning (CALL) application, TextMix, was developed as a proof-of-concept for applying Natural Language Processing (NLP) sentence chunking techniques to creating 'sentence scramble' learning tasks. TextMix addresses limitations of existing applications for creating sentence scrambles by using NLP to parse and scramble syntactic components of sentences, while connecting with Application Programming Interfaces (APIs) to provide repeated exposure to authentic sentences in the context of texts such as Wikipedia articles. In addition to identifying a novel application of NLP and APIs in CALL, this project highlights the need for teacher-friendly interfaces that prioritize pedagogically useful ways of chunking text.

Keywords: NLP, chunking, collocations, syntactic awareness.

1. Introduction

This paper describes the rationale, development, and implications of a CALL application, TextMix. It was developed on several premises. First, it served to explore the feasibility of using NLP sentence chunking techniques to generate 'sentence scramble' activities as a method for enhancing input to raise syntactic and collocational awareness. Second, it sought to demonstrate the viability of using APIs to import authentic online text, such as Wikipedia articles and news headlines, for generating learning activities. TextMix aims to serve as a model for developing other CALL tools that use such features.

1. Pine Manor College, Chestnut Hill, USA; albertsonbrendon@gmail.com; https://orcid.org/0000-0001-8932-835X

How to cite this article: Albertson, B. (2021). TextMix: using NLP and APIs to generate chunked sentence scramble tasks. In N. Zoghlami, C. Brudermann, C. Sarré, M. Grosbois, L. Bradley, & S. Thouësny (Eds), *CALL and professionalisation: short papers from EUROCALL 2021* (pp. 6-11). Research-publishing.net. https://doi.org/10.14705/rpnet.2021.54.1300

2. Background

2.1. Input enhancement and chunking

Textual input enhancement involves methods for drawing attention to target language such as underlining vocabulary or highlighting grammar structures. It aims to promote noticing as a prerequisite for learning and can be effective in tandem with other learning activities (Kim, 2010). Specifically, drawing attention to formulaic sequences may compensate for impoverished input and improve retention (Nguyen, 2014), while syntactic highlighting has been correlated with higher reading scores among low-proficiency learners (Park & Warschauer, 2016).

Chunking, or dividing, sentences can also serve as a form of input enhancement. Chunking complex sentences into clauses, for example, may help learners process their structure, while smaller chunks like adjective-preposition pairs can raise awareness of collocations such as *interested in*. Eye-tracking research by Pulido (2021) has demonstrated that L2 readers who chunked while reading were more efficient readers. However, chunking alone is not likely to improve comprehension; it is necessary to accurately connect and parse the relationships between chunks to form meaning (Nishida, 2013). A task that involves assembling chunks in the correct order would promote focusing on such semantic relationships. Chunking can be performed using NLP algorithms based on parts of speech. While NLP has several educational applications (Litman, 2016), chunking methods in NLP have not been examined for their potential to teach sentence structure or collocations.

2.2. The sentence scramble task

The sentence scramble task involves arranging the mixed parts of a sentence into the correct order. By requiring a focus on word order, sentence scrambles may improve understanding of sentence structure and noticing of grammar features (Murasawa & Brine, 2010), and provide an accurate measure of syntactic awareness (Chu & Ellefson, 2020). They also represent what Bjork (1994) terms a 'desirable difficulty', an additional processing demand that can aid learning.

An existing CALL system for generating sentence scrambles is FLAX (Murasawa & Brine, 2010), which rearranges several target words (e.g. prepositions) in each

sentence. The only other existing CALL tool for creating sentence scrambles is the J-Mix feature of the *Hot Potatoes* software suite (Half-baked Software, 2020), which requires manual sentence entry and scrambles either by every word or manually specified divisions. These tools have potential for enhancement by scrambling via chunks rather than words, connecting with larger sources of text, and making it easier to generate and share activities.

2.3. Design question

The design of the TextMix application was driven by three areas with potential: (1) chunking as a method for raising awareness of sentence structure and collocations, (2) the sentence scramble as a way for learners to work with chunks, and (3) the usefulness of applying NLP chunking techniques and large sources of online text to CALL applications.

The following three-part design question was posed. Can a web-based CALL tool:

- be designed to scramble sentences by meaningful chunks instead of words using NLP;
- be made compatible with large, existing sources of text including APIs;
- enable saving and sharing the generated activities via URL?

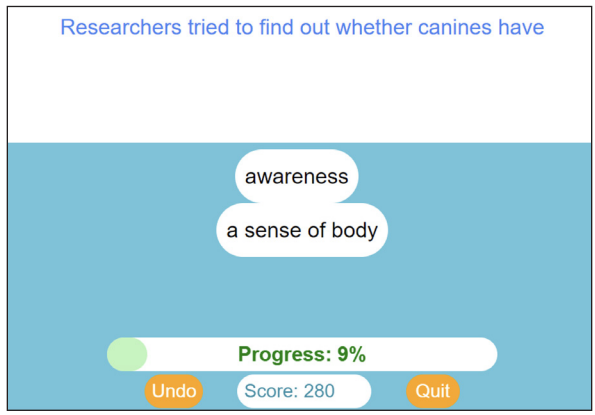
3. Design

3.1. The TextMix application

The online TextMix application generates a sentence scramble for each sentence in a text, performed by dividing the sentence into chunks via NLP and randomly reordering them (Figure 1).

When one sentence is unscrambled by the user, the application proceeds to the next. The source of text is specified by the user; options include news headlines via the News API, Wikipedia articles via the Wikimedia API, pasted text, a preloaded collection of open-source texts, and example dictionary sentences. Users can choose whether chunks should be combined to make the activity less difficult and can generate a URL to the saved activity.

Figure 1. A sentence scramble activity in TextMix



3.2. Chunking in TextMix

TextMix uses the Python Natural Language Toolkit, a software library that chunks sentences via a rule-based method using adjustable definitions for each type of chunk. The algorithm is not completely accurate and must prioritize one chunk type over another. For example, if a chunk is defined as a verb plus preposition, it would capture phrasal verbs but also other combinations (e.g. *think about*). Conversely, defining another type of chunk as a prepositional phrase would capture *about the decision* but would disrupt the former chunk type from capturing *think about*. Thus, the algorithm's usefulness depends on defining chunks appropriately for different learning focuses.

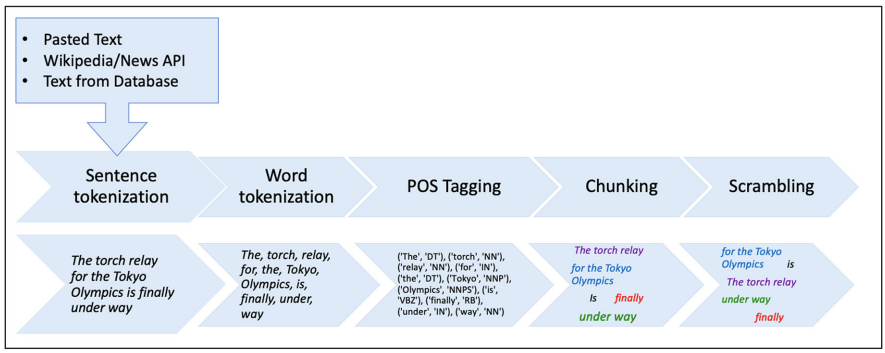
For this project, two types of chunks were assumed most useful to learners: collocations and meaningful syntactic units such as noun phrases or compound verbs. In some cases, a chunk would represent both. Consequently, six chunk types were defined: noun phrases, verb phrases, prepositional phrases, compound verbs, infinitive phrases, and relative clauses. These were programmed using the definitions shown in Table 1. The NLP flow of the application is shown in Figure 2.

Table 1. Definitions of chunks

Type of chunk	Definition
Prepositional or noun phrase	P* (DET) ADJ* N+
Compound verb or infinitive	(to) V+
Relative clause	RP V+

Note: Plus sign = 'one or more'; asterisk = 'zero or more'; parentheses = 'zero or one'

Figure 2. TextMix programmatic flow



4. Discussion

TextMix responds to the design question by automating the creation of chunk-based sentence scramble tasks using NLP and by importing sentences from APIs, while allowing users to save and share activities via URLs. The project revealed two implications of using NLP chunking in CALL. First, due to overlap, not every meaningful chunk can be captured at once, and choosing which chunks to prioritize depends on the learning focus and level. For example, chunking sentences into subject-predicate pairs may be suitable for beginners learning basic sentence structure, while chunking smaller units such as noun phrases and prepositional phrases would be more appropriate for learners with greater syntactic awareness. Adjusting the chunking method, however, is not quite teacher-friendly as it requires changing regular expression definitions in Python code. A possible solution is to provide predefined options for which chunks to prioritize, such as phrasal verbs. Another possibility is to directly identify and extract collocations as chunks using lists such as the Academic Collocations List (Lei & Liu, 2018). Second, because traditional NLP chunking algorithms aim to extract only semantic data from text, they may not be ideal for creating syntax or grammar-focused learning tasks. A teacher might wish, for example, to focus on infinitives; for these purposes new chunking algorithms must be defined.

5. Conclusion

TextMix demonstrates the feasibility of applying NLP chunking techniques and APIs to CALL by separating and drawing attention to meaningful units of sentences drawn from online text sources. The sentence scramble tasks may help

raise syntactic awareness and sentence processing ability by requiring learners to analyze the relationships between chunks. Furthermore, they may promote collocational awareness by drawing attention to words commonly found together. However, these possible learning benefits still require assessment. Finally, adjusting the learning focus by devising additional chunking methods and making these accessible to teachers remain areas worth exploring.

References

Bjork, R. A. (1994). Memory and metamemory considerations in the training of human beings. In J. Metcalfe & A. Shimamura (Eds), *Metacognition: knowing about knowing* (pp. 185-205). MIT Press. https://doi.org/10.7551/mitpress/4561.003.0011

Chu, C. P., & Ellefson, M. (2020). The development of a syntactic awareness task using the word-order correction paradigm. Cambridge Open Engage. https://doi.org/10.33774/coe-2020-5qdgd

Half-baked Software. (2020). *Hot potatoes* (Version 7.0) [Computer software]. http://hotpot.uvic.ca/

Kim, E. C. (2010). Textual input enhancement: applications in teaching. *ORTESOL Journal, 28*. https://ortesol.wildapricot.org/resources/Documents/Publications/Journals/2010/Textual%20Input%20Enhancement-%20Applications%20in%20Teaching%20p22.pdf

Lei, L., & Liu, D. (2018). The academic English collocation list: a corpus-driven study. *International Journal of Corpus Linguistics, 23*(2), 216-243. https://doi.org/10.1075/ijcl.16135.lei

Litman, D. (2016). Natural language processing for enhancing teaching and learning. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)* (pp. 4170-4176). Association for the Advancement of Artificial Intelligence.

Murasawa, F., & Brine, J. (2010). Focus-on-form using computer-generated scrambled sentences. *International Transactions on eLearning & Usability, 1*(1).

Nguyen, H. (2014). The acquisition of formulaic sequences in high-intermediate ESL learners. *Publicly Accessible Penn Dissertations*, 1385. http://repository.upenn.edu/edissertations/1385

Nishida, H. (2013). The influence of chunking on reading comprehension: investigating the acquisition of chunking skill. *Journal of Asia TEFL, 10*(4), 163-183.

Park, Y., & Warschauer, M. (2016). Syntactic enhancement and second language literacy: an experimental study. *Language Learning & Technology, 20*(3), 180-199. https://www.lltjournal.org/item/2974

Pulido, M. F. (2021). Individual chunking ability predicts efficient or shallow L2 processing: eye-tracking evidence from multiword units in relative clauses. *Frontiers in Psychology, 11*, 1-18. https://doi.org/10.3389/fpsyg.2020.607621

Published by Research-publishing.net, a not-for-profit association
Contact: info@research-publishing.net

CALL and professionalisation: short papers from EUROCALL 2021
Edited by Naouel Zoghlami, Cédric Brudermann, Cedric Sarré, Muriel Grosbois, Linda Bradley, and Sylvie Thouësny

Publication date: 2021/12/13

Typeset by Research-publishing.net

Cover Theme by © 2021 DIRCOM CNAM; Graphiste : Thomas Veniant
Cover Photo by © 2021 Léo Andres, Sorbonne Université
Cover Photo by © 2021 Sandrine Villain, Le Cnam
Cover Layout by © 2021 Raphaël Savina (raphael@savina.net)

Fonts used are licensed under a SIL Open Font License

ISBN13: 978-2-490057-97-9 (PDF, colour)

British Library Cataloguing-in-Publication Data.
A cataloguing record for this book is available from the British Library.

Legal deposit, France: Bibliothèque Nationale de France - Dépôt légal: décembre 2021.