Department of Computer Science

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Hamilton, New Zealand

Automating vocabulary tests and enriching online courses for language learners

by

Jemma König

This thesis is submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy in Computer Science
At The University of Waikato

September 2019

Abstract

The past decade has seen a massive growth in online academic courses, most of which are offered in the English language. However, although more people speak English as their second language than as their first, online course providers do not offer language assistance. Providing online learners with language resources would allow them to both learn about a subject through a foreign language and learn the foreign language through the subject. This is referred to as "content-based language learning".

Supporting content-based language learning using online courses raises several challenges, three of which are addressed in this thesis. First, courses teach subjects in particular domains, but supporting domain-specific language requires knowledge of specialised vocabulary. This thesis develops an automated approach to generating domain-specific corpora and wordlists, extracting domain-specific vocabulary in a way that can be applied to any online course.

Second, acquiring and measuring language come hand-in-hand. Tools that help learners acquire new language should also include methods for testing. This thesis takes an existing vocabulary test and automates it. This has two main advantages: it requires no assumed knowledge of the language, allowing automatic generation of vocabulary tests; and the tests reflect the wordlists used to create them, allowing them to be targeted toward a particular domain.

Finally, for content-based language learning to be used successfully, the language components must be smoothly integrated into courses without disturbing the original content. Furthermore, vocabulary support should include multi-word lexical items as well as single words. The thesis describes a tool that enhances online course content, via a browser extension. It is completely automated, though would also lend itself to selective teacher intervention. It is illustrated here with reference to courses offered by the FutureLearn MOOC consortium.

Acknowledgements

First, I would like to thank my chief supervisor, Ian Witten, without whom this PhD would not have been possible. I met Ian in the second year of my undergraduate studies, when I spent the summer developing language learning applications for him. Since then, I have worked with him over multiple summers, was lucky enough to have his insight as my honours supervisor, and now have the honour of his being my chief PhD supervisor. Thank you, Ian, for all of your guidance, support, and encouragement over the years. You have taught me how to focus on the nitty gritty, and how to look at the bigger picture; you have shown me the importance of hard work and perseverance, and shown me how important it is to take time out to enjoy life. Without you, I would not have become the researcher, developer, or woman that I am today.

I am also indebted to my other supervisors, Shaoqun Wu, Mark Apperley, and Andreea Calude. Shaoqun, thank you for your guidance, both in terms of the language learning portion of my PhD and the computational side of it. Knowing that I could come to you with any questions, whether they be collocation related or puzzling over telnet connections, has helped me immensely with progressing through this work. Mark, your advice and insight over the past few years have been invaluable. Thank you for always making the time to meet with me, particularly since it was usually with very little warning, and always providing indispensable guidance. The overarching structure of this PhD would not be what it is without your counsel. Andreea, thank you for your guidance through the (to me) once mysterious field of linguistics. I would not begin to claim that I am an expert in the field now, but the (reasonable) amount of linguistic knowledge I now possess can most certainly be attributed to you. Thank you for always providing guidance in this area and encouraging me to pursue it more, and of course, for the never ending supply of coffee, I most certainly would not have completed this PhD without that!

I am not aware of anyone else who was lucky enough to have four PhD supervisors, and I am eternally grateful. Each of you have supported me in so many ways during this time, whether it was field specific (language learning, software development, usability, linguistics, etc.) or simple moral support, I will forever appreciate the time you have each put into my personal and professional development.

A special thanks to Averil Coxhead, from the School of Linguistics and Language Studies at Victoria University of Wellington. Averil, I am so grateful for all of your support and guidance, for always making me feel welcome at VUW, and for introducing me to the amazing vocab group. I have thoroughly enjoyed collaborating with both you and Andreea on the pseudoword paper, and sincerely hope we can work together again in the future.

I would like to acknowledge the University of Waikato for funding this research through the generous Doctoral Scholarship, for funding travel to both domestic and international conferences, and for funding numerous trips down to Victoria University of Wellington.

Finally, I would like to thank my family for all their encouragement and support throughout this journey. First, to my mum, Karen, for being my sounding board and giving her unconditional love and support throughout, even when that meant allowing me to lock myself in a room for the entirety of my visit to see her. Thank you, mum, for always taking on my struggles, even on top of everything else. To my dad, Karl, who passed away before this process began, but that I know would be eternally proud of me, whether I accomplished this or not. To my sister, Dani, for always being my person, both before this process, during it, and after it, and for not just putting up with, but easing, my increasingly erratic moods, even if that meant a prod here and there. Thank you, soul sister, I couldn't have made it through this without you. To my niece, Bianca, for always being the bright light in my day, for lightening my mood and making me laugh, even throughout the struggles of writing, your love and maturity at such a young age continue to surprise me, and I am immensely proud of the woman you are becoming, thanks honey bunny. Finally, to my partner Grant, who came into this process at the most difficult time but showed immeasurable patience and understanding, thank you for all of your love and support over the past 10 months.

Contents

List of tables

List of figures

List of code examples

Chapter 1
Introduction

The World-Wide Web is an extraordinarily rich source of educational material. The past decade has seen a massive growth in online academic courses, many offered free of charge. This opens university-level study to new classes of learners, and many argue that it has the potential to revolutionize education (Kaplan & Haenlein, 2016). However, most online courses are offered in the English language, accessible only by learners with a working knowledge of general English, and even those learners often struggle. Yet online course providers do not offer language assistance. Yes, learners can use dictionaries, glossaries, and external language resources, but doing so would shift their focus away from course content and interrupt their learning. There is untapped potential here for using online courses for content-based language learning.

Content-based language learning is the dual concept of (i) learning a subject through a foreign language, and (ii) learning the foreign language through the subject (Marsh, 2002). Supporting content-based language learning raises three challenges. First, online courses provide subjects in particular domains, in keeping with content-based learning, but existing language resources focus on general-purpose vocabulary. Online course platforms provide a huge variety of courses, and the language can vary drastically between them. It is important to acknowledge that the language present in a particular domain is not the same as general-purpose language, and that domains differ from one another.

Second, each student's language growth can be measured using vocabulary tests, which could be integrated into online course platforms. However, vocabulary tests are generally not directed toward a particular subject or domain, but toward general-purpose language, yet the vocabulary that learners are exposed to for one course will not be the same for another. Measuring learners' vocabulary requires a test tailored to that domain, but creating different tests for each course would be time consuming and, if not done by an expert, likely unreliable.

Finally, content-based language learning is an integrated learning technique. For it to be successful, language teaching must be smoothly integrated into the course without disturbing the original content. Vocabulary is learned from natural contextualised language, with emphasis on the acquisition of both single word and multi-word lexical items. This raises the challenge of developing a system that integrates itself into online courses without disturbing the content, but while still providing sufficient language integration for content-based language learning.

1.1 Thesis statement

This thesis addresses three challenges that arise when using online courses for content-based language learning: identifying domain-specific vocabulary, automating a domain-specific vocabulary test, and integrating language resources into online courses.

1.1.1 Identifying domain-specific vocabulary

Identifying the vocabulary present in a domain is often achieved by creating a domain-specific wordlist, but, building a domain-specific wordlist first requires the collection of domain-specific text or corpora. This leads to the first challenge:

1. *Developing a set of automated processes that use online course content to build domain-specific corpora and wordlists.*

Not all vocabulary that exists in a language is present in a particular domain. Different domains contain different subsets of language. Determining the vocabulary present in a domain is often achieved by building corpora and analysing them to create domain-specific wordlists. However, although corpora are readily available for some domains, and for general-purpose language, other less common domain-based corpora are much harder, if not impossible, to find, and building new ones can be both difficult and time consuming.

In fact, the most time-consuming part of building a corpus is collecting and digitizing text. Collecting text from multiple sources in a variety of formats can take a considerable amount of time and financial resources. Instead, the World Wide Web has become a source for corpora content. Nevertheless, using the web as a corpus has its own limitations. It contains an ever-expanding amount of data, there is no assurance of its type or its level of quality. Online courses are readily available, well structured, have high quality content, and provide both spoken (video lectures) and written content within a particular domain. The first challenge is to develop an automated process for extracting course content, using it to build domain-specific corpora and wordlists.

1.1.2 Automating a receptive vocabulary test

Measuring learners' vocabulary growth is achieved using domain-specific vocabulary tests, but existing tests are hard to find and creating new ones can result in unreliable measures. This leads to the second challenge:

2. *Recreating an existing vocabulary test automatically using domain-specific vocabulary.*

Vocabulary tests are used to track learners' vocabulary growth. However, like corpora, vocabulary tests are readily available for general-purpose language, but not for all domains. Measuring learners' vocabulary in specialist domains requires custom tests, yet creating custom vocabulary tests is difficult and time-consuming, and can result in tests that are unreliable.

Furthermore, most existing vocabulary tests presuppose some deeper understanding of the language, for example, knowledge of vocabulary use within a

sentence, or knowledge of synonyms, collocates, and semantic relations. This makes recreating an existing vocabulary test using an automated process difficult.

The EFL Vocabulary Test requires nothing more than a wordlist and related pseudowords, but in order to create a reliable test, the pseudowords must be related to the domain. The second challenge can be split in two: (1) automatically generating domain-specific pseudowords, and (2) using those pseudowords to automatically generate domain-specific versions of the EFL Vocabulary Test.

1.1.3 Integrating language resources into online courses

Online courses are more often than not taught in English. Yet, although more people speak English as their second language than as their first, course providers do not support language acquisition. This leads to the third challenge.

3. *Integrating language resources into online courses without disturbing the original content.*

Content-based language learning involves non-language-based subjects, such as science, being learned in a second language, where language ability is essential for learning the subject. It is used by language teachers, integrating language into subjects such as engineering, mathematics, and geography, but this is usually done in the classroom, rather than online; to my knowledge, no one has applied content-based language learning to online courses.

Language includes not just single words but multi-word lexical items such as collocations and lexical bundles. Learners who are familiar with such items can express ideas simply and precisely, thereby communicating more effectively. However, to achieve lexical competence over the entire gamut of items – words, collocations, lexical bundles – learners must master a significantly larger collection of vocabulary than mere single words. The final challenge is to develop an application that integrates language resources into online courses without disturbing the original content.

1.2 Contributions

The contributions made during this investigation include: research-based contributions, a suite of linguistic software, language resources, and publications and presentations. They are as follows.

1.2.1 Research contributions

C*reating online domain-specific corpora* (Chapter 3): this thesis proposes the use of online course content to create domain-specific corpora. This has five main advantages over web-based corpus creation: online courses are readily available, they are more structured than the web, they have higher quality content, they provide both spoken and written content, and they provide content within a particular domain.

Creating domain-specific wordlists (Chapter 3): the criteria for including words in a wordlist depends on its purpose. This thesis proposes a generalised set of criteria for creating domain-specific wordlists from online course content (based on the criteria used for the Academic Word List). This technique has one important advantage over others, it is generalised. It can be applied to any corpus created from online course content, to create a wordlist of vocabulary specific to that course's domain.

Generating pseudowords (Chapter 4): *w*hile pseudoword generation techniques exist, each requires existing knowledge of the target language and lacks support for domain-specific pseudowords. This thesis developed a novel pseudoword generation technique, chaining character-grams to form pseudowords. This technique holds two main advantages (1) it does not require any knowledge of the language, thereby facilitating the generation of pseudowords in any language, and (2) the pseudowords reflect the wordlist used to create them, thereby facilitating the generation of pseudowords specific to a certain domain.

Generating vocabulary tests (Chapter 5): while receptive vocabulary tests exist, most require existing knowledge of the target language and lack support for specific domains. This thesis proposes a new technique for generating vocabulary tests entirely automatically, based on the structure of the well-founded EFL Vocabulary Test. this technique holds two main advantages (1) it does not require any knowledge of the language, thereby facilitating the automatic generation of vocabulary tests (2) the tests reflect the wordlist used to create them, thereby facilitating the generation of vocabulary tests for certain domains.

Supporting content-based language learning (Chapters 6, 7, and 8): content-based language learning is used in classroom scenarios, but is not often used online. This thesis proposes the integration of language resources into existing

online courses, supporting content-based language learning. The main advantage of this technique is that the content has already been created, and the language resources can be integrated into courses by using F-Lingo, allowing teachers to support content-based learning with very little additional effort.

1.2.2 Language resources

This work resulted in the creation of four language resources, each of which can be downloaded from GitHub[1].

1. The *DMwW corpus* is a domain-specific corpus. Created from the *Data Mining with Weka* courses on FutureLearn, it has 206,832 running words: 103,916 spoken English and 102,916 written. The corpus has been annotated with headwords, part of speech, and frequency bands. It is described in Section 3.7, and both plain text and annotated versions can be downloaded from GitHub.
2. The *DMwW wordlist* is a domain-specific wordlist. Created from the DMwW corpus, it is a list of 571 headwords that represent the vocabulary in the *Data Mining with Weka* domain. It is described in Section 3.7, can be downloaded from GitHub, and is included in this work in Appendix A.
3. The *DMwW pseudoword list* is a list of domain-specific pseudowords that were generated using the CGCA algorithm and the DMwW wordlist. It contains 400 pseudowords, 100 generated using 2-grams, 3-grams, 4-grams and r-grams. It is described in Section 4.8, can be downloaded from GitHub, and is included in this thesis in Appendix B.3.
4. The *DMwW Vocabulary Test* is a domain-specific version of the EFL Vocabulary Test. It was generated using the AEFL algorithm and the DMwW wordlist and contains 40 real words (from the DMwW wordlist) and 20 pseudowords. It can be used to measure learners' receptive vocabulary knowledge in the field of Data Mining. It is described in Section 5.7, can be downloaded from GitHub, and is included in this thesis in Appendix C.3.

[1] github.com/jlkonig/Language_Resources/

1.2.3 Linguistic software

This work resulted in the creation of six software applications, each of which can be downloaded from GitHub[2].

1. The *CourseCrawler*, described in Section 3.5.1, is a Chrome extension that can be used to crawl courses and extract spoken and written content from FutureLearn. It is not published in the Chrome Web Store, but can be download from GitHub.
2. The *CourseCorpusBuilder*, described in Sections 3.5.2 to 3.5.4, is a Python application that can be used to build and annotate corpora using text extracted from courses by the CourseCrawler.
3. The *CourseWordlistBuilder*, described in Section 3.6, is a Python application that can be used to build domain-specific wordlists from corpora built using the CourseCorpusBuilder. Word selection is based on criteria used by the Academic Word List.
4. The *CGCA Algorithm*, described in Section 4.4, is a Python application that can be used to generate domain-specific pseudowords using character-grams of a specified length.
5. The *AEFL Algorithm*, described in Section 5.3, is a Python application that can be used to generate domain-specific vocabulary tests, based on the EFL vocabulary test.
6. *F-Lingo*, described in Sections 6.6 and 7.1 to7.7, is a Chrome extension that works on top of FutureLearn to highlight words, phrases, and concepts within course content, and provides learners with additional lexical information about each. It can be downloaded and installed through the Chrome Web Store[3].

1.2.4 Presentations and publications

This work resulted in one publication and four presentations. The slides for the presentations can be found online[4].

[2] github.com/jlkonig/Linguistic_Software/
[3] https://chrome.google.com/webstore/detail /f-lingo/gpnkpjgnifiokihiclgdclhcghhenpnk
[4] https://www.cms.waikato.ac.nz/~jlk25/

1. König, J., Calude, A., & Coxhead, A. (in Press). *Using character-grams to automatically generate pseudowords and how to evaluate them*. Applied Linguistics Journal.
2. Fitzgerald, A., König, J., Witten, I. H. *F-Lingo: Integrating lexical feature identification into MOOC platforms for learning professional and academic English.* Accepted for presentation at the Learning with MOOCs (LWMOOCs) conference 2019, Milwaukee, USA.
3. König, J., Calude, A., & Coxhead, A. (2018). *Using character-grams to automatically generate pseudowords and how to evaluate them*. Presented at LingSoc2018 - Linguistic Society of NZ Conference, Wellington, New Zealand.
4. König, J. (2018). *Integrating lexical features and language tools into the FutureLearn platform*. Presented remotely, for the FutureLearn Academic Network Meeting, at the London School of Hygiene and Tropical Medicine, London UK
5. König, J. (2018). *N-grams, pseudowords, and vocabulary modelling*. Presented at the School of Linguistics and Applied Language Studies, Victoria University of Wellington, New Zealand.
6. König, J., Wu, S., Fitzgerald, A., Witten, I. H. (2017). *Developing language resources for MOOC students*. Presented at ESP2017 - International Conference on English for Special Purposes, New Technologies and Digital learning, Kowloon, Hong Kong.

1.3 Thesis structure

This thesis is structured as follows. Chapter 2 provides background on linguistic and language techniques. It defines single and multi-word lexical items, discusses word knowledge, frequency and coverage, and provides background on corpora and wordlists. Finally, it reviews content-based language learning and Massive Open Online Courses.

Chapter 3 investigates the first challenge, that online course content can be used to build domain-specific corpora and wordlists using a set of automated processes. It starts by discussing existing corpora and wordlists, then suggests a new approach, using course content to build domain-specific corpora. It introduces a suite of software that was developed for this thesis, and applies them to the *Data*

Mining with Weka courses that are available in the FutureLearn MOOC consortium to create the *DMwW corpus* and *DMwW wordlist*.

Chapters 4 and 5 investigate the second challenge, that an existing vocabulary test can be automatically recreated for domain-specific vocabulary, using automatically generated pseudowords. Chapter 4 is centred on generating domain-specific pseudowords. The first half of the chapter describes the CGCA algorithm, a new approach that chains character-grams together to form pseudowords, allowing linguistic researchers and language teachers to use domain-specific wordlists to produce domain-specific pseudowords. The second half of the chapter introduces a set of linguistic criteria for evaluating pseudowords, both in terms of their orthographic fit to the target language, and their suitability for use in lexical processing and language teaching. The evaluation criteria are used to provide a comparison with other current pseudoword lists, specifically the ARC Nonword Database, the English Lexicon Project, WordGen, Wuggy, and pseudowords used in Meara's EFL vocabulary tests.

Chapter 5 is centred on generating domain-specific vocabulary tests. The first half of the chapter describes and evaluates the AEFL algorithm, a system that builds receptive vocabulary tests automatically, replicating the format of the EFL Vocabulary Test, and allowing linguistic researchers and language teachers to use domain-specific wordlists or corpora to produce domain-specific vocabulary tests. The second half of the chapter evaluates the way in which the EFL Vocabulary Test is scored. Researchers have debated the original scoring method and have suggested several alternatives. This chapter evaluates which scoring method best evaluates the EFL Vocabulary Test based on the results of two user studies, and compares the results with those produced by other researchers.

Chapters 6, 7, and 8 investigate the third challenge, that language resources can be integrated into online courses without disturbing the original content. This challenge has been met by designing, implementing, and evaluating F-Lingo, a Chrome extension that works on top of FutureLearn to provide learners with language resources. Chapter 6 focuses on the research and design of F-Lingo. The first half of the chapter investigates an existing language teaching technique and reviews six existing online language applications. This includes subjecting them to a feature comparison, which is used to determine the set of vocabulary items and language resources used by F-Lingo. The second half of the chapter investigates

these items and resources, and proposes the design considerations and planning for F-Lingo.

Chapter 7 describes the development of F-Lingo, based on the design summary outlined in Chapter 6. It investigates identifying keywords, phrases, and concepts within text; describes traversing and extracting content from within a course; illustrates highlighting words, phrases and concepts on a page; outlines a set of online language resources for dictionary definitions, example sentences, related phrases, and disambiguated descriptions from Wikipedia; and describes the development of course-specific wordlists and vocabulary tests. Finally, it outlines the process of applying F-Lingo to a FutureLearn course.

Chapter 8 describes a set of evaluations and analyses that were conducted on F-Lingo. The first part of the chapter describes an expert heuristic evaluation. Conducted as a cognitive walkthrough, this was used to highlight any shortfalls in the design and usability of F-Lingo's user interface. The rest of the chapter outlines a data-based evaluation, analysing learner behaviour in relation to their use of the F-Lingo Chrome extension. The data-based analysis was divided into four parts: analysing participant information; analysing learners' interaction with words, phrases, and concepts; analysing the duration of time learners spent looking at language resources; and a more in-depth analysis of the actual words, phrases, and concepts that were clicked. Finally, Chapter 9 concludes the thesis and discusses future work.

Chapter 2
Background

This thesis describes the development of automated processes for extracting online course content, testing vocabulary, and enriching online text, the foundations of which lie within the fields of linguistics and language teaching. As with any challenge, before a solution can be found, some understanding of the underlying field must be obtained. In this case, before any automated processes can be developed, we must first gain some understanding in the field of linguistics, particularly in applied and corpus linguistics, and in language teaching. This chapter does not attempt to cover the vast amount of knowledge required by experts in these fields, but instead focuses on topics within them that are relevant to this work. It explores learning and measuring vocabulary, building corpora and wordlists, and investigates content-based language learning and the recent phenomenon of so called “Massive Open Online Courses”.

Table 2.1 Headwords and word families

Types	Headwords	Word families
teachers	teach	taught, teachable, teacher, teacherly, teachers, teaches, teaching, teachings
open	open	opened, opener, openers, opening, openings, openly, openness, opens, reopen, reopened, reopening, reopens, unopened
the	the	
door	door	doors, indoor, indoors, outdoor, outdoors
you	you	ye, yer, yerself, your, yours, yourself, yourselves, yous, youse
enter	enter	entered, entering, enters, unentered
by	by	bys
yourself	you	ye, yer, yerself, your, yours, yourself, yourselves, yous, youse

2.1 Vocabulary

The term "vocabulary" can refer to all the words in a language, the words present in a particular domain, or the words known by a learner. It can refer to both individual words and multi-word lexical items.

2.1.1 Words

In linguistics, the term "word" is used to refer to several different entities, each with its own meaning (Milton, 2009). This thesis focuses on four: running words, types, headwords, and word families. Consider this sentence, a Chinese proverb quoted in the online *Data Mining with Weka* course.

"Teachers open the door, you enter by yourself."

How many words does it contain? Counting the number of separate words, we can see that it contains eight: *teachers*, *open*, *the*, *door*, *you*, *enter*, *by*, and *yourself*. These are referred to as "running words" (or "tokens"). Consider the sentence again, with eight running words, how many types are there? "Types" are the unique words.

In this case, all of the words are unique, so there are both eight running words and eight types.

Each word belongs to a family comprising a headword and its inflected and derived forms (Bauer & Nation, 1993). A "headword" is the simplest version of the word, and inflected and derived forms are expanded from it. Look again at the example sentence, what are the headwords? The word *teacher* consists of the root *teach* and the suffix *–er*, so its headword is *teach*. In contrast, *open*, *the*, and *door* are the simplest versions of themselves, so they themselves are the headwords.

A word family consist of the headword's inflected and derived forms. However, they are determined not just by the structure of words but also by their semantics. The words *hard* and *hardly* are spelt similarly but do not have the same meaning, making them members of different word families. Because of this semantic relationship, researchers suggest that once a learner knows one form of a word they will be able to recognize other members of the same family with little to no added knowledge (Bauer & Nation, 1993; Milton, 2009). Table 2.1 illustrates both the headwords and word families for the sentence: *teachers open the door, you enter by yourself*, according to Nation, Heatley, and Coxhead (2002).

Reducing a word down to its headword is just one technique used by linguists, applied linguists and language researchers. Other techniques include lemmatization and stemming. Like headwords, stems and lemmas are the simplest version of a word. However, unlike headwords, the technique used to generate them does not take semantics into account. Both stemming and lemmatization are systematic process. The former involves using pattern matching to removing prefixes and suffixes from words, while the latter uses vocabulary and morphology to determine a word's base form (Manning, Raghavan, & Schütze, 2008). This thesis uses headwords and word families primarily, but supplements them with lemmas where necessary (Section 3.5.3).

2.1.2 Phrases

Consider another sentence.

"This course introduces you to practical data mining."

The words *data* and *mining* have their own individual meaning. *Data* is a collection of (often organised) information, while *mining* is the excavation of solid materials

Table 2.2 Collocation patterns and examples

Pattern	Example
Adjective + noun	Big brother
Adverb + adjective	Happily married
Noun + noun	Head teacher
Verb + adverb	Act quickly
Verb + prepositional phrase	Looking forward to

Table 2.3 Sentence-initial lexical bundle examples

Discourse bundles	Interactional bundles
The fact that the	As can be seen
The result of the	It should be noted
In the case of	It is interesting to

(Wiktionary, 2019). However, the phrase *data mining* denotes a technique for searching large-scale databases for patterns (Wiktionary, 2019). Vocabulary is not just single words, but also multi-word phrases. A phrase is a group of words that function as a single unit or express a single concept. There are several types of phrase, two of which are *collocations* and *lexical bundles*.

2.1.2.1 Collocations

A "collocation" is a sequence of two or more words that occur together more frequently than by chance, and that hold semantic meaning (Nation, 2001; Nattinger & DeCarrico, 1992). There are several types of collocations made from different combinations of part-of-speech, for example, collective noun phrases (noun + noun) and phrasal verbs (verb + prepositional phrase). Table 2.2 shows some examples.

Second language learners use collocations to increase their fluency, and by doing so exhibit a more competent grasp of the language. Collocations can be used to form more natural sounding sentences; even a grammatically correct sentence can stand out as unnatural if collocations are used incorrectly (*strong* tea versus *powerful* tea) (Halliday, 1966).

2.1.2.2 Lexical bundles

A "lexical bundle" is a sequence of words that appear repeatedly within a language and mark the direction or purpose of text, either by relating new information

(discourse bundles) or marking reader and writer involvement (interactional bundles) (Li, 2016). Lexical bundles are often not structurally complete, but become complete when other words are added, for example, *in the case of* or *it is interesting to.*

Lexical bundles are usually three or more words long and can occur either at the start of a sentence (sentence-initial bundles) or within it. Table 2.3 illustrates examples of sentence-initial discourse and interactional bundles.

2.2 Learning vocabulary

Both word frequency and range are important for vocabulary acquisition. The more often words appear (frequency) and the more text they appear in (range), the more likely they are to be learned. Words with higher frequency and range are thought to be acquired when beginning to learn a language, while less frequent words are more likely to be learned later on. When initially learning a language, vocabulary is often learned by rote memorizing wordlists. Deciding which words to include in these wordlists is determined by their frequently and range within a language.

2.2.1 Frequency

"Frequency" refers to the number of times a word occurs in a language, or a representation of that language such as a corpus. The words that appear most frequently are almost always function words; they carry little meaning on their own, but are crucial to creating coherent sentences (Milton, 2009). In contrast, content words tend to have much lower frequency counts, but hold more meaning and create context within a language.

Researchers often divide words into bands based on their frequency (Meara, 1992), specifically into 1000s of words, which are referred to as "frequency bands". The first thousand most frequent words belong to the 1K frequency band, the second thousand most frequent to 2K, the third thousand to 3K, and so on. Frequency contributes greatly to vocabulary acquisition. Researchers have long suggested that words that are more frequently used are more easily learnt (Mackey, 1967; McCarthy, 1990; Palmer, 1917). Figure 2.1 shows the frequency profile for a typical learner, where word knowledge is higher in the higher frequency bands, and lower in the lower frequency bands (Meara, 1992). More recent studies (Milton, 2009; Richards & Malvern, 2007) have given considerable support to the idea that

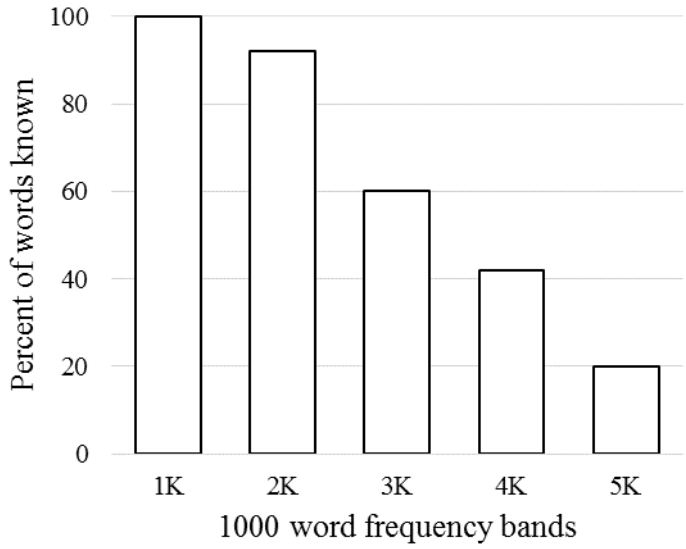


Figure 2.1 The frequency profile for a typical learner

the more frequent a word is, the more likely it is to be learned.

Determining the frequency of words that learners know can also give insight into how comprehensive their grasp of the language is. It has been suggested that a learner who knows only low-frequency words will be less able to express themselves than a learner who knows more high-frequency words (Milton, 2009).

2.2.2 Coverage

While *range* refers to the distribution of an individual word across text, *coverage* refers to the distribution of the words in a particular frequency list across written or spoken language. Table 2.4 illustrates word coverage across a five million word corpus (J. B. Carroll, Davies, & Richman, 1971; Nation, 1983). The first ten most frequent words cover 23% of text in the corpus, the first hundred cover 49%, the first thousand cover 74%, and so on.

Nation (2013) suggests three groups of words based on their coverage: high, mid, and low frequency. They are as follows.

1. *High frequency.* These are the most frequent 2000 words (1st and 2nd 1000-word lists), which cover approximately 80% of written text. These words include function words such as *the*, *a*, *in* and *of*. In Table 2.4 the first 2000 words account for 81% of the text.

Table 2.4 Word coverage in a 5 million word corpus

Word families	Percent of running words
86,741	100%
43,831	99%
5,000	89%
3,000	85%
2,000	81%
100	49%
10	23%

2. *Mid-frequency*. These are approximately 7,000 words from the 3rd to 9th 1000-word frequency lists. They are thought to be mainly general-purpose words. When combined with low frequency words, these make up the remaining 20% of text – a significantly larger number of words account for a much smaller portion of text. In Table 2.4 84,741 words make up the final 19%.
3. *Low frequency*. These are any words that appear in the 10th 1000-word frequency list or more. These words appear very infrequently but often hold great importance in the context of the corpus.

In addition to this, Nation (2013) suggests two further word groups, based on the type of text being considered: academic words, and technical words.

1. *Academic words*. These are words commonly seen in academic text across multiple disciplines, such as *policy*, *sustain*, and *exceed*. Academic text and word lists are discussed in further detail in Section 0.
2. *Technical words*. These are words that occur more often in a particular domain than in general-purpose language, referred to in this thesis as *domain-specific words.* See Section 0 for more details.

How much coverage is needed for comprehension? Researchers suggest that understanding the 2000 most frequent words provides learners with enough knowledge to begin to understand general ideas and language (Hu & Nation, 2000; Laufer, 1989, 1992; Milton, 2009), but that 95%-98% text coverage is required in order to provide adequate comprehension and have a manageable amount of unknown vocabulary (2%-5%) on the page (Nation, 2013; Schmitt, Jiang, & Grabe, 2011; van Zeeland & Schmitt, 2012).

2.3 Measuring vocabulary

It is often necessary to determine a learner's vocabulary size, whether it be required for tracking vocabulary growth or research purposes. However, before vocabulary size can be measured, we first must determine what a word is (discussed in Section 2.1.1), and what "knowing a word" means (Read, 1988). Being able to recognize a word is not the same as being able to use it competently in speech or writing.

2.3.1 Word knowledge

Knowledge is commonly divided into two types: *receptive* (or passive) and *productive* (or active) knowledge. It is important to clearly distinguish the two (Read, 2000). Receptive knowledge consists of the words that learners recognize when heard or read, while productive knowledge consists of the words they are able to call to mind, and that they can use in speech or writing (Pignot-Shahov, 2012). The former includes words that learners do not know well but can understand when encountered in context, while the latter includes words that learners can use appropriately and with the correct meaning. This thesis focuses on receptive vocabulary knowledge.

2.3.2 Vocabulary tests

Vocabulary tests can be used to measure learners' receptive vocabulary knowledge, estimating proficiency levels, representing progress, or as a tool for determining target wordlists. Vocabulary tests come in a plethora of forms, some for receptive and some for productive vocabulary. This section describes two common forms of receptive vocabulary tests: the *matching* (recognition) format and the *checklist* (yes/no) format, as well as some well-known and strongly founded examples of each.

2.3.2.1 Matching

Tests that use the *match* format rely on the recognition of items, often matching target words with short definitions, related words, or phrases (Read, 2000). Two such examples are the Vocabulary Size Test and the Word Associate Test.

The Vocabulary Size Test, developed by Nation and Beglar (2007), was designed to measure receptive vocabulary for first (L1) and second (L2) language

Table 2.5 Example question from the Vocabulary Size Test

1. soldier: He is a *soldier*.
    a. person in a business
    b. student
    c. person who uses metal
    d. <u>person in the army</u>

Table 2.6 Word frequency ranges

Frequency ranges	Most frequent words
High frequency	1st–2000th
Mid frequency	2001st–10,000th
Low frequency	10,001st onwards

learners. The test has several versions, both monolingual and bilingual, and includes words up to the 20th thousand-word frequency band. The test uses a multiple-choice format. Learners are given a word and a sample sentence that includes it. They are then supplied with four possible definitions, only one of which is correct. Table 2.5 shows an example, with the correct answer underlined.

Although the test includes words up to the 20th thousand-word frequency band, it is not a reliable measure of how well each individual band is known (Nation & Beglar, 2007). Covering all bands means that only a few words are included from each, which is insufficient for a reliable estimation. However, the results of the test can be used to estimate the number of word families that a learner knows, which can be expressed using three frequency ranges; high frequency, mid frequency, and low frequency, as shown in Table 2.6.

The Word Associate Test, developed by Read (1993), was intended to surpass conventional vocabulary tests. It provides a practical way of measuring word knowledge using word association. Learners are presented with a stimulus word and a group of other words, and are asked to identify the ones that relate.

There are two versions of the test, each containing 50 stimulus words. Each stimulus has four related and four unrelated words. Related words (associates) are chosen based on three characteristics: synonyms, collocates, and semantic relations,

Table 2.7 Example question from the Word Associate Test

1. sudden			
beautiful	surprising	change	doctor
quick	thirsty	school	noise

Table 2.8 Example words from the EFL Vocabulary Test

1. oligation	2. education	3. board
4. answer	5. clean	6. gummer

and unrelated words (distractors) are chosen based on their complete lack of semantic relatedness with the corresponding stimulus word. Table 2.7 shows an example, with associates underlined.

Each question is scored with a mark out of four, corresponding to the four associate words. However, scoring does not take into account incorrect guessing, for example, a learner who selects two associates and two distractors would get a score of 2, the same as someone who selected the two associate words alone.

2.3.2.2 Checklist

The *checklist* format (often referred to as *yes/no*) simply requires learners to express whether or not they think they know a word (Read, 2000). This is usually done by ticking a checkbox or answering "yes" or "no", hence the alternative name, "Yes/No Test". Three such example of this are the English for Foreign Language (EFL) Vocabulary Test, the X_Lex Test, and the V_YesNo Test, each of which was developed in part by Paul Meara.

The English for Foreign Language (EFL) Vocabulary Test, developed by Meara (1992), is an extension of his Yes/No Vocabulary Test (Meara & Buxton, 1987), and is an alternative to traditional multiple-choice vocabulary tests. It comprises a selection of real words and pseudowords, and learners must indicate whether or not they know each one. There are 20 versions of the test, each of which has five levels that correspond to the first five frequency bands. Each level contains 60 items: 40 real words and 20 pseudowords. Learners are asked to consider each word individually and indicate whether or not they know it. Table 2.8 shows an example, with real words underlined. The test is scored using hit rates, the real words that learners know, and false alarm rates, the pseudowords that learners claim

Table 2.9 Interpreting a V_YesNo vocabulary score

Score	Interpretation
< 1500	Typical for beginners. At this level the test is unreliable.
2500–4500	Intermediate level
4500–7500	Good level of competence
7500–900	High proficiency level

to know. Pseudowords are included in order to assess how much a learner is guessing, for example, if a learner claims to recognize several pseudowords, they may have overestimated the proportion of real words they know. Pseudowords are used to decrease the overall score to compensate for over-guessing.

The X_Lex Test, developed by Meara and Milton (2003) is a digital version of a Yes/No test that runs on the Windows operating system. Like the EFL Vocabulary Test, it tests vocabulary up to the 5K frequency band. The test consists of 300 words and pseudowords, with a selection of them shown to the learner each time the test is taken. It is self-scoring, providing learners with a rough lexical profile that can be used to assess an individual's vocabulary requirements (Wikipedia, 2019). The most recent version of X_Lex is v2.05. However, it (and earlier versions) are no longer supported or available for use.[5]

The V_YesNo Test is another Yes/No based vocabulary test, this time developed by Meara and Miralpeix (2015). The test structure loosely resembles X_Lex, but tests vocabulary up to the 10K frequency band. Like X_Lex, the V_YesNo test is digital. However, rather than running on Windows, V_YesNo is available online.[6] It provides learners with a selection of 200 words and pseudowords, displaying them one at a time and allowing learners to indicate whether they know the word ("yes") or not ("next"). Once the test is complete learners are presented with a final score, between 0 and 10,000, which estimates the learner's vocabulary. Table 2.9 illustrates how Meara and Miralpeix (2016) suggest the score should be interpreted.

[5] http://www.lognostics.co.uk/tools/
[6] http://www.lognostics.co.uk/tools/V_YesNo/V_YesNo.htm

2.4 Corpora

A *corpus* is a collection of machine readable text that represents a particular language or domain, and can be analyzed to draw conclusions that are useful for lexical processing and language learning (Xiao, 2010). A well-designed corpus uses text that fairly represents an entire language, or a specific domain, and can be used to make inferences and predictions (Meyer, 2002).

2.4.1 Building

Building a corpus involves planning, text collection, and digitization. The first involves determining the purpose of the corpus, which in turn dictates the type of text that will be collected. The type and number of texts should fairly represent the language or domain as a whole (Biber, 1993). The second involves collecting the text, often manually. This may seem simple but becomes increasingly complex and time consuming as the size of the corpus increases (Meyer, 2002). The third involves scanning and digitizing text, including data conversion for digitized files and transcribing speech (O'Keeffe & McCarthy, 2010). Both text collection and digitization are often still undertaken manually, at least in part. Corpora that include spoken text often need to be manually transcribed; corpora that include hard copies often require manual scanning before being digitally converted; and text that has been digitally converted often needs to be checked manually for any errors that may have occurred.

2.4.2 Structuring

Corpora are often structured in some meaningful way, usually as a way of representing the language they contain, for example, the British National Corpus (BNC) (*British National Corpus*, 2007), discussed in Section 2.4.5, contains 4049 texts, each of which is categorized as either: spoken demographic, spoken context-governed, written books and periodicals, written-to-be-spoken, or written miscellaneous (Burnard, 2007).

2.4.3 Annotating

After a corpus has been built, it can be annotated with additional lexical information to aid in corpus analysis. The type of annotation that is added depends on the

purpose of the corpus, for example, a researcher interested in vocabulary acquisition may annotate a corpus with headwords, while research into collocations and lexical bundles may include part-of-speech tagging and grammatical mark-up. Corpus annotation is usually undertaken computationally, for example, scripts are used to split text into individual tokens; stemming software is used to tag tokens with their headwords; and part-of-speech taggers are used to identify nouns, verbs, adjectives and so on.

2.4.4 Analysing

Once a corpus has been annotated, it can be analysed to extract useful information. This is most often undertaken computationally. According to Read (2007), computer-based corpus analysis has greatly influenced vocabulary studies, allowing researchers to define words, calculate frequencies, observe collocational behaviour, and so on.

The type of analysis conducted depends on the research question being asked, for example, looking at domain-specific lexical items may include determining which items are present in the corpus (wordlists), how often each item appears (frequency), and how many texts the item appears in (range). Like annotation, corpus analysis is usually undertaken using linguistic software, for example, AntConc (Anthony, 2004) can be used to analyse concordance data, generate word frequencies, and create word distribution plots.

2.4.5 Types of corpora

General-purpose corpora are those that fairly represent a language as a whole, for example, the British National Corpus, Brown Corpus, and Corpus of Contemporary English. The British National Corpus (BNC) (*British National Corpus*, 2007) is a collection of both written and spoken English that contains text from a wide range of categories (shown in Table 2.10), is 100 million words in size, and represents the British English language from the late 20th century (Burnard, 2007).

The Brown University Standard Corpus of Present-Day American English (the Brown Corpus) was created in the 1960s and consists of over 500 samples of text, each approximately 2000 words in size. It covers 15 genres, shown in Table 2.11, and is a representation of American English language in the 1960s (Kučera & Francis, 1967).

Table 2.10 Distribution of text in the BNC

Category	Range
Spoken demographic	10.13%
Spoken context-governed	7.09%
Written books and periodicals	72.94%
Written-to-be-spoken	1.73%
Written miscellaneous	8.09%

Table 2.11 Distribution of text in the Brown Corpus

Category	Articles
Press: reportage	44
Press: editorial	27
Press: reviews	17
Religion	17
Skills and hobbies	36
Popular lore	48
Biographies and memoirs	75
Government and house organs	30
Learned	80
Fiction: general	29
Fiction: mystery and detective	24
Fiction: science	6
Fiction: adventure and western	29
Fiction: romance	29
Humour	9

Table 2.12 Distribution of text in COCA

Category	Running word count
Spoken	85,000,000
Fiction	81,000,000
Popular magazines	86,000,000
Newspapers	81,000,000
Academic journals	81,000,000

Table 2.13 Categories in the Academic Corpus

Arts	Commerce	Law	Biology
Education	Accounting	Constitutional	Chemistry
History	Economics	Criminal	Computer Science
Linguistics	Finance	Family & Medical	Geography
Philosophy	Industrial Relations	International	Geology
Politics	Management	Pure Commercial	Mathematics
Psychology	Marketing	Quasi-Commercial	Physics
Sociology	Public Policy	Rights and Remedies	

The Corpus of Contemporary American English (COCA) (Davies, 2008), originally developed in 2008, contains more than 560 million words, 20 million each from the years 1990 to 2017 and is one of the largest freely-available corpora of English. It is evenly divided between five categories: spoken, fictional, magazines, newspapers, and academic texts, as shown in Table 2.12, and as the name implies, it is a representation of American English.

Domain-specific corpora represent a particular domain within a language. The Academic Corpus (Coxhead, 2000) is one such example. As a corpus of academic written English, it contains approximately 3,500,000 running words from four categories (arts, commerce, law, and biology) and 28 sub-categories (see Table 2.13). Each subcategory contains approximately 125,000 running words. Coxhead (2000) analysed the Academic Corpus to determine which lexical items were present, how often each item appeared (frequency), and how many categories it appeared in (range). This type of analysis helps to build useful wordlists.

2.5 Wordlists

A wordlist is a list of words that appear in text, or in a corpus. It often includes frequency counts, and the words are ordered from most to least frequent. Linguistic researchers create wordlists from corpora to analyse the words that appear in text, and language teachers use them to help vocabulary acquisition by providing learners with lists of words to study.

2.5.1 Building wordlists

Wordlists can be created simply by splitting text into words and recording them.

Table 2.14 The first twenty headwords in the GSL

a	absence	account	act
able	absolutely	accuse	actual
about	accept	accustom	add
above	accident	ache	address
abroad	accord	across	admire

Table 2.15 The first twenty words in the Range Programme Lists

a	abler	about	absolutism
an	ablest	above	absolutist
able	ably	absolute	absolutists
abilities	inability	absolutely	accept
ability	unable	absolutes	acceptability

However, some tasks require further tuning. Wordlists can contain headwords, types, or word families. They can include all of the words in text, only the first 1000, or only those that are domain-specific. There are no hard and fast rules for building wordlists, instead linguists adopt criteria based on their intended use.

2.5.2 Types of wordlists

General-purpose wordlists, like general-purpose corpora, represent a language as a whole. The General Service List and the Range Program Lists are two examples. The former contains 2000 headwords taken from a written corpus, representing words of the greatest ‘general’ use in English. It was created by West (1953). He did not describe the underlying corpus in detail, but tells of its construction from “many sources, such as encyclopaedias, magazines, textbooks, novels, essays, biographies, and books about science, poetry and the like” (West, 1953, p. xi). The original corpus contained 2.5M words and was supplemented by a second sample of the same size. Words were selected based on frequency, ease or difficulty of learning, necessity, coverage, stylistic level, and intensive or emotional content. Each entry in the GSL includes a headword, derived forms, and definitions from the Oxford English Dictionary.

The Range Programme Lists (Nation et al., 2002) are a collection of 25 wordlists containing 1000 word families each. The first two were created from a

Table 2.16 The first 25 headwords in the Academic Word List

analyse	authority	constitute	define	establish
approach	available	context	derive	estimate
area	benefit	contract	distribute	evident
assess	concept	create	economy	export
assume	consist	data	environment	factor

specially designed corpus of 10M tokens, 6M from spoken English and 4M from written English. The word families for numbers (*one*, *two*, *three*, etc.) and weekdays were added to the first list, and months were added to the second.

These two wordlists contain a set of high frequency words that are suitable for creating language courses or graded readers (Nation et al., 2002). The remaining lists contain 1000 word families each from the COCA/BNC rankings, excluding those used in the first two lists. Table 2.15 illustrates the first few; the headwords for each word family are underlined.

Domain-specific wordlists are used to study specialized language in a particular domain, for example, the Academic Word List (AWL) – a list of word families derived from the Academic Corpus – was developed to aid learners and teachers of tertiary English (Coxhead, 2000). Proper nouns, Latin forms such as *et al* and *etc.*, and words from the GSL (West, 1953) were excluded. All remaining word families had to meet the following criteria:

1. Range: must occur in all main categories of the corpus, and at least half of the sub-categories;
2. Frequency: must occur at least 100 times in the 3,500,000-word corpus;
3. Uniformity: must appear at least 10 times in each main category.

This resulted in a list of 570 word families, the first twenty headwords of which are shown in Table 2.16.

2.6 Content-based learning

Content-based language learning, or Content and Language Integrated Learning (CLIL) is a dual-learning concept: (i) learning a subject through a foreign language, and (ii) learning the foreign language by studying the subject (Marsh, 2002). It refers to non-language-based subjects, such as science, being learned in a second language, where language knowledge becomes the means of learning the subject.

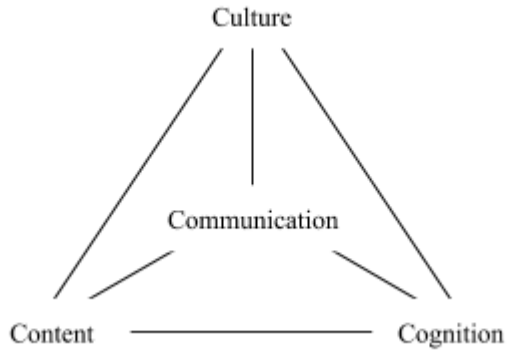


Figure 2.2 The 4 Cs Conceptual framework

Language is learned from natural contextualised text. In essence, it provides a practical approach to learning both subject and language (Darn, 2006).

2.6.1 Pedagogy

Underlying content-based language learning is the "4 Cs Conceptual Framework" (Coyle, 1999), which incorporates the four elements shown in Figure 2.2:

1. *Content* reflects the knowledge, skills, and understanding that students are expected to learn in a particular domain;
2. *Communication* uses foreign language to learn a subject, while using the subject to learn the foreign language (Pancheva & Antov, 2017);
3. *Cognition* involves allowing learners to build their own understanding, develop critical thinking skills, and form language;
4. *Culture* involves deepening cultural understanding and awareness through exposure to alternative perspectives.

Content-based language learning pedagogy, in many ways, resembles that of English Language Teaching. Darn (2006) provides five similarities. The first, *situational learning*, refers to content being presented as natural contextualised language, emulating real-life context and allowing language acquisition to take place. The second focuses on language *acquisition*, where content-based learning facilitates the acquisition of new language whilst reusing existing knowledge. The third refers to a *natural approach*, whereby fluency is developed by exploring language in natural and meaningful contexts. The fourth relates to the learner's *motivation*, which is boosted by using language naturally and in subjects of interest. Finally, content-based language learning reflects English language *teaching*

practices, which regard grammar as secondary to vocabulary, accuracy as secondary to fluency, and exposes learners to language in chunks.

2.6.2 Benefits

Content-based language learning has foundations in cross-curricular teaching, allowing students to use the knowledge learned in one context as a knowledge base for others. Students apply, integrate, and transfer knowledge, which can increase their motivation for learning. They see the value of what they are learning, and are encouraged to become more actively involved and use their language skills to explore and communicate what they are learning (Darn, 2006).

Wu and Witten (2007) highlight three advantages. First, learning language within a subject of interest makes acquisition more interesting and motivational; second, it helps learners increase their linguistic knowledge in a particular domain; and third, subject-specific content introduces the culture of a language in a natural and meaningful way.

2.6.3 Applications

Pancheva and Antov (2017) applied content-based language learning techniques when teaching specialized engineering subjects. Their results show improved cognitive development, cultural awareness, and motivation to learn foreign language.

Dourda, Bratitsis, Griva, and Papadopoulou (2014) taught English to Greek primary school students using an online educational geography game. They measured student performance through the game, and found significant improvement from pre-test to post-test.

Jäppinen (2005) used content-based learning to teach mathematics. A control group that learned in their mother tongue, Finnish, was compared with a group who learned in English, French, or Swedish. Significant improvements in cognitive development were found between the two groups.

2.6.4 Software

The *InGenio* authoring tool and content manager was designed to aid in the dual teaching of non-language subjects and foreign language (Gimeno, Seiz, De

Siqueira, & Martinez, 2010). It can be used to create activities and support task-based and project-based learning.

The *Tools for CLIL Teachers* project (Dónaill & Gimeno-Sanz, 2013) supports content-based language learning by providing an online tool, called *Wordlink*, that links every word to an online dictionary, providing support for dictionaries in over 100 languages.

Wu and Witten (2007) used content-based language learning to support topic-specific language acquisition through specially built digital library collections. This work has since been expanded to include FLAX, described in Section 6.2.1.1, an online tool that supports language learning in a variety of domains (Wu, 2010).

2.7 Massive Open Online Courses

Massive Open Online Courses (MOOCs), a recent development in distance education, provide open access to lessons via the web. Ideally, they do not restrict participation and allow unlimited student enrolment from around the world. They often do not enforce time restrictions, allowing students to learn at their own pace. MOOCs have been said to revolutionize universities and academic learning (Kaplan & Haenlein, 2016). Many MOOCs offer filmed lectures, readings, quizzes, assignments and user forums. Three prominent examples are Coursera, edX, and FutureLearn.

Coursera[7] was founded in 2012 by two Stanford University professors, and now boasts 35 million learners, 150 university partners, and 2700 courses in 250 specializations. Courses include video lectures, auto-graded assignments, peer-graded assignments, and discussion forums (Coursera, 2019).

edX,[8] founded by Harvard University and MIT in 2012, has 130 global partners including universities, non-profits and institutions, it has 18 million learners, and 2200 courses, and claims to be the only MOOC platform that is non-profit and open source (edX, 2019).

FutureLearn,[9] founded in 2013 by The Open University, has 7.9 million users and 1000 courses (FutureLearn, 2019a). Owned by the Open University, it

[7] https://www.coursera.org/
[8] https://www.edx.org/
[9] https://www.futurelearn.com

a) Enrollment by continent b) Enrollment by gender

Figure 2.3 FutureLearn demographics

has hundreds of partners and offers courses from leading universities and institutions around the world. It is free to use and available on mobile, tablet, and desktop. It is described in a little more detail below because it is used in the present research to support content-based language learning.

FutureLearn published the demographics shown in Figure 2.3. At the time, (2016), there were 3 million learners, mostly based in Europe, Asia, and North America. They varied in age from under 25 to over 65, and more than half were female (Walton, 2016). An online review, conducted in 2018, ranked FutureLearn as the 3rd best MOOC platform, based on five core metrics: credentialing, course diversity, course features, social features, and partner institutions (The Review Team, 2018).

MOOCs have been touted as enabling online learners to gain university level education, but this presupposes a high level of English language ability in the domain of study. This thesis proposes the integration of language resources into MOOC courses, helping learners to increase their English vocabulary, and in turn, helping them obtain knowledge in the subject. The first step is to identify the vocabulary used in courses by building domain-specific corpora and wordlists to support domain-specific language acquisition.

Chapter 3

Domain-specific vocabulary

Vocabulary /vəʊˈkæbjʊləɪɪ/

(noun) (1) the collection of words a person knows and uses (2) the stock of words used in a particular field (3) the words of a language collectively.

(Wiktionary, 2019)

It is reasonable to assume that words such as *analytically* and *conceptualisation* would appear more frequently in academic writing than they would in general language. Vocabulary that is frequent in a domain but not necessarily in general is called "domain-specific" language. There are an unlimited number of domains, all of which can contain different vocabulary, and understanding that they differ from one another – and from general-purpose language – is important when teaching, learning, and measuring vocabulary. For example, measuring learners' lexical knowledge in data mining would yield very different results for someone studying it than someone who is not; the former is more likely to understand terms like *entropy* and *antecedent* than the latter.

Measuring learners' vocabulary in a domain first requires knowledge of which lexical items are present. This is often achieved by analysing the vocabulary

in a domain-specific corpus to build a wordlist. However, finding an existing corpus can be impossible, and creating new ones can be both difficult and time consuming. This has led us to explore the potential for creating domain-specific corpora and wordlists from online courses.

3.1 Corpora and wordlists

A *corpus* is a collection of text that fairly represents a particular language or domain, and can be analyzed to extract useful lexical information (Section 2.4). There are two main approaches to creating corpora, referred to as *the monitor corpus approach* and *the balanced corpus approach* (McEnery & Hardie, 2011). The former involves continually expanding the corpus, collecting more and more text over time, while the latter involves collecting a careful sample of text, representing language at one point. The second is the approach used here. Creating corpora involves: planning, collecting, and digitizing text (Section 2.4.1), structuring the corpus in a meaningful way (Section 2.4.2), annotating it with additional lexical information (Section 2.4.3), and analyzing it to produce some form of linguistic findings (Section 2.4.4).

Corpus analysis often results in the creation of wordlists, representing the vocabulary present in a particular language or domain (Section 2.5). Wordlists are often compiled based on word frequency and range, and can contain either headwords, types, or word families. They can contain: all words present in a corpus, representing the language or domain as a whole; the first n-many most frequent words, representing the most highly occurring vocabulary in the corpus; words that occur more frequently in the corpus then in general language, representing domain-specific vocabulary; and so on. There are no hard and fast rules for building wordlists, instead linguists and language teachers design criteria based on their intended use.

3.2 A new approach

Often the most time-consuming part of building a corpus is collecting and consolidating text. Collecting text from multiple sources (new paper, film, audio, academic text, etc.), in a variety of formats (hard copy, pdf, jpeg, mp4, text file, etc.) can take a considerable amount of time and financial resources, making corpus

collection an arduous task. Additionally, for some research areas, even corpora like the BNC are not big enough (Baroni & Ueyama, 2006), and less commonly used domains, languages, and varieties of languages are not always represented in existing corpora (Hundt, Nesselhauf, & Biewer, 2007). For these and other reasons, researchers have begun building corpora using content from the World Wide Web. Web crawlers are used to gather large amounts of data in short periods of time, making corpora creation much easier and faster.

However, using the web as a corpus has its own limitations. The web contains an ever-expanding amount of data, with no assurance of its type or quality. Using commercial web crawlers to extract small subsets of data tends to return varying results, with search engines creating local biases towards users' existing preferences. Furthermore, given the dynamic and sometimes short-lived nature of web content, replicability of a web crawled corpus is often impossible (Hundt et al., 2007).

The considerable limitations of web corpora have led us to consider whether online course content could be used instead, creating corpora for a variety of academic domains. Of course, this would result in much smaller corpora than using the web as a whole, but would represent a much smaller, more focused, domain, and would not be subjected to the same limitations as corpora created from the web. The quality of the material would be of a high standard, having been approved and published on online course platforms, and the corpus would be replicable, given that a specific set of pages would be used. This chapter describes a set of automated processes for building, structuring, annotating, and analysing corpora and generating wordlists, using online course content. Each automated process is demonstrated using a set of data mining courses on FutureLearn.

3.3 Online course platforms

FutureLearn is an online MOOC platform that provides learners with free online courses from leading universities, specialist organisations, and institutions around the world. Discussed in Section 2.7, it has millions of learners and is available on mobile, tablet, and desktop.

Learners have three study options: short courses, programs, and degree courses. Short courses are usually between five and ten weeks long and include a selection of videos, audios, articles, discussions, and quizzes, plus online

assessments for those who pay to upgrade their enrolment. Programs combine multiple short courses, and learners can enrol in degrees ranging from graduate certificates to full Masters Degrees.

The University of Waikato has one program running on FutureLearn, the *Practical Data Mining* program. It contains three five-week data mining courses: *Data Mining with Weka*, *More Data Mining with Weka*, and *Advanced Data Mining with Weka*, created by Professor Ian Witten from the University of Waikato. These courses are used to demonstrate the automated processes described in this chapter.

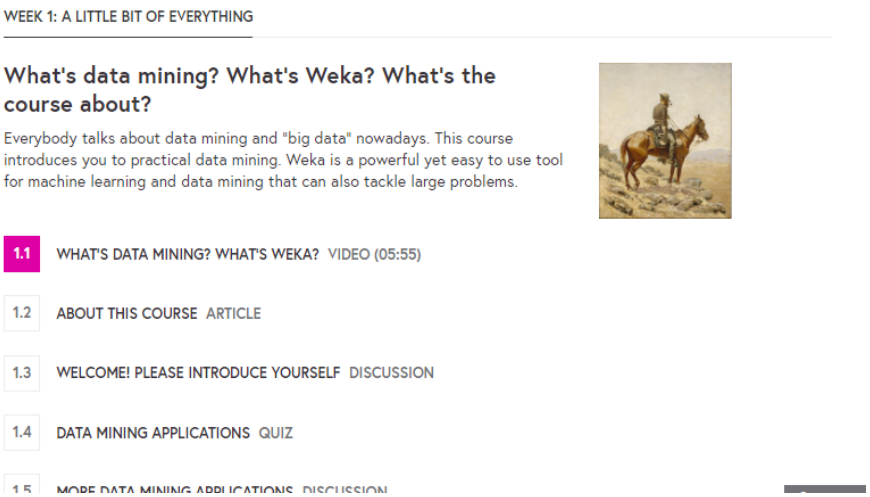
3.4 Online course structure

All FutureLearn courses are structured similarly. As previously mentioned, they are often between five and ten weeks long and include a selection of videos, articles, discussions, and quizzes, plus online assessments for those who pay to upgrade their enrolment. Course topics vary from "Maintaining a Mindful Life" to "Radiation Oncology".

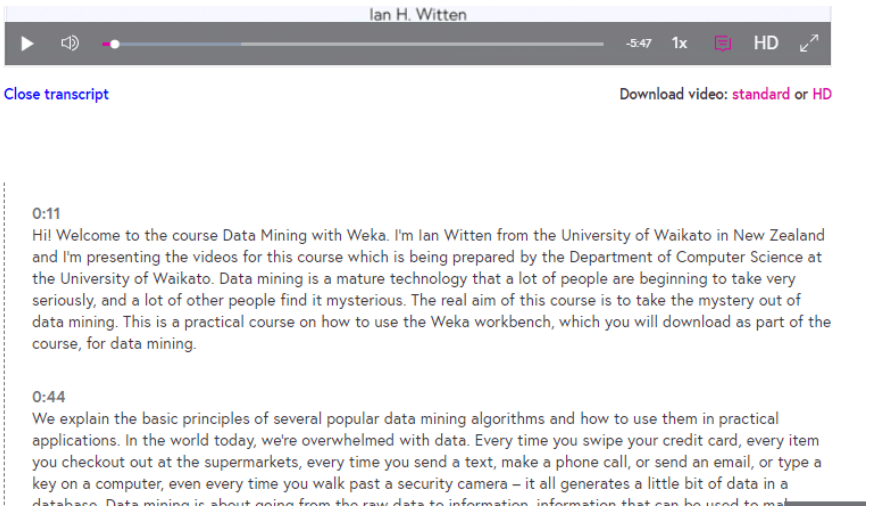
The *Data Mining with Weka* courses are aimed at anyone who deals with data and is interested in obtaining information from it. They are a set of practical data mining courses that teach learners how to use the Weka workbench to mine their own data. The first course, *Data Mining with Weka*, is an introductory course that is targeted toward beginners, while the two subsequent courses, *More Data Mining with Weka* and *Advanced Data Mining with Weka* introduce more advanced applications of the Weka workbench, plus additional topics in data mining and machine learning. The courses consist of both written English, in the form of articles and discussions, and spoken English, in the form of video tutorials.

3.4.1 Metadata

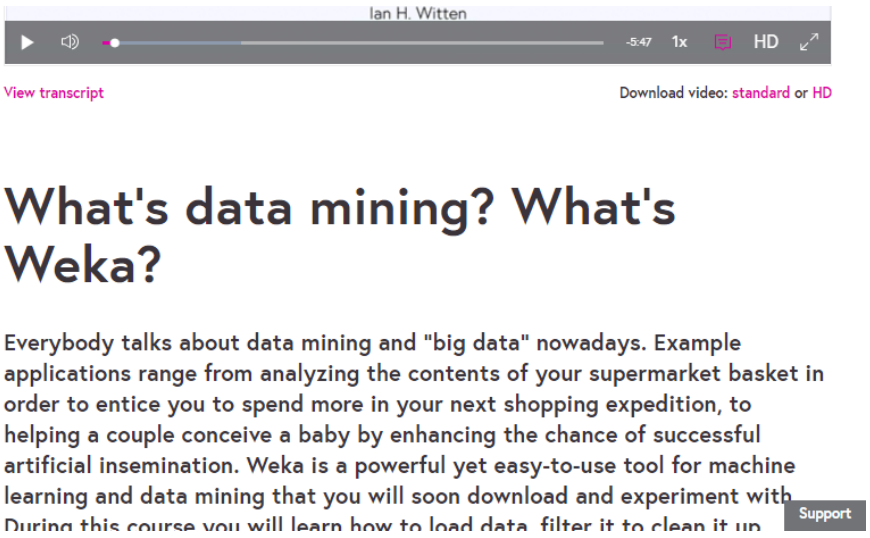
FutureLearn courses are broken into weekly blocks and are displayed as weekly 'To Do' pages. Each page includes small sections of explanatory text, plus a list of the steps that a learner is required to complete that week. Each step includes a step number, title, the type of page (video, article, discussion, quiz), and a clickable link that allows learners to navigate easily into the main content for that page. The *Data Mining with Weka* courses contain five weekly blocks with approximately twenty steps each. Figure 3.1a shows the start of the 'To Do' page for Week 1 of *Data Mining with Weka*.



a) Metadata: a weekly 'To do' page



b) Spoken content: video transcript text



c) Written content: a "Video" page

Figure 3.1 FutureLearn: metadata, spoken, and written content

3.4.2 Spoken content

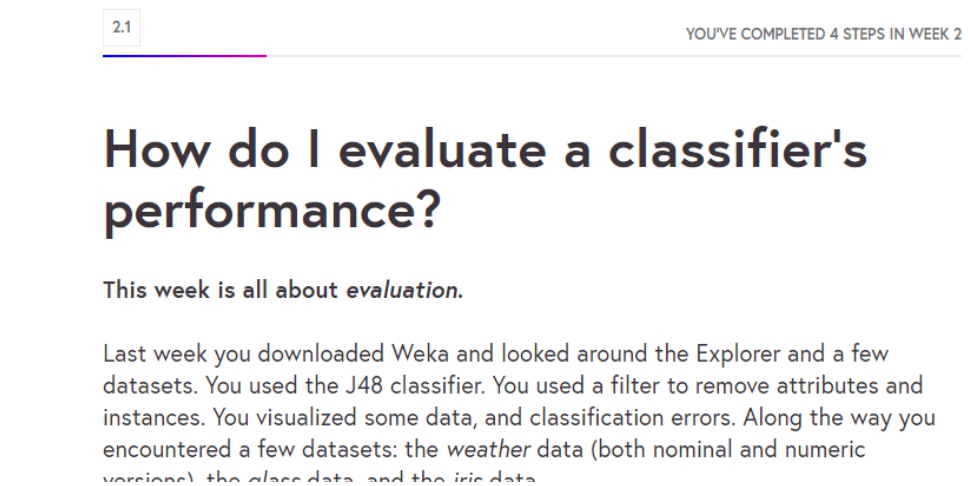
FutureLearn courses often include video tutorials and audio lessons. The *Data Mining with Weka* courses make extensive use of video tutorials, each course containing approximately 30 (distributed across the five weeks). Each video ranges between a few minutes and ten minutes long and includes a transcript of the spoken text. Figure 3.1b shows the beginning of one.

Course steps that include a video and transcript often also include some written text introducing the topic that is covered in the video and giving a description of the content or a set of instructions to follow. Figure 3.1c shows an example of the written text that appears on video tutorial pages.

3.4.3 Written content

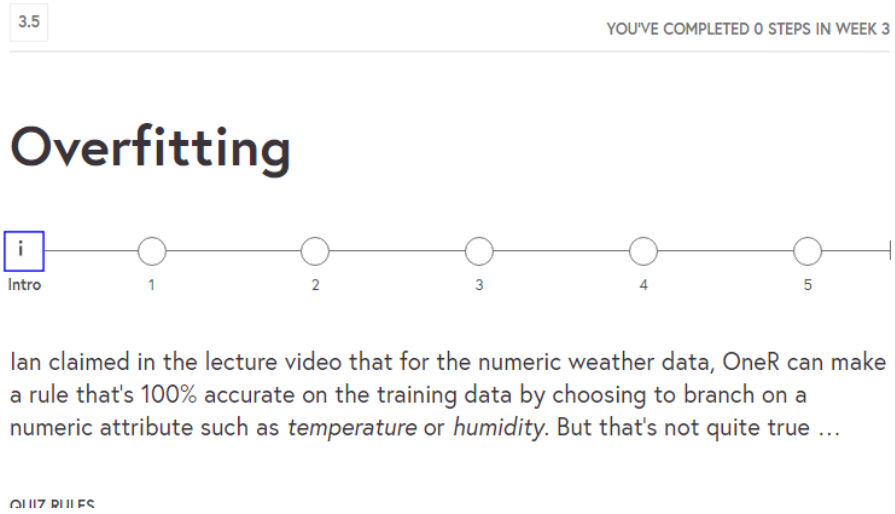
FutureLearn courses also include a variety of other pages, including articles, discussions, and quizzes. *Articles* and *discussions* provide learners with majority of their written text. Articles usually introduce a topic, elaborate on previously learnt material, or lead into the following steps. Figure 3.2a shows the start of an article. Discussions pose questions to learners or give them a topic that they are encouraged to discuss. Learners are encouraged to talk about ideas and concepts as they encounter them, as a way to learn and reiterate information. Figure 3.2b shows the start of a discussion page.

Quizzes challenge learners to test what they have learnt so far, and are scattered throughout the weekly steps. They include three different types of written text: an introduction, the questions themselves, and feedback from educators. Some introductions are longer than others, but each includes a title and small blurb about the quiz. They can also include leading dialog or instructions that need to be followed before starting the quiz. Figure 3.2c shows an example. After reading the introduction, learners can begin the quiz, which takes them to the first question. Figure 3.3a shows an example. This usually describes some operation that the learner needs to perform, plus a question they need to answer. Finally, when a learner submits their answer, they get a response that states whether they are correct or incorrect, which often includes some form of feedback from the lead educator. The feedback text either confirms that the answer is correct or gives an expanded explanation to help learners get the correct answer next time (shown in Figure 3.3b).



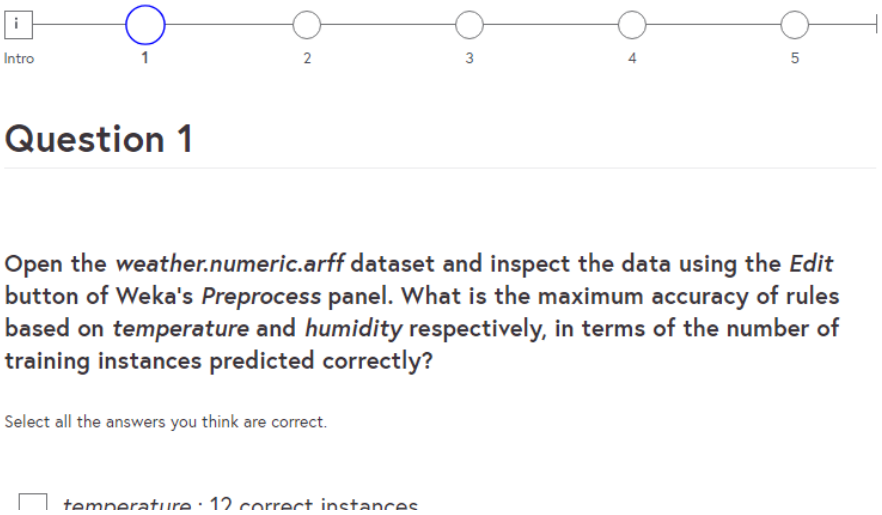
a) Article: “How do I evaluate a classifier’s performance”

b) Discussion: “Reflect on this week’s big question”

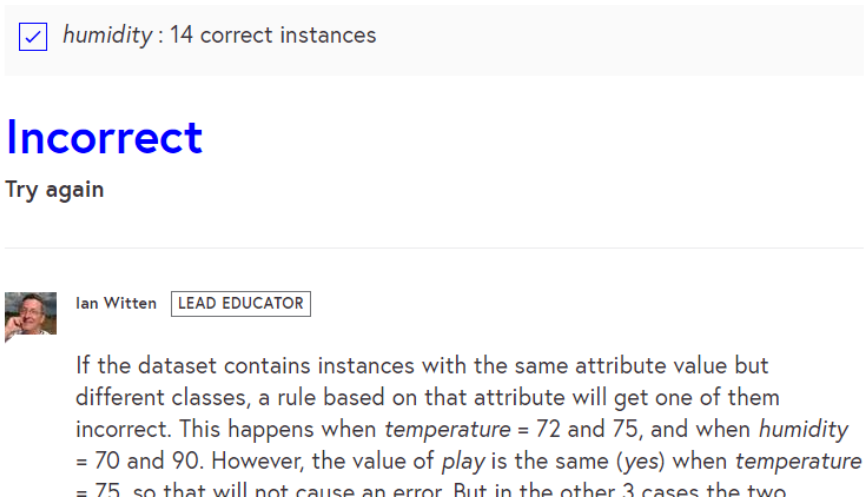


c) Quiz: “Overfitting”

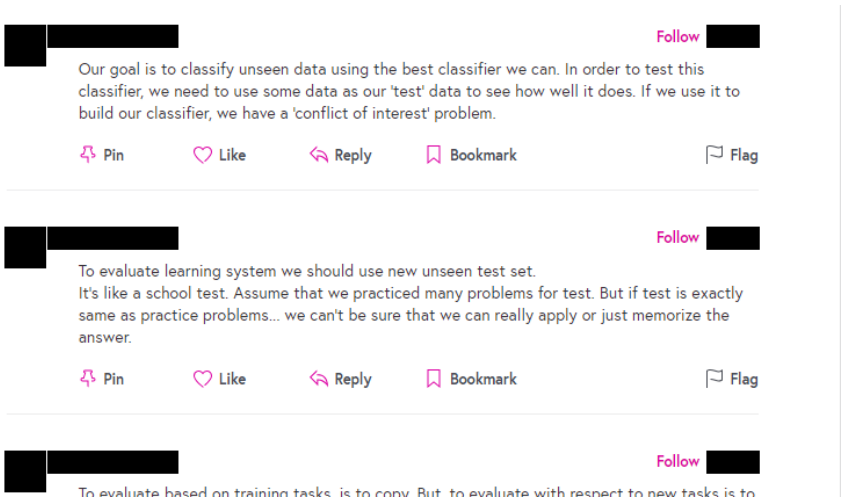
Figure 3.2 FutureLearn: articles, discussions, and quizzes



a) Quiz: Question 1



b) Quiz: educator's feedback



c) Posts: anonymised post examples

Figure 3.3 FutureLearn: quiz questions, quiz feedback, and learner posts

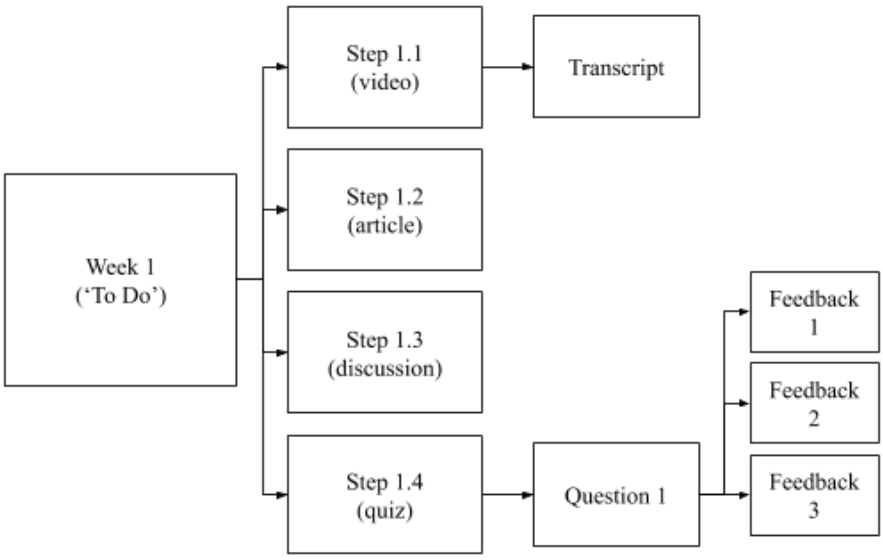


Figure 3.4 CourseCrawler: traversing online course content

Another substantial source of written text are learners' posts. As well as being encouraged to discuss concepts and topics throughout the course, learners also ask questions if they encounter any problems. Over 1500 comments were posted during the first run of Data Mining with Weka. Figure 3.3c shows some examples. Learners' posts may produce intriguing lexical information about informal written language, but they are not of interest for this work. We are only interested in course content created by educators, not comments posted by learners. Therefore, learners' posts will not be collected or included in the corpora created from online courses.

3.5 Automating a corpus

The rest of this chapter describes a set of automated processes for building domain-specific corpora and wordlists from online course content, and applies them to the *Data Mining with Weka* courses to build the *DMwW corpus* and *DMwW wordlist*. This involves using a Chrome extension to collect course content and using a Python application to divide the content into categories, annotate the corpus with additional lexical information, and analyse it to extract relevant information.

3.5.1 Text collection

A Chrome extension, called *CourseCrawler*[10], was developed for this thesis, which

[10] Download from https://github.com/jlkonig/Linguistic_Software/CourseCrawler.git.

Table 3.1 CourseCrawler: identifying page types in FutureLearn

Page type	Determiner
To Do	Base URL is followed by */todo/*
Article	None of the listed determiners are found
Discussion	Class name “discussion-comments-container” is found
Video	Class name “video-container” is found
Quiz	Class name “quiz-container” is found

extracts course content from the *Data Mining with Weka* courses. It works on top of FutureLearn to recursively follow links through a course, extracting content from the ‘To Do’ pages for each week, plus every video, transcript, article, discussion, and quiz. Figure 3.4 shows an example, starting at the ‘To Do’ page, and following links for the first four steps.

To start processing content, the CourseCrawler takes the name and run number for a course and appends them to the base FutureLearn URL, retrieving the Week 1 ‘To Do’ page. For example, for *Data Mining with Weka*, the course name is *data-mining-with-weka* and the current course run is 5, resulting in the URL `www.futurelearn.com/courses/data-mining-with-weka/5/todo/`. This URL is referred to as the *starting seed* and is used by the CourseCrawler to recursively crawl courses, extracting content from each page as it goes.

Only URLs that match the following conditions are crawled. They must: (i) start with the base URL, (ii) have not been seen by the CourseCrawler before, (iii) and must end in either *todo*, *step*, or *quizzes*, for example, the URL `www.futurelearn.com/courses/data-mining-with-weka/5/todo/` would be crawled, but the URL `www.futurelearn.com/courses/data-mining-with-weka/5/activity-feed/` would not. This prevents the extension from extracting unwanted content from other areas of the course, such as learners’ comments and learners’ statistics.

FutureLearn pages include substantial amounts of additional content including the head banner, navigation links outside the current course, additional information about FutureLearn, and links to social media. The CourseCrawler ignores all such material and focuses only on course-related content. To achieve this, a set of tightly specify restrictions were outlined, determining acceptable content rather than simply extracting all content from a page. Each type of page

Code Example 3.1 CourseCrawler: extracting page titles and steps

```
var el          = document.createElement( 'html' );
el.innerHTML    = response;

// Extract step number
var stepNumEl   = el.getElementsByClassName("a-stepnumber")
var stepNumText = trimNewlines(stepNumEl[0].textContent);

// Extract title
var titleEl     = el.getElementsByClassName("a-article-h1")
var titleText   = titleEl[0].textContent.split("\n")[1];
```

(video, article, discussion, and quiz) is structured differently, and therefore has to be processed differently. The CourseCrawler identifies page types using the HTML classes illustrated in Table 3.1.

'To Do' pages are structured as lists of activities. The CourseCrawler extracts content by finding HTML elements that are tagged with the class name "activities list", several of which often occur on one page. It concatenates the content from each element into a continuous string, referred to as the *content string*.

Video content is processed in two steps. First, written content is extracted by finding HTML elements that are tagged with the class name "a-text-content". Then, video transcript content is extracted by finding elements with the class name "transcript__para". Both types of content are concatenated into a content string and the spoken transcript is bracketed with <start transcript> and <end transcript> tags to distinguish it from written content.

Articles and discussions are structured similarly and are the simplest to process. Content is extracted from each by finding HTML elements that are tagged with the class name "a-text-content".

Quiz pages are structured differently. 'To Do' pages, videos, articles and discussions are all static pages and the HTML contains all of their content. Quiz pages use JavaScript to move between questions, to validate answers, and to give feedback. Instructors often give feedback on both correct and incorrect answers, and when there is more than one incorrect answer there are often multiple feedback responses. This means that each answer for each question must be traversed, in order to return every possible feedback response. To extract content from one quiz question, the CourseCrawler extracts the question text using the "quiz-item" class name, extracts the IDs of all possible answers using a query selector, and sends a POST request to return the feedback text for each answer ID. It then sends a GET

Table 3.2 CourseCrawler: content extracted from *Data Mining with Weka*

Page type	Example
To Do	<-- 1.0 Todo --> What's data mining? What's Weka? What's the course
Articles	<-- 1.2 Article --> This course introduces you to practical data mining
Discussions	<-- 1.3 Discussion --> Hey! I'm Ian. Formally I'm Professor Ian Witten
Videos	<-- 1.1 Video --> Everybody talks about data mining and "big data"
Transcripts	<Start Transcript> I'm Ian Witten from the University of Waikato in New
Quizzes	<-- 1.4 Quiz --> Question 1 Can data mining be applied to

request to move to the next quiz question. The content from each quiz question, possible answers, and feedback response are concatenated into a content string.

Finally, the CourseCrawler extracts the page title using the "a-article-h1" class, and extracts the step number using the "a-stepnumber" class, as demonstrated in Code Example 3.1. These are appended to the start of each content string.

Once all pages have been traversed, the CourseCrawler saves the content of an entire course as one single text file. Within the file, each page's content is included in descending order, by step number, and begins with a tag specifying its page type. Table 3.2 shows an example of how each type is represented within the resulting text file.

3.5.2 Structuring

A Python script, called the *CourseCorpusBuilder*[11], has been developed to automatically structure, annotate, and analyse corpora using the text extracted from courses by the CourseCrawler. It structures a corpus by using the tags shown in Table 3.2 to split course content into a series of individual files corresponding to each page, each of which is classified into one of six categories: written-todo,

[11] Download from https://github.com/jlkonig/Linguistic_Software/CourseCorpusBuilder.git.

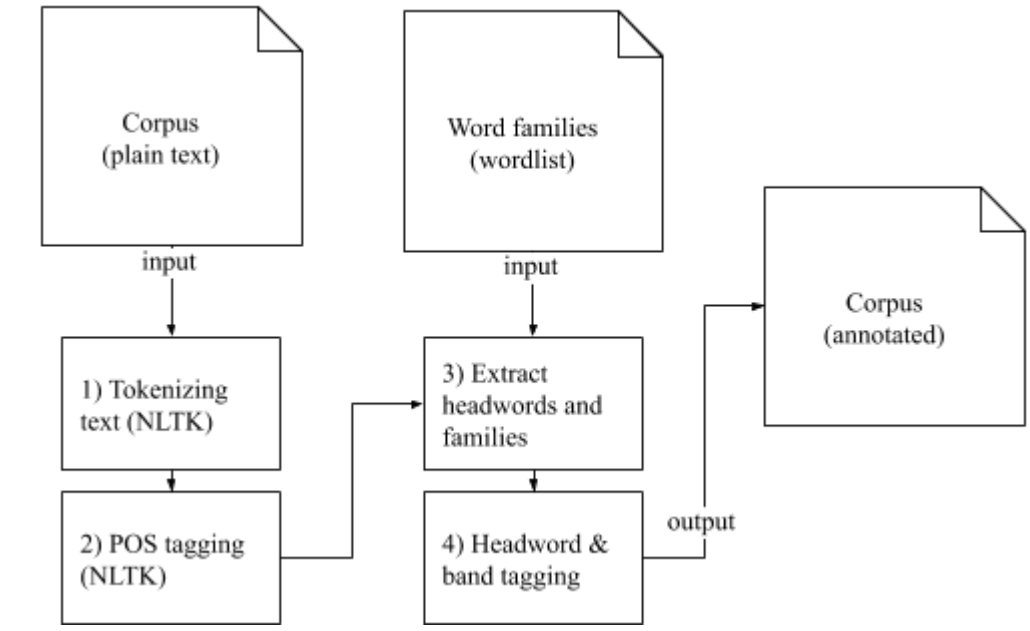


Figure 3.5 CourseCorpusBuilder: annotating a corpus

written-article, written-discussion, written-video, spoken-transcript, and written-quiz.

3.5.3 Annotating

Next, the CourseCorpusBuilder annotates corpora with headwords, part of speech, and frequency bands. The first is used later in this chapter to create domain- specific wordlists, the second can be used to help identify collocations, and the third can be used to compare domain-specific and general-purpose language.

Corpus annotation is often done programmatically, using natural language processing software to tag large amounts of text quickly and easily. The CourseCorpusBuilder takes advantage of this by using the Natural Language Toolkit (NLTK) (Loper & Bird, 2002). Figure 3.5 illustrates the process. First, it uses the NLTK tokenizer to break strings into substrings containing words and punctuation. Code Example 3.2 illustrates tokenizing words, punctuation, and contractions using the NLTK tokenizer. The NLTK tokenizer ignores newlines. To circumvent this, the CourseCorpusBuilder breaks each text file into paragraphs first, tokenizing each paragraph separately.

Next, the CourseCorpusBuilder annotates each token with part-of-speech, again using NLTK. By default, NLTK uses the Penn Treebank tagset (Santorini, 1990) . It takes a list of tokenized words and, using the TreebankWordTokenizer

Code Example 3.2 NLTK: tokenizing

```
>>> from nltk.tokenize import word_tokenize
>>> s = "What's data mining? What's Weka?"
>>> word_tokenize(s)
['What', "'s", 'data', 'mining', '?', 'What', "'s", 'Weka', '?']
```

Code Example 3.3 NLTK: part-of-speech tagging example

```
>>> from nltk.tokenize import word_tokenize
>>> from nltk import pos_tag
>>> s = "What's data mining? What's Weka?"
>>> t = word_tokenize(s)
>>> pos_tag(t)
[('What', 'WP'), ("'s", 'VBZ'), ('data', 'NN'), ('mining',
'NN'), ('?', '.'), ('What', 'WP'), ("'s", 'VBZ'), ('Weka',
'NNP'), ('?', '.')]
```

Code Example 3.4 CourseCorpusBuilder: headword tagging example

```
>>> from nltk.tokenize import word_tokenize
>>> from familiarizer import familiarize
>>> s = "What's data mining? What's Weka?"
>>> t = word_tokenize(s)
>>> familiarize(t)
[('What', 'what', 1), ("'s", 'be', 1), ('data', 'data', 3),
('mining', 'miner', 3), ('?', '?', 0), ('What', 'what', 1),
 ("'s", 'be', 1), ('Weka', 'weka', 23), ('?', '?', 0)]
```

Code Example 3.5 CourseCorpusBuilder: annotation example

Original text:	“What’s data mining? What’s Weka?”
Tagged text:	What/WP/what/1 's/VBZ/be/1 data/NN/data/3 mining/NN/miner/3 ?/./?/0 What/WP/what/1 's/VBZ/be/1 Weka/NNP/weka/23 ?/./?/0

(Bird, Tan, & Nothman, 2018), returns a list of tokens with their corresponding part-of-speech tag. See Code Example 3.3.

Finally, the CourseCorpusBuilder annotates corpora with headwords and frequency bands. The former can be done using either stemming, lemmatization, or word families. The General Service List (West, 1953), the Academic Word List (Coxhead, 2000), and the Range Programme Lists (Nation et al., 2002) all use word families rather than stems or lemmas, and since each is referenced and used throughout this thesis, the CourseCorpusBuilder uses the same. However, annotating words with their headwords using word families is not as simple as using an existing stemming and lemmatizing library. Stemming and lemmatization are

done through a systematic process, as discussed in Section 2.1.1, and are both available through NLTK.

Conversely, word families are determined not just by the form of the word, but also its semantics. For example, as discussed in Section 2.1.1, the words *hard* and *hardly* are spelt similarly but have different meanings, therefore are not members of the same word family. The fact that word families take semantics into account make them much more difficult to determine programmatically. As a solution, the CourseCorpusBuilder includes a small algorithm that matches word forms to an existing list of word families and their corresponding frequency bands. However, given the semantic constraint on word families, only words that match those in the pre-defined list are annotated this way. The rest are annotated using NLTK to retrieve their lemmas. Lemmatization was chosen over stemming because it takes both morphology and part-of-speech into account, rather than pattern matching affixes.

The CourseCorpusBuilder uses the Range Programme Lists, which is a collection of 25 frequency-based lists, each containing 1000 word families (Nation et al., 2002). It builds a Python dictionary of word families, where the *key* is the word form (i.e. abilities) and the *value* is an array containing: (i) the headword (i.e. able), and (ii) its frequency band (the number of the list that the family originated from). Once the dictionary has been created, it can be used to map running words (tokens) in the corpus with their corresponding word families and headwords. If the word does not exist in the dictionary, then the NLTK lemmatizer is used to retrieve the lemma for that word instead. In the latter case, the frequency band is set to 0. Code Example 3.4 demonstrates annotating running words with their headwords.

It should be noted that there are cases where the NLTK lemmatizer does not successfully lemmatize a word. In these cases, the word is tagged with itself, assuming it is its own lemma. For some words, this is correct. However, for others, it is not. The word “updatable”, for example, is not included in the word family for “update” (Nation et al., 2002), this means that it will be lemmatized by NLTK, but the NLTK lemmatizer does not lemmatize “updatable” to be “update”, instead it is tagged with itself as the headword. These words are marked with a frequency band value of 0 so researchers can check them manually (if they wish). This is discussed further in Section 3.7, since they may affect the words included in wordlists.

Table 3.3 CourseWordlistBuilder: word selection criteria

Criteria	**Details**
Range	Must occur in both spoken and written English within the corpus, and in at least 50% of categories (article, discussion, video, etc.).
Frequency	Must make up at least 0.00286% of the corpus. In the AWL, it had to occur 100 times in the 3,500,000 word corpus (0.00286%).
Uniformity of frequency	Must make up at least 0.00114% of spoken and 0.00114% or written words. In the AWL, it had to occur at least 10 times in each of the overarching categories (0.00114%).
Excluding GSL	Word families that occur in the first 2,000 most frequent words in English (according to the GSL) will be excluded.
Excluding others	Proper nouns (i.e. “Waikato”) and Latin forms (i.e. “etc.”) should be manually excluded.

Applying the CourseCorpusBuilder to a corpus results in a collection of categorized files, relating to each course page, annotated with part of speech, headwords, and frequency bands. Code Example 3.5 shows an example of annotated text. Each lexical attribute is punctuated with a forward slash, for example, `running-word/part-of-speech/headword/frequency-band`.

3.5.4 Analysing

Finally, the CourseCorpusBuilder analyses the annotated corpus to extract information about word frequency and range. It iterates over every word in the corpus, counting occurrences of headwords (frequency) and the number of texts the headword appears in (range). Finally, it outputs a text file that lists headwords with their corresponding frequency and range counts.

Table 3.4 DMwW corpus: category distribution

	Files (by course)		Running words (by course)	
written-todo	15	(5+5+5)	5,782	(1753+2065+1964)
written-video	90	(30+30+30)	8,983	(2909+2895+3179)
spoken-transcript	90	(30+30+30)	103,916	(31,144+34,339+38,433)
written-article	68	(15+27+26)	15,461	(3115+7383+4963)
written-discussion	43	(14+15+14)	6,516	(2066+2647+1803)
written-quiz	95	(31+32+32)	66,174	(16,893+21,594+27,687)
TOTAL	401	(125+139+137)	206,832	(57,880+70,923+78,029)

3.6 Automating a wordlist

A domain-specific wordlist can be generated from a domain-specific corpus. It can be used to identify words that occur more frequently within a particular domain than in general-purpose language, aiding teachers and learners with vocabulary acquisition within that domain. A Python script, the *CourseWordlistBuilder*[12], has been developed to automatically build domain-specific wordlists from domain-specific corpora. The process involves four steps.

1. Splitting the tagged corpus into individual words and extracting the headword for each
2. Excluding words that do not meet the selection criteria (Table 3.3)
3. Computing the frequency for each word
4. Computing word range

The selection criteria are based on those for the Academic Word List (described in Section 0), for example, for a word to be included in the AWL, it had to occur: 100 times in the 3,500,000 word corpus, 10 times in each of the four over-arching categories (arts, commerce, law, and biology), and at least once in at least half of the twenty-eight subcategories (education, history, linguistics, philosophy, etc.).

3.7 The DMwW corpus and wordlist

The CourseCrawler, CourseCorpusBuilder, and CourseWordlistBuilder have been applied to the *Practical Data Mining* project on FutureLearn, which contains three

[12] Download from https://github.com/jlkonig/Linguistic_Software/CourseWordlistBuilder

Table 3.5 DMwW corpus: language distribution

	Files (by course)		Running words (by course)	
Written	311	(95+109+107)	102,916	(26,736+36,584+39,596)
Spoken	90	(30+30+30)	103,916	(31,144+34,339+38,433)
TOTAL	401	(125+139+137)	206,832	(57,880+70,923+78,029)

Table 3.6 DMwW corpus: frequency and range

Frequency			
correct (1935)	go (1380)	question (1071)	instance (1032)
data (1726)	dataset (1104)	answer (1053)	class (960)
attribute (1623)	get (1075)	weka (1074)	set (872)
Range			
data (272)	one (219)	classifier (210)	different (181)
weka (259)	set (212)	learn (190)	get (179)
attribute (252)	dataset (210)	new (184)	instance (172)

data mining courses: *Data Mining with Weka*, *More Data Mining with Weka*, and *Advanced Data Mining with Weka*. First, course content was extracted from FutureLearn using the CourseCrawler extension. It was done this way, rather than using Witten's original source files, in order to demonstrate that the CourseCrawler can be applied to any FutureLearn course. This resulted in three text files, one for each of the data mining courses.

Next, the CourseCorpusBuilder was applied to the files, resulting in the creation of the *DMwW corpus*[13]. The CourseCorpusBuilder first structured the corpus, generating a total of 401 files, split into six categories, as shown in Table 3.4. The first course, *Data Mining with Weka* contributed the smallest number of files, at 125. The second, *More Data Mining with Weka* had 139 files, and the third, *Advanced Data Mining with Weka*, had 137.

The DMwW corpus is not evenly distributed between categories. The largest category, written-quiz, had 95 texts, while the smallest, written-todo, only had 15. This is not surprising, since each course runs for five weeks, with one 'To Do' page per week. Table 3.4 also shows the running word count for the DMwW

[13] Download from https://github.com/jlkonig/Language_Resources/DMwW-Corpus

Table 3.7 DMwW wordlist: most frequent words

data (1722)	classifier (824)	select (495)	bayes (380)
attribute (1590)	feedback (736)	predict (447)	naive (374)
weka (1051)	accurate (587)	filter (444)	miner (364)
instance (1032)	different (540)	method (433)	classification (360)
dataset (973)	evaluate (517)	error (407)	parameter (282)

corpus. Like the number of texts, the running word count is not evenly distributed between categories. The largest category, spoken-transcript, has 103,916 running words, while the smallest, written-todo, only has 5,782.

Table 3.5 shows the distribution of text by spoken and written English. Spoken English is derived from spoken-transcripts, and has a running word count of 103,916 words, while written English has a running word count of 102,916 and is derived from the rest. Although the categories themselves are not evenly distributed, the corpus itself, in terms of spoken and written English, is.

Once the text within the corpus was structured and categorized, the CourseCorpusBuilder annotated it, as illustrated earlier in Code Example 3.5. Then the CourseCorpusBuilder analysed the corpus, returning a wordlist with frequency and range counts. Table 3.6 shows the ten words with the highest frequency and range (excluding function words). Four words are present in the top ten for both frequency and range, *data*, *attribute*, *Weka*, and *dataset*, each of which reflect the *Data Mining with Weka* domain.

Finally, generating the *DMwW wordlist*[14] from the corpus provides learners with a set of domain-specific words that could be used to create target wordlists, domain-specific pseudowords (Chapter 4), or domain-specific vocabulary tests (Chapter 5). The CourseWordlistBuilder was applied to the annotated corpus, resulting in a list of 571 domain-specific words. As per the wordlist selection criteria described in Section 3.6, each word occurred: in at least three out of six categories (50%), at least seven times in the corpus (0.00286%), and at least twice in spoken and twice in written English (0.00114%).

[14] Download from https://github.com/jlkonig/Language_Resources/DMwW-Wordlist.git.

Table 3.7 shows the twenty most frequent words in the DMwW wordlist, while Appendix A shows the full list. Although the wordlist-building process is automated, researchers may still wish to manually check and alter it in ways that suit their needs, particularly in cases where words were not found in word families and were not lemmatized by NLTK (i.e. "updatable"), as mentioned in Section 3.5.3. In this case, some proper nouns were manually excluded. However, others were not. Place and people's names (*Waikato*, *New Zealand*, *Ian Witten*) were excluded, while names of software, classifiers, and concepts within the field were not (*Weka*, *zeroR*, *SMO*). The latter are important terms for learners to fully understand the content of the courses. The DMwW wordlist, combined with the General Service List, covers 85% of text in the DMwW corpus. This is similar to the Academic Word List, which, when combined with the General Service List, covers 86% of the text in the Academic Corpus (Coxhead, 2000, p. 225).

3.8 Implications

This chapter has introduced a set of automated processes for creating domain-specific corpora and wordlists (*CourseCrawler*, *CourseCorpusBuilder*, and *CourseWordlistBuilder*), and two resources have resulted from it (*DMwW corpus* and *DMwW wordlist*). Although the processes were applied to a specific set of courses, they have been developed as a general solution for creating corpora and wordlists from online course content and can be applied to any course on FutureLearn.

This concludes the first challenge, identifying domain-specific vocabulary and automating the process of building domain-specific corpora and wordlists. The next chapter begins investigating the first half of the second challenge, generating domain-specific pseudowords for use when automating receptive vocabulary tests.

Chapter 4

Generating pseudowords

Pseudo /ˈsjuːdəʊ/

(adj) (1) other than what is apparent; spurious; sham. (2) a poseur; one who is fake.

(Wiktionary, 2019)

Pseudowords play an important role in lexical processing research and language teaching. They are used in receptive vocabulary tests (Meara, 1992), in phonetic decoding (Cardenas, 2009; Groff, 2003), in measuring pronunciation latency (Schwartz, 2013), and in visual word recognition (Balota et al., 2007).

Chapter 5 describes the development of a domain-specific version of the EFL Vocabulary test, a test that uses words and pseudowords to measure learners' vocabulary. However, a domain-specific version of the test cannot be developed without a set of domain-specific pseudowords. This chapter provides a practical and empirically founded approach to generating and evaluating domain-specific pseudowords. The first half of the chapter sets out a novel way of generating pseudowords – a character-gram chaining algorithm whose major advantage is that

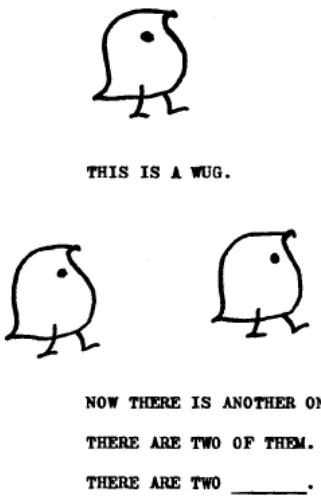


Figure 4.1 One of the hand-drawn panels from the original WUG test

it requires no linguistic knowledge, thereby facilitating the generation of pseudowords in any language. The second half offers a set of linguistic criteria for evaluating pseudowords – both in terms of their legal orthographic form and their suitability for different lexical tasks.

4.1 Pseudowords

A *pseudoword* is a unit of text or speech that has no meaning in a language but conforms to the usual orthographic and phonological structure (Nordquist, 2018). It has the form of a word and is spelled in a predictable way, but does not exist in the lexicon (Groff, 2003). Pseudowords are used by linguistic researchers to test language production processes in morphology experiments, and by applied linguists and language teachers in non-native vocabulary tests, as a means of assessing the credibility of learner's responses.

Perhaps the most famous pseudoword of all, *WUG*, was used in morphology experiments to test plural rule productivity (see Figure 4.1) (Berko, 1958). Berko (1958) demonstrated children's implicit knowledge of linguistic morphology, using pseudowords to evaluate knowledge of morphological rules, proving that children can form suitable endings, for example producing plurals, possessives, past tense, and so on. Receptive vocabulary tests also use pseudowords, but as a means of assessing the credibility of learner's responses. Meara's (1992) EFL Vocabulary Test assesses vocabulary size and incorporates pseudowords to help validate learners' responses.

Table 4.1 Manipulating the stimulus *pilot*

Insertion	Deletion	Transposition	Composition
piloth	*pilt*	*pliot*	*pilotation*

Table 4.2 Combining high frequency bi-grams to create the pseudoword *reroin*.

bigram:	*re*	*er*	*ro*	*oi*	*in*
frequency:	4760	7279	2840	468	7156

Table 4.3 Combining sub-syllabic elements to create the pseudoword *shib*

Onset	Nucleus	Coda	Pseudoword
š (as in *show*)	ɪ (as in *tin*)	b (as in *bib*)	š-ɪ-b = *Shib*

While the WUG test requires polymorphic pseudowords that include meaningful affixes (plural, possessive, past tense, etc.), Meara's test requires pseudowords whose meaning cannot be inferred. It assesses the credibility of a learner's response rather than testing the implicit knowledge of language rules. This suggests that different types of pseudowords may be better suited for different types of lexical task.

4.2 Techniques and applications

Three main techniques are used to generate pseudowords: manipulating a stimulus, using high frequency bi-grams, and combining sub-syllabic elements. The first starts with a stimulus word and manipulates it in some way to create a pseudoword. The stimulus can be altered by changing one or two characters, either by insertion, deletion, transposition, or replacement. Alternatively, a composite pseudoword can be created by adding a prefix or suffix to the stimulus, so long as it does not form a real word (R Harald Baayen & Schreuder, 2011). Table 4.1 demonstrates how the stimulus *pilot* can be manipulated to form pseudowords.

The second, using high frequency bi-grams, involves combining two-letter sequences that appear together frequently to form pseudowords. This technique is often augmented by neighbourhood size, orthographic relatedness, and tri-gram frequency. Table 4.2 demonstrates using WordGen (described in Section 4.7.1) to combine high frequency bi-grams to create the pseudoword *reroin*.

Finally, combining sub-syllabic elements involves breaking existing syllables into their sub-syllabic elements and joining them back together to form pseudowords. A syllable is a unit of sound, typically made up of a nucleus (usually a vowel) and an optional onset (initial sound) and coda (final sound). This approach takes an onset, nucleus, and coda from existing words and combines them to form pseudowords (Keuleers & Brysbaert, 2010). Table 4.3 demonstrates combining sub-syllabic elements to create the pseudoword *shib*.

Linguists and language teachers often select pseudowords from existing databases or use software applications to generate new ones. Four such applications are as follows:

1. *The English Lexicon Project* (Balota et al., 2007) is a database of 40,000 stimulus words and the same number of pseudowords. Pseudowords are created by changing one or two characters in each stimulus word, alternating the location of the manipulated characters between words.
2. *The ARC Nonword Database* (Rastle, Harrington, & Coltheart, 2002) holds a collection of 350,000 monosyllabic pseudowords, which are created by joining sub-syllabic elements, combining onset, nucleus, and coda.
3. *WordGen* (Duyck, Desmet, Verbeke, & Brysbaert, 2004) is a software application that uses bigram frequency for pseudoword generation in English, Dutch, German, and French. It creates pseudowords by generating random collection of letters which are accepted only if they are absent from the lexicon and meet a set of seven constraints including summated, minimum, initial, and final bigram frequency.
4. *Wuggy* (Keuleers & Brysbaert, 2010) is a sub-syllabic pseudoword generator that takes a list of syllabified words, segments each word into sub-syllabic elements, and builds a tree of all possible legal sub-syllabic combinations. The tree is then traversed to retrieve all possible pseudowords.

4.3 Limitations

Each method for generating pseudowords has its own set of limitations, either in the method itself, in its performance, in language restrictions, or in the lack of any empirically founded evaluations, for example, *manipulating a stimulus* requires knowledge of which characters can be inserted, deleted, or transposed while still

resulting in a legal pseudoword. If the stimulus is being manipulated manually, success depends on the creator's experience and language knowledge. Similarly, using *high frequency bi-grams* requires researchers to be aware of which character combinations exist in the language, for example, an English word can begin with *pl* (as in play) but cannot begin with *mz* (Rastle et al., 2002). *Combining sub-syllabic elements* requires knowledge of syllabification, and if phonetic syllables are used, as in the case of the ARC Nonword Database, then the ability to convert from phonetic to orthographic forms is needed.

Combining letter sequences or sub-syllabic elements can result in an overwhelming number of pseudowords; monosyllabic words have hundreds of thousands of possible combinations, while polysyllabic words have billions (Keuleers & Brysbaert, 2010). To alleviate this, some applications provide building constraints or search criteria which restrict the pseudowords that are returned. However, this creates limitations. WordGen, for example, asks the user to specify the number of neighbours, word frequency, and summated bigram frequency. This makes searching for pseudowords much more achievable, but can result in more complex software applications that can be confusing for researchers to interact with.

The lexicons used to create pseudowords can also become a limitation, for example, two lexicons are used predominately in the field of pseudoword generation: CELEX and Lexique. *CELEX* (R H Baayen, Piepenbrock, & van Rijn, 1993) is a lexical database that contains information on orthography, phonology, morphology, syntax, and word frequency. It is used by all four databased and software applications. *Lexique* (New, Pallier, Brysbaert, & Ferrand, 2004) is a lexical database for the French language that contains information on gender, grammatical category, and word frequency, and is used by both of the systems that support French (WordGen and Wuggy). Both CELEX and Lexique are general-purpose lexicons, meaning that they generate general-purpose pseudowords, and do not support domain-specific generation.

Table 4.4 shows the languages and lexicons used by the English Lexicon Project, ARC Nonword Database, WordGen, and Wuggy. Two are unilingual and only supporting English; one supports four languages, English, Dutch, German, and French; and one – Wuggy – supports seven. Wuggy also has the capacity to be extended to support any alphabet-based language, but it requires a list of syllabified

Table 4.4 Lexicons and languages used by pseudoword generation software

System	Lexicon	Language
ARC	CELEX	English
ELP	CELEX, Kučera and Francis	English
WordGen	CELEX	English, Dutch, German
	Lexique	French
Wuggy	CELEX	English, Dutch, German
	Lexique	French
	E-HITZ	Spanish
	B-PAL	Serbian
	The Frequency Dictionary of Contemporary Serbian Language	Basque

words in the desired language and information about how the syllables are segmented. This highlights a substantial gap in the field – there is a real need to develop a system that can easily handle a wider range of languages, and that can generate more than just general-purpose pseudowords.

Finally, all these applications strive to generate plausible pseudowords, but none of them conduct any form of post-evaluation, either for legal structure or suitability for lexical tasks. Each has a different approach to generating pseudowords, and different criteria/grammars/principles for restricting the forms of the pseudowords that are generated. Perhaps more importantly, they have differing views on what constitutes a legal pseudoword. The ARC database focuses on phonological principles, allowing illegal bigrams, while Wuggy focuses on orthographic forms. Each approach mentions the importance of suitable pseudowords, but there is no evidence of their suitability having been evaluated. Are the pseudowords that are being generated too similar to real words, or not similar enough? Are they within the range of what is required for lexical tasks, and do different lexical tasks require more, or less, wordlikeness?

The remainder of this chapter proposes a new approach to creating pseudowords that is not susceptible to any of the limitations of the existing approaches, and proposes a way of conducting post-creation evaluations on pseudowords. I start by introducing the character-gram-chaining algorithm, which is a computationally simple approach to generating pseudowords. Then, I

Table 4.5 Order *n*–1 models of generated text

Order 0 text (single character)	*fsn'iaad ir lntns hynci,..aais oayimh t n ,at oeotc fheotyi t afrtgt oidtsO, wrr thraeoe rdaFr ce.g*
Order 5 text (6-gram)	*Number diness, and it also light of still try and amoung Presidential discussion is department-trans*
Order 11 text (12-gram model)	*Papal pronounced to the appeal, said that he'd left the lighter fluid, ha, ha"? asked the same numbe*

describe a set of evaluation criteria, designed to evaluate both the legal form of pseudowords and their suitability for lexical tasks. Finally, I will show that these criteria can be used to compare pseudowords that have been generated by different applications, and demonstrate how the character-gram chaining algorithm can be used to create domain-specific pseudowords without requiring any more knowledge of the language or domain than a simple wordlist.

4.4 The character-gram chaining algorithm

Bell, Cleary, and Witten (1990) used statistical analysis of n-gram frequencies to model natural language for text compression, and suggested that n-grams could be used to predict the *n*th character from the preceding *n*–1 characters (Bell et al., 1990). Table 4.5 demonstrates text generated by 1-grams, 6-grams, and 12-grams. Although the models are far from perfect, the resemblence to natural language improves noticeably for each increase in *n*. This has led us to explore whether a similar technique could be used to construct individual pseudowords, chaining character-gram together to form pseudowords.

The Character-gram Chaining Algorithm (CGCA) was developed for this thesis as a novel way of generating pseudowords, chaining character-gram together based on their probability of appearing within the language. Figure 4.2 demonstrates an overview of the algorithm, broken down into four steps. Given a corpus or wordlist as input, the CGCA: (1) builds an origin wordlist, (2) extracts all possible character-grams, (3) chains character-grams back together to form pseudowords, and (4) validates the pseudowords against a lexicon.

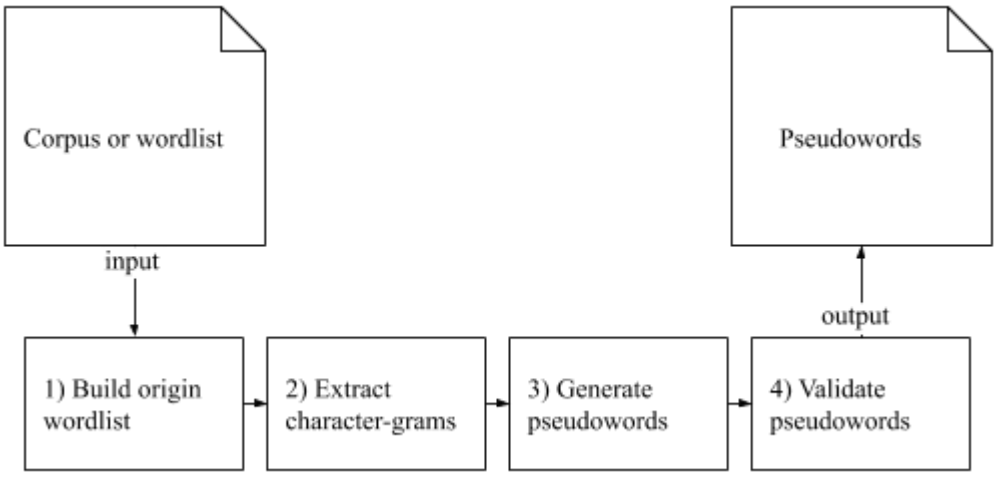


Figure 4.2 CGCA: the steps involved in the character-gram chaining algorithm

4.4.1 Building an origin wordlist

Any corpus or wordlist can be used as input. However, they may include spelling mistakes, duplicate words, non-words, or partial words. For this reason, the CGCA's first step is to build an origin wordlist from an input corpus or wordlist, and clean it. This involves the following.

1. *Breaking the input into tokens*: determined by the NLTK Tokenizer.
2. *Cleaning tokens*: converting all tokens to lower case, removing duplicates, and removing tokens that included non-alpha characters. This results in a list of unique tokens (types).
3. *Validating types*: using Wiktionary to determine if a type is a real word.

4.4.2 Extracting character-grams

Once an origin wordlist has been created, CGCA extracts character-grams from each of the types. It extracts 2-grams, 3-grams, up to 8-grams, and each character-gram is stored along with three counts.

1. *Initial*: the number of times it appears at the start of a word
2. *Final*: the number of times it appears at the end of a word
3. *Total*: the number of times it appears at the start, middle, or end of a word

These counts are updated each time a character-gram is extracted and are used later by the CGCA to identify initial and final character-grams, and to probabilistically determine which character-grams should be chained back together to create a

Code Example 4.1 CGCA: extracting character-grams

```
FOR each type in the origin wordlist
  WHILE not at the end of the type:
    Extract a character sequence of length n
    IF the character sequence has not previously been seen:
       Save the character sequence with an initial, final,
       and total count.
    ELSE
       Update the initial, final and total count.
    END IF
    Move along one character
  END WHILE
END FOR
```

Table 4.6 CGCA: character-grams extracted from the word *language*

2-grams	3-grams	4-grams	5-grams	6-grams	7-grams
la	lan	lang	langu	langua	languag
an	ang	angu	angua	anguag	anguage
ng	ngu	ngua	nguag	nguage	
gu	gua	guag	guage		
ua	uag	uage			
ag	age				
ge					

pseudoword. Code Example 4.1 demonstrates extracting character-grams from an origin wordlist, while Table 4.6 shows the character-grams extracted from the word *language*.

4.4.3 Generating pseudowords

The next step is to chain character-grams back together to form pseudowords. First the CGCA selects a character-gram from a set of all those that have an initial count greater than zero (this is called the *starting character-gram*). The selection is determined by frequency; the more frequently occurring character-grams are, the more likely they will be selected. Next it constructs a list of all character-grams that have a total count greater than their initial count, and whose first n–1 characters match the last n–1 characters of the starting character-gram. A character-gram is selected from this list, again based on its frequency, and the CGCA appends its final character to the end of the starting character-gram (this is called a *partial pseudoword*). These steps are repeated until the selected character-gram is one with

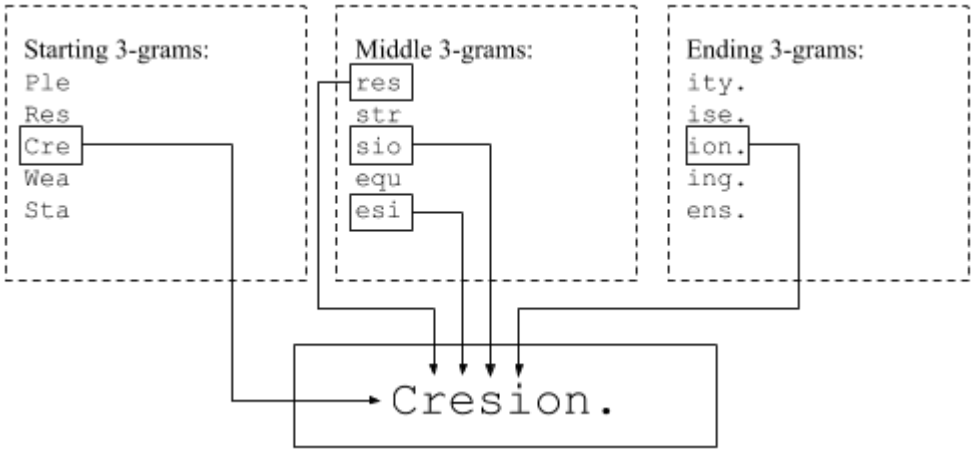


Figure 4.3 CGCA: chaining character-grams together to build pseudowords

a final count greater than zero. If no character-grams match the last *n*–1 characters of the partial pseudoword, it is discarded, and the process begins again. Figure 4.3 demonstrates chaining 3-grams together to form the pseudoword *cresion*.

4.4.4 Validating pseudowords

Finally, potential pseudowords are validated in three ways: (1) they must not appear in the origin wordlist; (2) they must not duplicate a previously generated pseudoword; and (3) they must not be present in a pre-specified lexicon.

The origin wordlist used to generate pseudowords is not necessarily large, for example, good pseudowords can be generated from a list of 1000 words. Thus it is necessary to check that the pseudowords do not exist as real words in a larger lexicon. CGCA uses Wiktionary, which supports over 8,000 languages (Wiktionary, 2017). Before validating a potential pseudoword, an HTTPS request is sent to Wiktionary. The response includes either an error tag or a text tag. The former indicates that the pseudoword is not present in any language, which means that it is valid and can be included in the list of pseudowords. For the latter, an HTML parser is used to determine the language. If the target language is found the pseudoword is invalid and excluded from the list.

4.5 CGCA pseudowords

The CGCA Algorithm was used to generate 800 pseudowords, 100 for each

Table 4.7 CGCA: 2-gram, 3-gram, 5-gram, 8-gram, and r-gram pseudowords

2-gram	3-gram	5-gram	8-gram	r-gram
scon	punit	untalentleman	uncertification	eightist
cens	recollusted	unlabelling	representably	braveller
nes	cree	registract	unstructure	unexception
vois	dward	injusting	undifference	disbehaviour
sunt	witle	orches	intergovernment	ninthood

character-gram size (2-grams to 8-grams), plus 100 that were generated with a randomised gram size (r-grams). Table 4.7 shows five pseudowords generated using 2-grams, 3-grams, 5-grams, 8-grams, and r-grams, while Appendix B.1 shows the full list.

These pseudowords were generated using a subset of the Range Programme Lists that were described in Section 0. The first five lists were used, which contain 1000 word families each. Extracting both headwords and types from each word family resulted in an input wordlist containing 28,256 types, which was then cleaned to create an origin wordlist containing 27,029 types. Cleaning the origin wordlist involved removing partial contractions (i.e. *didn* and *t* from *didn't*), abbreviations (i.e. *thurs*), and slang (i.e. *cuzzies*). This wordlist was then used to extract charater-grams and generate the 800 pseudowords.

4.6 Evaluating pseudowords

This section evaluates the performance of the CGCA algorithm, then introduces two sets of criteria for evaluating pseudowords. The first, *orthographic legality* is designed to evaluate the legal form of pseudowords. The second, *lexical suitability*, evaluates their suitability for lexical tasks.

4.6.1 Evaluating the CGCA algorithm

The CGCA algorithm counts each attempt at creating a pseudoword. An iteration count is updated each time a pseudoword is (1) discarded because it is a real word, (2) discarded because it is a duplicate, or (3) discarded because there is no final character-gram to complete the process. The process fails if the number of attempts exceeds 200 times the target pseudoword count. For example, it will make up to

Table 4.8 CGCA: evaluation with 27,000 origin wordlist

N-gram size	Iterations	Unique pseudowords
2-gram	214	100
3-gram	212	100
4-gram	337	100
5-gram	1121	100
6-gram	3248	100
7-gram	6842	100
8-gram	16511	100
9-gram	20000	57
10-gram	20000	26
r-gram	2418	100

Table 4.9 CGCA: evaluation with 1,500 origin wordlist

N-gram size	Iterations	Unique pseudowords
2-gram	204	100
3-gram	225	100
4-gram	721	100
5-gram	6957	100
6-gram	20,000	17
7-gram	20,000	6
r-gram	1674	100

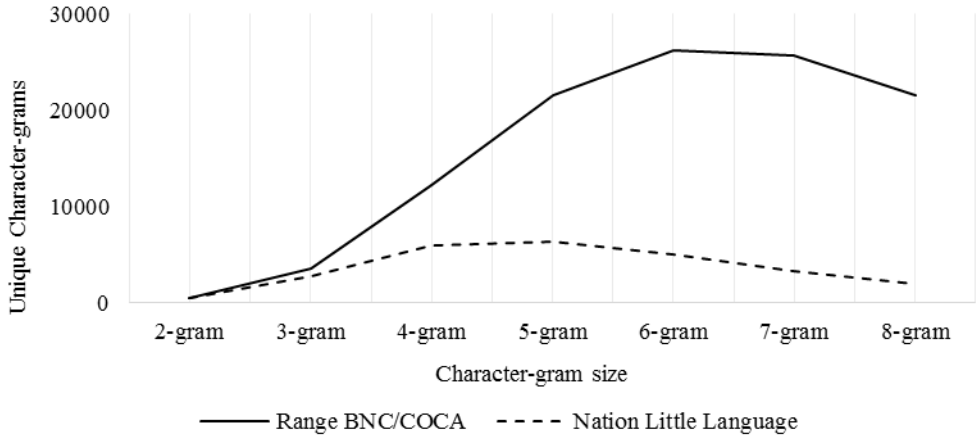


Figure 4.4 CGCA: comparing unique character-grams

20,000 attempts to create 100 pseudowords.

Table 4.8 shows the iteration and pseudoword counts for ten character-gram sizes, using the Range Programme Lists, which contained a total of 27,029 types. Eight character-gram sizes successfully generated 100 pseudowords before hitting the limit. In comparison, Table 4.9 shows the same evaluation, but for pseudowords that were generated using an origin wordlist derived from Nation's Little Language (Nation, 1986) which contains 1,426 types. In this case, the CGCA successfully generated 100 pseudowords for five character-gram sizes. Not surprisingly, the larger the origin wordlist the more pseudowords generated.

Figure 4.4 shows the unique character-gram count for the Range Programme and Little Language origin wordlists. The former peaks at 26,241 unique character-grams, while the latter only reaches 6,400. This difference in the number of unique character-grams indicates why fewer pseudowords can be generated from the smaller wordlist.

4.6.2 Orthographic legality

Pseudowords are described as non-words that have no real meaning in a language but have the correct orthographic or phonological structure, that is, non-words whose structure is legal within the language. In general, pseudoword applications apply constraints to the process of creating pseudowords. They use techniques such as bi-gram frequency, orthographic neighbours, and legal syllable structure to either generate pseudowords or restrict the search criteria. However, none have outlined any formal criteria for evaluating the orthographic structure of pseudowords after they have been generated.

The pseudowords are formed by chaining character-grams together. However, in some cases, chaining character-grams does not result in legal character combinations. For example, although using 2-grams results in legal character sequences of size 2, it can result in larger non-legal character combinations. The CGCA uses n–1 characters of a character-gram to chain the nth character. In this case, only one character is being compared and matched, which can result in longer character combinations that do not appear in the lexicon.

In this section we propose three legality criteria that can be applied to pseudowords after they have been created. We define legal pseudowords as those that conform to these criteria, illustrated in Table 4.10. Strings of one or more

Table 4.10 Orthographic legality criteria

C+	Extract sequences of consecutive consonants from a pseudoword and validate them only if they appear within a type in the origin wordlist, for example, for the pseudoword *conferious*, its consecutive consonants are: *c*, *nf*, *r*, and *s*.
V+	Extract sequences of consecutive vowels from a pseudoword and validate them only if they appear within a type in the origin wordlist, for example, for the pseudoword *conferious*, its consecutive vowels are: *o*, *e*, and *iou*.
CV+C	Extract sequences of consecutive vowels including one leading and one trailing consonant from a pseudoword and validate them only if they appear within a type in the origin wordlist, for example, for the pseudoword *conferious*, its CV+C patterns are: *con*, *fer*, and *rious*.

consecutive consonant, one or more consecutive vowel, and a consonant followed by one or more consecutive vowels followed by a consonant, are extracted from a pseudoword and checked against each of the types in a wordlist, corpus, or lexicon.

The orthographic legality evaluation can be conducted using the origin wordlist. This would allow researchers to conduct the evaluation without requiring any additional language knowledge, resources, or support. However, depending on the size of the origin, this could result in a smaller language to validate against. Nevertheless, if each of the character combinations that appear in the pseudoword also appear in the language, then the pseudoword is considered legal. Conversely, if even one character combination that appears in a pseudoword does not appear in the language, the pseudoword is considered suspect.

Table 4.11 shows the results of applying the orthographic legality criteria to the CGCA pseudowords, using the origin wordlist as the lexicon. Twelve (out of 800) pseudowords incurred errors: ten from 2-grams, one from 3-grams, and one from r-grams. This gives the CGCA pseudowords a 98.5% success rate for legal orthography across character-grams; a percentage that improves when evaluating character-gram sizes individually (with the exception of 2-grams).

Table 4.12 shows the CGCA pseudowords that incurred an error. Four of these pseudowords (*gymma*, *dyntin*, *gympart*, and *rhydrate*) contained a *y*, which

Table 4.11 Orthographic legality evaluation results, for CGCA pseudowords

Pseudowords	C+	V+	CV+C	Sample
2grams	4	0	6	100
3grams	0	0	1	100
4grams	0	0	0	100
5grams	0	0	0	100
6grams	0	0	0	100
7grams	0	0	0	100
8grams	0	0	0	100
r-grams	1	0	0	100

Table 4.12 Orthographic legality error examples, for CGCA pseudowords

yies	eiguit	tbscrap	gymma
yied	jous	faugh	gympart
vois	dyntin	reuniour	rhydrate

although acts as a vowel at times, was not considered a vowel in this evaluation, otherwise they may not have incurred errors. Of the twelve non-legal pseudowords, only one had a clearly illegal orthographic form, *tbscrap*. The character combination *tbs* is not found in the English language but appears here due to the shorthand *tbspoon* appearing in the origin wordlist. The remaining non-legal pseudowords (*yies*, *yied*, *vois*, *eiguit*, *jous*, *faugh*, and *reunior*) include character combinations that do not occur in the origin. Although these combinations may still occur in the English language, they are marked as non-legal since they do not occur in the origin wordlist.

4.6.3 Lexical suitability

The vast majority of CGCA pseudowords were deemed legal by the orthographic legality evaluation. However, it is apparent from the pseudowords shown earlier in Table 4.7 that there is a clear difference in the wordlikeness between character-gram sizes, for example, the 2-gram pseudoword *scon* seems less plausible then the 8-gram pseudoword *uncertification*. Pseudowords are used in a vast selection of ways, including morphology experiments, decoding and pronunciation tests, and vocabulary tests, each of which require pseudowords with varying degrees of

Table 4.13 Lexical suitability criteria

Compound	A pseudowords that is made up of two or more real words within the language, for example *captime* (cap-time).
Polymorphic	A pseudoword that consists of a real root plus one of more affixes, for example, *indetermines* (in-determine-s). Note that a compound can also be polymorphic, for example *captimed* (cap-time-ed).
Near polymorphic	A pseudoword whose root does not exist in the language but includes one or more affixes, for example *alphise* (alph-ise). Note that a pseudoword can be either polymorphic or near polymorphic, but not both. It either has a real root or it doesn't.
One-character dissimilarity	A pseudoword that is *easily* identifiable as one character away from a real word within the language, for example, *overses* (overseas).

wordlikeness. Lexical decision tasks have found that the more dissimilar a pseudoword is to a word, the faster the reaction time (Keuleers & Brysbaert, 2010); vocabulary tests (Meara, 1992) and identification-as-retrieval tasks (Rueckl & Olds, 1993) require pseudowords whose meaning cannot be inferred (i.e. not compound or polymorphic); morphology experiments rely on inferred meaning (plural, possessive, past tense, etc.) (Berko, 1958), and decoding-tasks use pseudowords of varying difficulty (Proença et al., 2017).

This leads to a second set of post-production measurements, measuring the *suitability* of pseudowords for lexical processing and decision tasks. We propose a set of four criteria for evaluating the lexical suitability of pseudowords: *compound*, *polymorphic*, *near polymorphic*, and *one-character dissimilarity*, as outlined in Table 4.13. While the CGCA algorithm is applicable to different languages, the suitability criteria have been developed for the English language.

Each pseudoword receives a score of either 0 or 1 for each criteria, and the results can be used to indicate whether pseudowords may be suited for certain lexical tasks. For example, pseudowords that score 1 in the polymorphic and near polymorphic criteria may be more suited to morphology experiments, while

Table 4.14 Coded affixes, derived from Bauer and Nation (1993)

Inflectional suffixes	Frequent and regular derivational affixes	Frequent orthographically regular affixes
-s -ies -es -ed -d -t -en -ing -er -es	-able -er -ish -less -ly -ness -th -y non- un-	-al -ation -ess -ful -ism -ist -ity -ize -ise -ment -ous in- im-

pseudowords with a 0 score in these criteria may be more suited to receptive vocabulary tests.

Pseudowords were manually coded (independently) by two researchers who gave either 0 or 1 for each of the criteria. For the first three (compound, polymorphic, and near polymorphic), any discrepancies between the two results were discussed and, if possible, resolved. In case of disagreement, a third researcher was recruited to make the final decision. For consistency, the polymorphic and near polymorphic criteria were coded using the list of affixes shown in Table 4.14 (Bauer & Nation, 1993). These were either inflectional suffixes, among the most frequently occurring and regular derivational affixes (abid., pp. 258-259), or frequent and orthographically regular affixes (abid., pp. 259-260). For the fourth criterion (one-character dissimilarity) only pseudowords that had been scored 1 by both researchers were included in the final results. This is because the criterion requires the pseudoword to be *easily* identifiable as one-character away from a real word within the language.

Table 4.15 shows the results of manually coding the criteria. The different character-gram sizes produced different results for each. There were very few occurrences of *compound* pseudowords, though 4-grams generated the most. This suggests that researchers should use 4-grams if they are interested in generating compound pseudowords. The number of pseudowords that showed *one-character dissimilarity* decreased from 2-grams to 4-grams before flattening out, suggesting that the smaller the character-gram size the more *easily identifiable* they are as one character away from real words. The occurrence of *polymorphic* pseudowords increased steadily from 2-grams to 8-grams, suggesting that the larger the character-gram size, the more word-like (real root plus affix) the CGCA pseudowords are. *Near-polymorphic* counts rose, then fell, climbing from 2-grams up to 3-4-5-grams and back down to 8-grams. This final drop corresponds to the climb in polymorphic

Table 4.15 Lexical suitability evaluation results, for CGCA pseudowords

Category	Compound	Polymorphic	Near Poly	Char Diss
2-gram	3	2	22	43
3-gram	5	10	40	39
4-gram	10	22	40	20
5-gram	5	44	40	18
6-gram	1	71	18	16
7-gram	1	80	16	21
8-gram	4	85	9	20
r-gram	5	47	33	14

Table 4.16 Pseudowords generated using external sources

ARC	ELP	WordGen	Wuggy	Meara
grev	drimaced	daney	dre	berrow
bloap	nightkine	biled	woubt	whaley
shrusks	sonehead	ragio	istye	contrivial
zoc	creemason	applk	hu	detailoring
spails	selectove	hoory	roud	eldred

occurrences. These trends suggest that researchers could determine the character-gram size to use based on the type of pseudowords that they require (compound, polymorphic, etc.) for a certain lexical task.

4.7 Comparing pseudowords

We have described two sets of post-production evaluation criteria; one for measuring the orthographic legality of pseudowords, the other for measuring the suitability of pseudowords for particular lexical tasks. Now we use these criteria to compare the CGCA pseudowords with five external sources: two existing pseudoword databases, two software applications, and a pseudoword list derived from a vocabulary test.

4.7.1 External pseudowords

Table 4.16 shows five pseudowords from each external source: the ARC Nonword Database, the English Lexicon Project, WordGen, Wuggy, and pseudowords used

in Meara's EFL vocabulary test. Appendix B.2 shows the full list.

The ARC Nonword Database (ARC) is a database that includes an online interface for pseudoword selection. It provides 22 optional parameters (number of letters, neighbourhood size, etc.) and 23 optional output fields (summed frequency of neighbours, bigram frequency, etc.). We specified values for four variables: the number of pseudowords (100), that the pseudowords should contain only orthographically existing onsets, only orthographically existing bodies, and only legal bigrams. The rest of the parameters were left as their default. The first column in Table 4.16 shows five pseudowords returned by the ARC Nonword Database.

The English Lexicon project (ELP) is also a database that includes an online interface for pseudoword selection. It requires users to enter the minimum and maximum values for seven constraints: pseudoword length, number of orthographic neighbours, summative bi-gram count, average bi-gram count, summative bi-gram count by position, mean reaction time, z-score, standard deviation, number of observations, and mean accuracy. The interface provides users with the average (X) and standard deviation (SD) for each, which we used to calculate a reasonable minimum and maximum parameter (min= X-SD/2, max=X+SD/2). This resulted in 261 pseudowords, from which a random sample of 100 was selected.

WordGen[15] is a software application that runs locally. Once installed, users must enter the length of the pseudowords that will be returned. To avoid losing length variability, WordGen was run three times to obtain 33 four-letter pseudowords, 33 five-letter pseudowords, and 34 six-letter pseudowords (with default options throughout). These three lists were then consolidated to form the full 100 pseudoword dataset.

Wuggy[16] also runs locally. It uses a list of existing words as a basis for generating pseudowords. A random subset of 100 words derived from the first wordlist from the Range Programme List (Nation et al., 2002) were used to generate one pseudoword for each existing word. Most options were left as their default values, but the *matched transition frequency* and *matched sub-syllabic elements* were both set, resulting in 100 pseudowords.

[15] http://www.wouterduyck.be/?page_id=132
[16] http://crr.ugent.be/programs-data/wuggy/downloading-and-installing

Table 4.17 Orthographic legality evaluation results, for external pseudowords

Pseudowords	C+ errors	V+ errors	CV+C errors	Sample
2grams	4	0	5	100
3grams	0	0	2	100
4grams	0	0	1	100
5grams	0	0	0	100
5grams	0	0	0	100
6grams	0	0	0	100
7grams	0	0	0	100
8grams	0	0	0	100
r-grams	1	0	0	100
ARC	3	1	14	100
ELP	4	2	6	100
WordGen	8	5	18	100
Wuggy	3	1	19	100
Meara	0	0	6	100

Finally, Meara's (1992) EFL Vocabulary Test uses pseudowords to measure the reliability of learners self-assessment. Each test includes 40 real words and 20 pseudowords. There is very little documentation describing how Meara created the pseudowords. He states that a proportion include a Greek or Latin stem, but gives no further details of how they were generated. Therefore, a random sample of pseudowords was collected from the tests themselves, 20 from each of the five levels, making up 100 pseudowords in total.

4.7.2 Comparing legality

We have used the orthographic legality criteria to compare pseudowords from different sources. Table 4.17 shows the results. Since majority of the external pseudowords were generated using the CELEX database, but the CGCA pseudowords were not, it was decided to create a new lexicon for this evaluation. It comprised four corpora: the Corpus of Historical American English (COHA) (Davies, 2002), the Wikipedia Corpus (Davies, 2015), News on the Web (NOW) (Davies, 2013b), and Global Web-Based English (GloWbE) (Davies, 2013a). All word types (types) were extracted from the corpora and only words that were

Table 4.18 Orthographic legality error examples, for external pseudowords

ARC	ELP	WordGen	Wuggy	Meara
luilt	opioles	shwrk	saturcip	moffat
sprymphs	heartek	brft	deyityte	hoult
thwibbed	thlorides	grrpe	imeyits	haque
gwurs	lambslin	toint	ymn	balfour
gwibed	neafiest	mynger	polyll	menstruable

validated as real words by Wiktionary were kept. The resulting lexicon contained 72,783 types.

Each set of 100 pseudowords was compared against the COHA-Wikipedia-NOW-GloWbe lexicon, and any character combinations (C+, V+, CV+C) that appeared in a pseudoword but not in the lexicon were noted. For the CGCA pseudowords, when using an external lexicon as opposed to the origin wordlist, the error counts changed very slightly (Table 4.11 versus Table 4.17), with an overall success rate of 98.4% across all character-gram sizes. Comparatively, WordGen had a success rate of 74%, Wuggy 79%, the ARC Nonword Database 84%, the English Lexicon Project 90%, and Meara 94%. This shows that even with an unrelated lexicon the CGCA pseudowords had a higher success rate (across all character-gram sizes) than any of the external pseudowords. However, comparing each character-gram size with the externally generated pseudowords showed that the English Lexicon Project had a similar success rate to CGCA 2-grams (90% and 91% respectively). Meara was the only external set to have a higher success rate than one of the CGCA character-gram sizes (96% for Meara and 91% for 2-grams). The CGCA 5-grams to 8-grams incurred no errors whatsoever, showing that all character combinations were legal according to the COHA-Wikipedia-NOW-GloWbe lexicon. This is not surprising, given that sequences of five to eight consecutive characters were used to make up these pseudowords.

Finally, Table 4.18 shows five pseudowords that incurred errors from each of the external generators. The ARC Nonword Database, WordGen, and Wuggy all incurred errors for pseudowords that did not contain a vowel (*sprymphs*, *shwrk*, *brft*, and *ymn*), although the *y* in *sprymphs* (ARC) could be considered one. This suggests that using constraints to generate or select pseudowords may not be enough for ensuring orthographic legality, and that there is a real need for legal evaluations.

Table 4.19 Lexical suitability evaluation results, for external pseudowords

Category	Compound	Polymorphic	Near Polymorphic	Char Dissimilarity
2-gram	3	2	22	43
3-gram	5	10	40	39
4-gram	10	22	40	20
5-gram	5	44	40	18
6-gram	1	71	18	16
7-gram	1	80	16	21
8-gram	4	85	9	20
r-gram	5	47	33	14
ARC	1	1	58	34
ELP	6	3	58	43
WordGen	1	8	36	48
Wuggy	3	7	37	47
Meara	9	10	26	14

4.7.3 Comparing suitability

The lexical suitability criteria can be used to compare pseudowords from different sources. Table 4.19 shows the result. The CGCA 4-grams and Meara's EFL pseudowords generated the highest number of *compound* pseudowords, suggesting that they should be preferred over other methods. The CGCA pseudowords (except 2-grams) had a higher *polymorphic* count than the externally generated ones, suggesting that the CGCA pseudowords are more word-like, in terms of a real root plus affix. The ARC Non-word Database and the English Lexicon Project had the highest *near-polymorphic* counts, while the highest *one-character dissimilarity* counts came from WordGen and Wuggy. These results suggest that some generators may be more suited to certain lexical tasks than others.

4.8 The DMwW pseudowords

Assuming researchers have access to domain-specific wordlists, the CGCA algorithm can be used to generate domain-specific pseudowords. To illustrate this, we have used the *DMwW wordlist* (described in Section 3.7) to generate the *DMwW*

Table 4.20 CGCA: the DMwW pseudowords

Pseudowords	Words
misclassifier	classification
logistinct	logistics
perceptide	perceptron
apriorate	apriori
technicate	technology

pseudowords. Table 4.20 shows a selection of them, while Appendix B.3 contains the full list. The CGCA algorithm was used to generate 100 pseudowords each for 2-grams, 3-grams, 4-grams, and r-grams. As shown in the table, there is a clear resemblance between the DMwW pseudowords and the words present in the DMwW wordlist.

4.9 Applications of CGCA

There are three main applications for this work: generating pseudowords for other languages, generating domain-specific pseudowords, and generating pseudowords that resemble those from other sources.

First, the CGCA Algorithm generates pseudowords using character-grams extracted from the origin, which results in pseudowords whose orthographic form reflects that of the origin wordlist. This allows CGCA to generate pseudowords for both English, and other languages, without needing any more knowledge of the language than a wordlist. Table 4.21 shows a sample of pseudowords generated using wordlists from four different languages: German, Spanish, Italian, and English. The first three wordlists were derived from subtitles from movies and television series (Buchmeier, 2008a, 2008b, 2009), and the last was derived from the Range Programme Lists (Nation et al., 2002).

Second, if a domain-specific wordlist were used, CGCA would generate domain-specific pseudowords. This was demonstrated in the previous section by generating the DMwW pseudowords. However, Table 4.22 also shows pseudowords generated from wordlists in different domains: an academic domain and grade school vocabulary. The academic wordlist was derived from the New Academic Word List (Coxhead, 2000), and the grade school wordlist from the Basic

Table 4.21 CGCA: German, Spanish, Italian, and English pseudowords

German	Spanish	Italian	English
bisscheint	mirande	abbastardo	acknowier
kinden	puestra	dicevuto	reorganic
viellen	suficio	dentre	sweaten
wassen	oporta	momente	clinist
alleich	histos	dimente	inflatting

Table 4.22 CGGCA: Academic and grade school-based pseudowords

Academic	Grade school
unconverse	brough
enormat	brothes
corresponse	withough
illustract	mountries
emergins	grandmothes

Vocabulary Spelling List (Graham, Harris, & Loynachan, 1993). As illustrated in Table 4.22, there is a clear difference in form and complexity between pseudowords generated from each domain.

Finally, although the lexical suitability criteria are not language independent, they can be applied to any English pseudowords, regardless of how they were created. This can help determine an appropriate character-gram size to use to generate pseudowords that resemble those from other systems. For example, to recreate pseudowords like those created by Wuggy, one could analyse the lexical suitability scores to determine the best character-gram length. The following chapter develops this idea and replicates Meara's pseudowords for use in an automated version of his EFL Vocabulary Test.

Chapter 5

Automating vocabulary tests

Automate /ˈɔtoʊˌmeɪt/

(verb.) (1) to make automatic (2) done by machine (3) to replace or enhance human labour with machines.

(Wiktionary, 2019)

Extensive research has been conducted in relation to quantitatively measuring receptive vocabulary acquisition and vocabulary testing, and numerous tests have been built based on it (Meara, 1992; Nation & Beglar, 2007; Read, 1993). However, these tests are not usually directed toward a particular subject or domain, but are tailored toward general-purpose language. If an applied linguist or language teacher wanted to measure learners' specialised vocabulary, they would need to find an existing vocabulary test that matched their specifications, or create one, but creating custom tests is difficult and time-consuming, and can result in tests that are unreliable.

Section 2.3.2 investigated five well-founded receptive vocabulary tests: The Vocabulary Size Test, the Word Associate Test, and the EFL Vocabulary Test, the X_Lex Test, and the V_YesNo Test. This chapter describes the process of

automating one of them. First, a set of criteria for selecting the test to automate (the EFL Vocabulary Test) are defined, then the *AEFL algorithm*, a system that builds receptive vocabulary tests automatically, is described and evaluated. AEFL allows linguistic researchers and language teachers to use domain-specific wordlists to produce domain-specific vocabulary tests.

The second half of the chapter evaluates the way in which the EFL Vocabulary Test is scored. Researchers have debated the original scoring method and have suggested several alternatives. I will determine which method best evaluates the EFL Vocabulary Test for these learners, and compare the results with those produced by other researchers (Huibregtse, Admiraal, & Meara, 2002; Mochida & Harrington, 2006).

5.1 Selecting a test format

None of the vocabulary tests investigated in Section 2.3.2 are directed toward a particular subject or domain, but rather they are general-purpose vocabulary tests that measure frequency bands of general-purpose words. This thesis aims to automate one of these tests in a way that allows a domain-specific wordlist to be applied, thereby creating a domain-specific vocabulary test. This vocabulary test could then be used to create a rough estimation of domain-specific vocabulary. The resulting test must be capable of creating a rough estimation of domain-specific vocabulary, which, when taken periodically, can provide some indication of progress within domain-specific vocabulary. In determining which type of vocabulary test is best suited to automation (matching or checklist), the following criteria have been defined.

1. The existing test must be automatable. That is, creating test content should not require the experienced knowledge of an applied linguist or language teacher – it cannot require any in-depth knowledge of the language (this is needed for the automated process).
2. The test must provide at least a rough estimate of learner vocabulary. The test does not need to provide an in-depth measure of vocabulary. The aim is to automate a test that can provide some indication of progress within domain-specific vocabulary. It does not need to give an accurate account of all the domain-specific vocabulary that has been acquired.

3. Full and detailed instructions must be readily available. In order to automate an existing test, the details of how the original test was created and/or structured must be clearly defined (such as the wordlists used to create it).
4. A full copy of the original test must readily available. This will be used for comparison with the automated version.

First, only one of the two test types meets the first criteria. As described in Section 2.3.2, *matching* tests map target words with short definitions, related words, or phrases (Read, 2000). Automating this type of test would require a list of target words and the definitions/words/phrases that map to them, which, if done correctly, requires the expertise of a linguist or language teacher. Both of the two example matching tests, the Vocabulary Size Test and the Word Associate Test, illustrate this, as they both presuppose some deeper understanding of the vocabulary: knowledge of vocabulary use within a sentence in the former case, and knowledge of synonyms, collocates, and semantic relations in the latter. In contrast, *checklist* tests require nothing more than a wordlist and related pseudowords. This allows us to eliminate *matching* tests and focus on *checklist* tests for the remaining criteria.

The next criteria requires the test to provide a rough estimate of vocabulary. In these terms, either a *matching* or *checklist* tests would suffice. Matching tests inarguably measure more in-depth knowledge than checklists. The Vocabulary Size Test measures word knowledge by matching target words with their definitions, and the Word Associate Test measures word knowledge using association. However, they did not meet the first criteria. In contrast, checklist tests simply rely on learners indicating whether or not they know a word, but, the criteria only requires a rough estimate of vocabulary, and when discussing the EFL test (a checklist test), Meara (2010, p. 4) stated that "teachers can use the tests to provide a rough lexical profile of individual students, or to monitor the progress of students whose lexical profile has already been established."

The last two criteria lean more toward selecting a specific test, rather than the test type. They are both centred on the details that are available for a particular test version. Is the creation process fully documented, including wordlists and ratios? Is the full version of the test readily available for use? Given that the first criteria eliminated matching-type tests, the last two criteria will focus on the three checklist tests: the EFL Vocabulary Test, the X_LEX Test, and the V_YesNo Test. First, as mentioned in Section 2.3.2, X_Lex is no longer supported or available. The

only documentation I was able to find on this test was a summary by the Centre for Applied Linguistics (Wikipedia, 2019). The EFL and V_YesNo tests are both well documented. The newest version of the EFL test is described in full in Meara (2010), including both of the wordlists used, the full structure of the test, a detailed account of the scoring, and 20 versions of each of the five levels. Likewise, Meara and Miralpeix (2016) outline the V_YesNo test. They describe the structure of the test, include instructions on running the test online, state that the test uses an equal ratio of words and pseudowords, and adequately describe the method of scoring. However, the online version of the test appears to be the only one readily available. It is ultimately for this reason that I have decided to select the EFL Vocabulary Test for automation, although it should be noted that the principles applied throughout the rest of this chapter could have been applied to any checklist based test, so long as full and detailed instructions and a full copy of the original test were readily available.

5.2 The EFL vocabulary test

The EFL Vocabulary Test has five levels, with 40 real words and 20 pseudowords in each. The words in each level are derived from two wordlists and represent five frequency bands (described in Section 2.2.1). Meara (2010) created 100 versions of the EFL Vocabulary Test, 20 for each level.

5.2.1 Wordlists

The real words included in each level were derived from two wordlists: the English Language Institute wordlist (ELI) (Nation, 1986), and the Cambridge English Lexicon (CEL) (Hindmarsh, 1986). Level 1 words came from the first 1000 words from the English Language Institute wordlists, which are commonly referred to as the Little Language. Words were included in the Little Language based on eight principles: language needs, frequency, range, economy, regularity, defining power, classroom and teaching needs, and loan words. They can be used for both receptive and productive vocabulary, as a reference or goal list, as an aid to writing, or to construct exercises and word definitions (Nation, 1986). See Table 5.1.

Level 2 words came from the second 1000 words from the English Language Institute wordlists. They were designed to extend the Little Language and

Table 5.1 First level words, derived from the ELI's first 1000 words

about	add, added	again
accept, accepted	address	age
accident	advertisement	agree, agreed
act, acted	advise, advised	air

Table 5.2 Second level words, derived from the ELI's second 1000 words

able	abstract	accurate
abroad	acceptance	accuse
absent	access	achieve
absolutely	account	acquire

Table 5.3 Third level words, derived from CEL Grade 3

abroad	actress	admire
account	actual	admit
ache	actually	advance
actor	add	advantage

Table 5.4 Fourth level words, derived from CEL Grade 4

absence	accurate	active
absent	accuse	activity
absolute	achieve	ad
accent	acid	addition

Table 5.5 Fifth level words, derived from CEL Grade 5

ability	accord	actively
absolutely	according	additional
academic	accountant	adequate
accelerate	accustom	adjective

were compiled with the aid of the General Service List (West, 1953). Words were included only if they had high frequency in a range of material, and were based on six principles: range, meaning, related meaning, inclusion of basic scientific concepts, contextual requirements, and phrases (Nation, 1986). Unlike the first 1000 words, the second 1000 were intended for use with receptive vocabulary only. Table 5.2 shows a small sample.

Level 3, Level 4, and Level 5 words came from the Cambridge English Lexicon (Hindmarsh, 1986), which was created for inclusion in the University of Cambridge's First Certificate of English examination. The Cambridge English Lexicon is divided into five grades, from post-beginner or beginner (Grade 1) to intermediate – at the level of the Cambridge English First exam (Grade 5). The EFL Vocabulary Test uses Grade 3 words for Level 3, Grade 4 for Level 4, and Grade 5 for Level 5. Tables 5.3, 5.4, and 5.5 show small samples.

5.2.2 Real word selection

Each level of the EFL contains 40 real words and 20 pseudowords. Real words were selected from the wordlists. However, it is unclear how the words were chosen, or even whether it was random or carefully thought out. The same word is sometimes repeated twice in the same test version, for example, *certain* is repeated twice in the eighth version of Level 1 (Meara, 2010, p. 25), *patient* in the thirteenth version of Level 2 (Meara, 2010, p. 52), and *pretend* in the seventeenth version of Level 3 (Meara, 2010, p. 78).

5.2.3 Pseudoword selection

Like real word selection, pseudowords selection is also unclear. Meara (1992) states that a proportion included a Greek or Latin stem, but gave no further details of how they were generated. However, it is generally accepted that they should conform to the phonological and morphological rules of the language (Beeckmans, Eyckmans, Janssens, Dufranne, & Van de Velde, 2001; Huibregtse et al., 2002; Mochida & Harrington, 2006).

5.2.4 Convening the test

Learners are asked to read through the words carefully, indicating for each one whether they know the word or not by putting "Y" or "N" in a box beside it. What

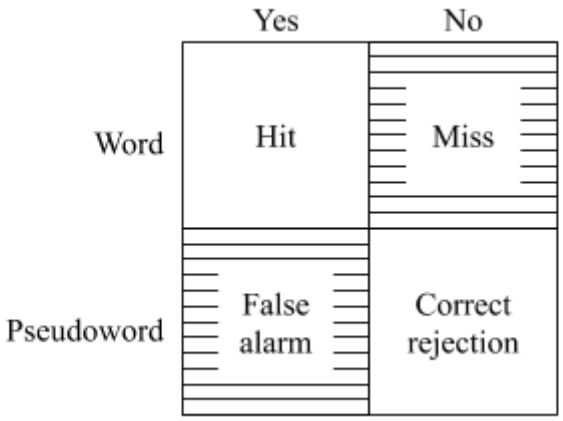


Figure 5.1 Response matrix for a Yes/No vocabulary test

it means to *know* a word is left up to the learner. Figure 5.1 shows the four response types: hits, misses, false alarms, and correct rejections. Learners receive a score for the test, based on the number of real words and pseudowords they claim to know (hits and false alarms).

False alarms are included in the scoring calculation to offset a learner's predisposition to answer liberally or conservatively. For example, if a learner incorrectly claimed to know several pseudowords, it could be assumed that they also incorrectly claimed to know several real words. Conversely, if a learner did not claim to know any pseudowords, it could be assumed that they were more conservative in their self-evaluation, and that it was less likely they claimed to know real words that they did not know. In the original EFL Vocabulary Test, learners' hit and false alarm counts are used to retrieve a percentage vocabulary score for each test level, using Meara's scoring matrix (discussed further in Section 5.8.1).

5.3 The AEFL algorithm

The AEFL (Automatic EFL) algorithm, developed for this thesis, automates the creation of an EFL Vocabulary Test. It selects 40 real words and generates 20 pseudowords, from an input corpus or wordlist, to create an automated version of the EFL Vocabulary Test. The resulting test reflects the wordlist used to create it, thereby facilitating the creation of custom vocabulary tests. The algorithm takes a wordlist or corpus as input and implements three tasks: building an origin wordlist,

selecting real words and generating pseudowords. The final test is output as a text file.

5.3.1 Building an origin wordlist

The AEFL algorithm builds an origin wordlist from an existing wordlist or corpus. Like the Character-gram Chaining Algorithm (CGCA) described in Section 4.4.1, the AEFL accepts a wordlist or corpus as input and extracts word types, building an origin list of all unique real words.

5.3.2 Real word selection

Once an origin wordlist has been created, the AEFL selects 40 real words. Selection is done without replacement, meaning that each word can only be selected once. However, using an origin wordlist that includes both a headword and its derived and inflected forms can result in multiple words being selected from the same word family. Words should be considered carefully before being included in the origin wordlist.

5.3.3 Generating pseudowords

The AEFL algorithm uses the Character-gram Chaining Algorithm (CGCA) described in Section 4.4 to generate pseudowords that reflect the domain and language represented in an origin wordlist. The CGCA creates pseudowords by chaining character-grams together, based on their probability of appearing together in real words.

Section 4.5 demonstrated using character-grams as small as 2-grams and as large as 8-grams. However, there was a noticeable difference in the orthographic and morphologic form of the pseudowords produced. Pseudowords generated by the AEFL algorithm need to have similar orthographic and morphologic forms to those used in Meara's original tests. Section 4.6.3 established four suitability criteria for evaluating the wordlikeness of pseudowords: compound, polymorphic, near polymorphic, and one-character dissimilarity. As suggested in Section 4.9, results from the suitability criteria can be used to compare Meara's pseudowords with the pseudowords generated using each character-gram size.

A series of two-tailed z-tests were conducted, determining which character-gram size generated pseudowords that best match Meara's. Two-tailed z-

Table 5.6 AEFL: counts and test statistics for samples of 100 pseudowords

Category	Compound	Polymorph	Near Poly	One-char
2-gram	3 (1.79)	2 (2.38)	22 (0.66)	43 (-4.54)
3-gram	5 (1.11)	10 (0)	40 (-2.11)	39 (-4.01)
4-gram	10 (-0.24)	22 (-2.31)	40 (-2.11)	20 (-1.13)
5-gram	5 (1.11)	44 (-5.41)	40 (-2.11)	18 (-0.77)
6-gram	1 (2.6)	71 (-8.79)	18 (1.37)	16 (-0.4)
7-gram	1 (2.6)	80 (-9.95)	16 (1.74)	21 (-1.3)
8-gram	4 (1.43)	85 (-10.6)	9 (3.16)	20 (-1.13)
r-gram	5 (1.11)	47 (-5.8)	33 (-1.09)	14 (0)
Meara	9 (0)	10 (0)	26 (0)	14 (0)

tests were chosen over other statistical methods for three reasons: (1) the variables are categorical (for example, compound / not compound, polymorphic / not polymorphic), (2) we are interested in the proportion of individual pseudowords with a certain characteristic (for example the proportion of compound pseudowords), and (3) the samples are independent (Rumsey, 2019). The null hypothesis, H_0, is that the two population proportions are equal, where p_1 is the proportion of CGCA pseudowords with a certain suitability criterion, and p_2 is the proportion of Meara's pseudowords with the same.

$$H_0: \ p_1 - p_2 = 0$$

First, the results from the suitability evaluation can be used to compute the test statistics for each of the two-tailed tests. The test statistic is as follows.

$$test\ statistic = \frac{(p_1 - p_2)}{\sqrt{p(1-p)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

where p_1 is the proportion of CGCA pseudowords with a certain suitability criterion, and p_2 is the proportion of Meara's pseudowords with the same. n_1 and n_2 = 100 respectively, for the total number of instances in each sample, and p is the total number of instances that have the attribute of interest from both samples, divided by the total number of instances for both samples.

Table 5.6 shows the results from the suitability evaluation, along with the test statistic (in brackets). The test-statistic represents the degree of agreement

Table 5.7 AEFL: p-value results for samples of 100 pseudowords

Category	Compound	Polymorph	Near Poly	One-char
2-gram	~~0.0367~~	~~0.0087~~	0.2546	~~0~~
3-gram	0.1335	0.5	~~0.0174~~	~~0~~
4-gram	0.4052	~~0.0104~~	~~0.0174~~	0.1292
5-gram	0.1335	~~0~~	~~0.0174~~	0.2206
6-gram	~~0.0047~~	~~0~~	0.0853	0.3446
7-gram	~~0.0047~~	~~0~~	~~0.0409~~	0.0968
8-gram	0.0764	~~0~~	~~0.0008~~	0.1292
r-gram	0.1335	~~0~~	0.1379	0.5

between the data and the correctness of the null hypothesis (Bijma, Jonker, & van der Vaart, 2017). The magnitude of the test statistic determines whether the null hypothesis will be rejected, while its sign represents the direction of the difference between two population proportions. For example, when Meara has a higher population proportion than the CGCA, the sign is positive (i.e. Table 5.6, column 1, row 1), but when Meara has a lower population proportion, it is negative (i.e. Table 5.6, column 1, row 3). A test statistic of zero is obtained when two population proportions are equal, shown in the final row of Table 5.6, where Meara's pseudowords are being compared against themselves.

Finally, Table 5.7 demonstrates the corresponding p-values. The p-value is the maximum probability of a more extreme test statistic occurring if the experiment was recreated identically (Bijma et al., 2017). When the p-value is less than or equal to alpha (α=0.05), the probability is small enough for us to reject the null hypothesis. Character-gram sizes that have rejected the null hypothesis are eliminated from the pool of character-grams that may be used to generate AEFL pseudowords. However, as shown in Table 5.7, each character-gram size has been rejected (crossed out) for at least one suitability criteria.

Automating Meara's EFL Vocabulary Test requires pseudowords with a similar form to the original tests, but all character-gram sizes showed a statistically significant difference in at least one suitability criteria. Another measure needs to be added to the CGCA generation to create pseudowords that better match Meara's.

The CGCA pseudowords generated using smaller character-gram sizes have a much higher one-character dissimilarity count than Meara's pseudowords

Table 5.8 AEFL: counts and test statistics, after applying the WSM

Category	Compound	Polymorph	Near Poly	One-char	Sample
2-gram*	3 (0.67)	2 (1.31)	8 (1.44)	8 (-0.28)	51
3-gram*	4 (0.37)	5 (0.18)	19(-1.12)	11(-0.97)	55
4-gram*	7 (-1.27)	3 (0.58)	19(-2.15)	2 (1.63)	43
5-gram*	1 (0.95)	4 (-0.64)	19 (-4.1)	1 (1.52)	28
6-gram*	0 (1.08)	6 (-3.74)	4 (-0.54)	0 (1.39)	12
7-gram*	0 (0.99)	3 (-1.87)	5 (-1.61)	0 (1.27)	10
8-gram*	2 (-2.61)	2 (-2.46)	1 (0.04)	0 (0.8)	4
r-gram*	2 (0.53)	10(-2.83)	17(-2.72)	2 (1.22)	33

(43 for 2-grams vs 14 for Meara), and CGCA pseudowords generated using higher character-gram sizes have a much higher polymorphic count (85 for 8-grams vs 10 for Meara). By creating an additional variable, referred to as the *Word similarity metric*, we can restrict these criteria and generate pseudowords more similar to those used in Meara's original test.

The Word similarity metric (WSM) was developed for this work. It is based on a simple spelling corrector, developed by Norvig (2016), but rather than suggesting corrections it uses Python's SequenceMatcher (Python Software Foundation, 2019) to compute a pseudowords orthographic similarity to existing words. Pseudowords with no similarities to real words score 0, those identical to real words score 1, and those with varying similarities score something in between. The word similarity metric has been applied to the CGCA pseudowords and words with a WSM score exceeding 0.85 have been removed. This reduces the number of pseudowords that are one-character away from real words or that contain a root and an affix (polymorphic and near polymorphic). For example, the pseudoword *exhausive* was one character away from *exhaustive* and had a WSM value of 0.9474; *unconsolidate* contained the prefix *un* and the root *consolidate* and had a WSM value of 0.963. Both have been removed from the data.

Table 5.8 shows counts and test statistics after the WSM was applied, while Table 5.9 shows the corresponding p-values. Both 2-grams and 3-grams (underlined) showed no statistically significant difference. Consequently, the AEFL algorithm will use 3-grams and a WSM of 0.85 or less to generate pseudowords.

Table 5.9 p-values after applying the word similarity metric

Category	Compound	Polymorph	Near Poly	One-char
2-gram*	0.2514	0.0951	0.0749	0.3897
3-gram*	0.3557	0.4286	0.1314	0.166
4-gram*	0.102	0.281	~~0.0158~~	0.0516
5-gram*	0.1711	0.2611	~~0~~	0.0643
6-gram*	0.1401	~~0.0001~~	0.2946	0.0823
7-gram*	0.1611	~~0.0307~~	0.0537	0.102
8-gram*	~~0.0045~~	~~0.007~~	0.484	0.2119
r-gram*	0.2981	~~0.0023~~	~~0.0033~~	0.1112

5.3.4 Outputting the test

Finally, the AEFL algorithm outputs a text file of 40 real words and 20 pseudowords, plus indices for each pseudoword. The text file can be read by a word processor for formatting. Meara's original test had five levels corresponding to five frequency bands. To replicate this, the AEFL algorithm can be run five times using five frequency wordlists.

5.4 Pilot study tests

The AEFL algorithm has been developed to create custom vocabulary tests. However, custom vocabulary tests pose the threat of unreliability. Before the AEFL algorithm can be used to create custom tests, its reliability needs to be tested. If the AEFL algorithm can create a test that is statistically indistinguishable from the original EFL Vocabulary Test, it can, in turn, be used with appropriate wordlists or corpora to create useful domain-specific tests, or tests in any language. A pilot study was conducted with 10 participants, to test the reliability of the AEFL algorithm.

5.4.1 Participants

Ten individuals participated in the study. Each one was a post graduate student from the University of Waikato. Of the ten participants, five were L2 English speakers.

5.4.2 Selecting the EFL test (control)

Meara (2010) provides 100 versions of the EFL Vocabulary Test, 20 for each level.

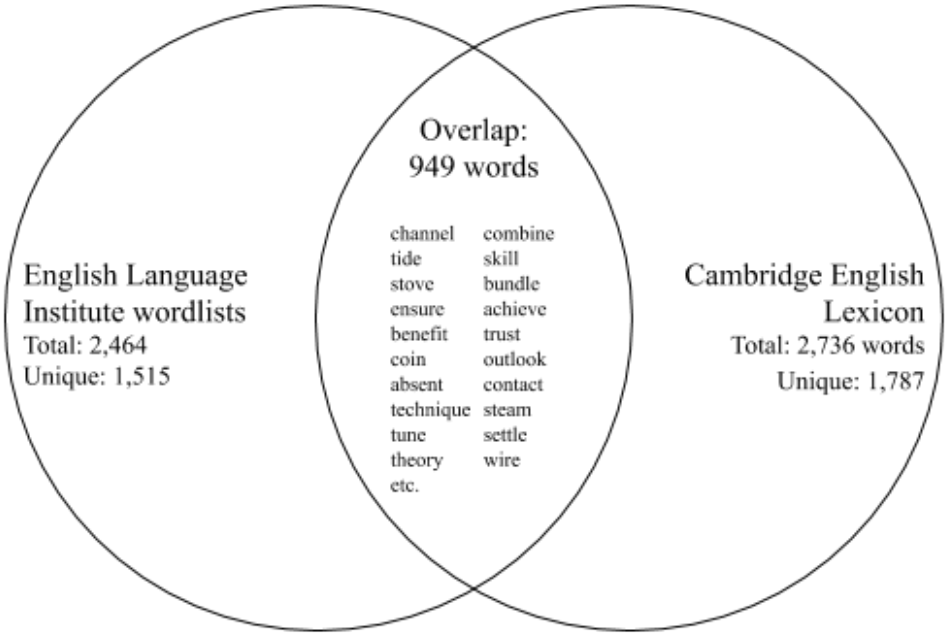


Figure 5.2 Venn diagram depicting overlapping words

One version was selected from each level as the control. However, some include inconsistencies: the pseudoword *maltmass* appears twice in one test; *ryan* is used as a pseudoword in another, when it is a commonly recognized name; the real word *paralogue*[17] is used as a pseudoword; and the proper noun *muslim* is present, but without the proper capitalization. Based on these results, four restrictions have been enforced when selecting the versions of the test to use as the control.

1. There can be no duplicates within a single test.
2. Pseudowords cannot be easily recognisable names.
3. Real words cannot be used as pseudowords.
4. Proper nouns without proper capitalization cannot be used.

One version from each level was selected: level 1 version 12, level 2 version 12, level 3 version 20, level 4 version 20, and level 5 version 8 (Meara, 2010, pp. 29,51,81,103,113). Appendix C.1 shows the tests that were selected.

5.4.3 Generating the AEFL test

Fairly replicating the original EFL test requires the use of the original wordlists: the English Language Institute wordlist (Nation, 1986) and the Cambridge English

[17] A pair of genes that derives from the same ancestral gene (Wiktionary, 2019)

Lexicon (Hindmarsh, 1986). Being unavailable online, they were manually transcribed from hard copy and separated into wordlists corresponding to the five levels described in Section 5.2.1. Meara suggests using tests from each of the five levels to build a frequency profile for learners that corresponds to the first five word frequency bands (Meara, 2010, p. 5).

As shown in Figure 5.2, there is considerable overlap between the English Language Institute wordlist (1st 1000 and 2nd 1000) and the Cambridge English Lexicon (Grades 3, 4, and 5). 949 words appear in both lists, which may affect the results of the original EFL Vocabulary Test. However, since the AEFL algorithm uses the same wordlists as the original test, any affected results should also be reflected in the test generated by the AEFL.

The AEFL algorithm was run five times, initially using the first 1000 words from the English Language Institute wordlist. The four consecutive runs used the corresponding wordlists for Levels 2, 3, 4 and 5. Appendix C.2 shows the tests that were generated.

5.4.4 Methodology

Each participant was given both the control test (EFL) and the AEFL test. The two tests were combined, and the full set was ordered by test level. The order of the control test and AEFL test was alternated between participants. This meant that both Level 1 control and Level 1 AEFL were completed before Level 2 (in alternating order); both Level 2 control and Level 2 AEFL were completed before Level 3 (in alternating order); and so on.

The test was sat in person with a pen and paper test sheet, formatted to match the original EFL Vocabulary Test. Participants were asked to read through the list of words carefully, and for each word, if they knew what it meant, write "Y" (for YES), or put a tick in the box. If they did not know what it meant, or if they weren't sure, write "N" (for NO) or put an *X*. They were also given a hint that some of the words may be imaginary. Participants were not given a time restriction, but were asked to complete all test levels within one sitting. Ethical consent was applied for and approved prior to the start date for the study, as shown in Appendix F.1.

Table 5.10 Pilot study: Hits and p-values for the EFL and AEFL tests

	Level 1		Level 2		Level 3		Level 4		Level 5	
	control,AEFL(p)		control,AEFL(p)		control,AEFL(p)		control,AEFL(p)		control,AEFL(p)	
P1	40, 40	(1.0)	40, 40	(1.0)	40, 40	(1.0)	40, 40	(1.0)	40, 40	(1.0)
P2	39, 38	(0.5552)	32, 36	(0.2112)	30, 36	(0.0768)	28, 28	(1.0)	21, 16	(0.2628)
P3	40, 37	(0.0768)	33, 38	(0.0768)	~~29, 37~~	~~(0.0188)~~	~~36, 27~~	~~(0.014)~~	24, 22	(0.6528)
P4	40, 39	(0.3124)	39, 40	(0.3124)	40, 40	(1.0)	-, -	(-)	39, 38	(0.5552)
P5	39, 40	(0.3124)	36, 37	(0.6892)	37, 40	(0.0768)	36, 34	(0.4964)	27, 23	(0.3576)
P6	40, 40	(1.0)	40, 40	(1.0)	40, 40	(1.0)	38, 40	(0.1528)	40, 40	(1.0)
P7	40, 40	(1.0)	40, 40	(1.0)	40, 40	(1.0)	39, 40	(0.3124)	40, 40	(1.0)
P8	40, 40	(1.0)	40, 40	(1.0)	40, 40	(1.0)	40, 39	(0.3124)	40, 40	(1.0)
P9	40, 40	(1.0)	38, 39	(0.5552)	36, 36	(1.0)	34, 33	(0.7642)	27, 28	(0.8104)
P10	40, 40	(1.0)	40, 40	(1.0)	40, 40	(1.0)	40, 40	(1.0)	40, 40	(1.0)

Table 5.11 Pilot study: False alarms and p-values for the EFL and AEFL tests

	Level 1		Level 2		Level 3		Level 4		Level 5	
	control,AEFL(p)		control,AEFL(p)		control,AEFL(p)		control,AEFL(p)		control,AEFL(p)	
P1	2, 1	(0.5484)	1, 0	(0.3124)	0, 0	(1.0)	3, 1	(0.2938)	4, 1	(0.1528)
P2	1, 1	(1.0)	1, 5	(0.0768)	0, 0	(1.0)	1, 2	(0.5484)	1, 1	(1.0)
P3	1, 1	(1.0)	~~0, 5~~	~~(0.0168)~~	2, 4	(0.3734)	3, 6	(0.2542)	2, 6	(0.114)
P4	1, 0	(0.3124)	1, 0	(0.3124)	1, 0	(0.3124)	-, -	(-)	0, 0	(1.0)
P5	4, 5	(0.704)	2, 3	(0.6312)	0, 1	(0.3124)	5, 2	(0.2112)	1, 2	(0.5484)
P6	0, 0	(1.0)	0, 1	(0.3124)	0, 0	(1.0)	0, 1	(0.3124)	1, 0	(0.3124)
P7	0, 0	(1.0)	0, 0	(1.0)	1, 0	(0.3124)	2, 1	(0.5484)	0, 0	(1.0)
P8	~~7, 1~~	~~(0.0178)~~	6, 3	(0.2542)	~~6, 0~~	~~(0.0078)~~	8, 5	(0.3124)	5, 1	(0.0768)
P9	3, 1	(0.2938)	4, 3	(0.6744)	1, 1	(1.0)	0, 0	(1.0)	1, 0	(0.3124)
P10	2, 0	(0.147)	0, 0	(1.0)	2, 0	(0.147)	1, 2	(0.5484)	2, 0	(0.147)

5.4.5 Results

Each test was scored by tallying the number of real words that were indicated as known (hits) and the number of pseudowords that were indicated as known (false alarms). Meara's original test instructions stated that a score of less than 10 hits or more than 10 false alarms resulted in a score that was extremely unreliable (Meara, 1992). For this reason, one participant (P0) was excluded due to an exceedingly high number of false alarms. Another participant, P4, did not attempt Level 4 of the control test. However, they did complete the remainder of the test. For this reason, they have been included in the analysis, but their result for the Level 4 AEFL test has been excluded. All other participants completed all tests.

5.4.5.1 Hits

Table 5.10 shows the hit counts for each participant. A series of two-tailed z-tests were conducted, to compare the hit counts of the AEFL test with the hit counts of the control. The null hypothesis, $H_{0,}$ was that the two population proportions were equal. Table 5.10 shows the resulting p-values (in brackets). In this case, rejecting the null hypothesis meant there was a statistically significant difference between the hit counts gained in the control and AEFL tests. One participant (P3) showed a statistically significant difference between their hit counts (crossed out in Table 5.10). The rest (90%) showed no statistically significant difference in any test level. 49 z-tests were conducted in total, 5 for each participant, except for P4 who did not complete the Level 4 test. Of the 49 z-tests, 47 (96%) showed no statistically significant difference.

5.4.5.2 False Alarms

Table 5.11 shows the false alarm counts and corresponding p-values (in brackets) for each participant. Like hit counts, a series of two-tailed z-tests were conducted, to compare the false alarm counts of the AEFL test with the false alarm counts of the control. Again, the null hypothesis, $H_{0,}$ was that the two population proportions were equal. In this case, rejecting the null hypothesis meant there was a statistically significant difference between the false alarm counts gained in the control and AEFL tests. Two participants (P3 and P8) showed a statistically significant difference between their false alarm counts (crossed out in Table 5.11). The rest (80%) showed no statistically significant difference in any test level. Again, 49 z-tests were conducted in total. Of those z-tests, 46 (94%) showed no statistically significant difference.

5.4.6 Discussion

We can draw conclusions about the accuracy of the AEFL algorithm from the hit and false alarm results. First, 96% of tests showed no statistically significant difference for hit counts, showing that the AEFL's real word selection fairly replicates Meara's original real word selection. This is not surprising, since the real words used in both tests originated from the same wordlists. Second, 94% of tests showed no statistically significant difference for false alarm counts, demonstrating that the AEFL's pseudoword generation fairly replicates the orthographic and

morphologic form of those used in Meara's test, and supports the use of 3-grams with a Word Similarity Metric no higher than 0.85. Finally, 90% of tests showed no statistically significant difference for both hit and false alarm counts, showing that the AEFL algorithm's automated vocabulary test is a fair replica of Meara's EFL Vocabulary Test.

5.5 Main study tests

The pilot study, with ten participants, found that the AEFL test was statistically indistinguishable from Meara's original EFL Vocabulary Test for 80% of participants and 90% of tests. This section outlines a larger study, which was conducted online, with 100 participants.

5.5.1 Participants and tests

The study was conducted with 100 participants: 90 L1 English speakers and 10 L2 English speakers. Table 5.12 shows the L1 and L2 languages specified by participants. Tables 5.13 and 5.14 show the words used in the control and AEFL tests. These are the same words that were used in the Level 1 test of the pilot study.

5.5.2 Methodology

The test was administered via an online survey form, which was posted on social media. The form contained real words and pseudowords for Level 1 of both the control and AEFL test. The test was displayed, one word per line, with a checkbox next to each. Words and pseudowords were reordered between each participant.

Participants were asked to state their L1 language, plus any L2 languages that they could speak. They were asked to read through the list of words carefully, and for each word, if they knew what it meant, put a tick in the box. If they did not know what it meant, or if they weren't sure, they were asked not to put a tick in the box. Participants were also given a hint that some of the words could be imaginary. They were not given a time restriction, but were asked to complete the test in one sitting. Ethical consent was applied for and approved prior to the start date for the study, as shown in Appendix F.2.

Table 5.12 Main study: L1 and L2 languages from the larger case study

L1 language				
Afrikaans	Bislama	German (2)	Korean	Tamil
Albania	English (90)	Konkani	Slovak	Vietnamese
L2 languages				
Afrikaans	Filipino	Hindi	Marathi	Spanish
Chinese	French (4)	Japanese (2)	Samoa	Swedish
English (6)	German	Konkani	Sinhalese	Māori (3)

Table 5.13 Main study: real words and pseudowords from the EFL test

Real words
also, bad, bite, bridge, camera, circle, copy, curtain, day, earn, engine, explain, feature, forest, govern, half, hurt, kill, lazy, lie, mad, modern, moon, next, patient, person, prison, private, rice, row, shake, shell, size, speak, street, tax, test, trousers, ugly, warm
Pseudowords
absalom, aspection, attard, berrow, catling, classinate, cymballic, expedalize, hallett, hapgood, harmonical, lowry, mascarate, murtagh, plebocrat, portingale, postherent, retrogradient, rickard, suddery

Table 5.14 Main study: real words and pseudowords from the AEFL test

Real words
back, best, break, cause, cover, dance, date, dreamt, egg, fairly, fiercely, flew, fork, kindly, lie, light, loved, met, noisily, paint, pale, piece, practise, practised, pretend, put, rainy, reach, rose, seriously, skin, solid, space, stairs, study, tap, thank, unsuccessfully, woman, wrong
Pseudowords
brint, busin, corrow, coung, danch, furned, jumperime, lavate, lenger, maineer, posite, regun, sistmas, terriend, tirect, trience, untrol, villow, yellen, yestric

Table 5.15 Main study: percentage of tests with no statistical difference

	Hits	False alarms	Neither
L1	100%	88%	88%
L2	89%	78%	67%
All	99%	87%	86%

Table 5.16 Main study: mean hit and false alarm counts for the EFL and AEFL

	Hits	False alarms
	EFL, AEFL (p)	EFL, AEFL (p)
L1	39.84, 39.76 (0.8966)	1.98, 0.56 (0.3576)
L2	40.00, 38.89 (0.2892)	3.00, 1.33 (0.3954)
Both	39.86, 39.68 (0.7948)	2.07, 0.63 (0.3628)

5.5.3 Results

Hit and false alarm counts were calculated for each of the 100 participants. However, 3 were excluded from the analysis due to an exceedingly high number of false alarms. This resulted in a total of 97 participants, 88 L1 learners and 9 L2 learners. A series of two-tailed z-tests were conducted to compare the hit and false alarm counts of the AEFL and control test. The null hypothesis, $H_{0,}$ was that the two population proportions were equal.

Table 5.15 shows the percentage of tests with no statistically significant difference for hit counts, no statistically significant difference for false alarm counts, and no statistically significant difference in either hit or false alarm counts. The table also splits the results into L1 and L2 participants. When evaluating all participants (both L1 and L2), 99% showed no statistically significant difference between hit counts for the control and AEFL test; 87% showed no difference for false alarm counts; and 86% showed no difference for both hit and false alarms counts.

There is a noticeable difference between the percentages of tests for L1 and L2 participants, particularly when evaluating both hit and false alarm counts (88% versus 67%). However, L2 participants only contributed 9% of the data (9 out of 97 participants). I would recommend further studies be conducted to evaluate the difference between L1 and L2 learners.

Table 5.16 shows the mean hit and false alarm counts for the control and AEFL tests, plus the p-values associated with them (in brackets). As shown in the table, there is no statistically significant difference for mean hit counts or mean false alarm counts, for L1 participants, L2 participants, and all participants combined.

5.5.4 Discussion

For all participants, 99% of tests, in the main study, showed no statistically significant difference between hit counts. This supports the AEFL's real word selection, replicating Meara's original real word selection. Again, this is not surprising, since the real words used in both tests originated from the same wordlists. Second, 87% of tests showed no statistically significant difference between false alarm counts. Although this is lower than the pilot study results (94%), it demonstrates that the AEFL's pseudoword generation replicates the orthographic and morphologic form of those used in Meara's test. Finally, 86% of tests showed no statistically significant difference between both hit and false alarm counts. Again, although lower than the pilot (90%), it shows that the AEFL algorithm's automated vocabulary test is a fair replica of Meara's EFL Vocabulary Test.

5.6 Implications of AEFL

The first half of this chapter investigated the process of re-creating an existing vocabulary test entirely programmatically. the hypothesis was that vocabulary tests generated using the AEFL algorithm – using character-grams of length three (3-grams), and excluding any pseudowords that have a Word Similarity Metric greater than 0.85 – would produce results similar to those produced by the original EFL Vocabulary Test, so long as the same wordlists were used to create both. This hypothesis can be broken into two sections, (1) were the correct pseudowords used in the AEFL test, and (2) is the AEFL test a good substitute for Meara's original EFL Vocabulary Test?

First, the false alarm counts for both the pilot and main vocabulary studies validate the choice of AEFL pseudowords. The suitability criteria (compound, polymorphic, near polymorphic, and one-character dissimilarity) were used to select the character-gram size and determine the Word Similarity Metric for the pseudowords that were generated. The results from the false alarm scores

Table 5.17 Real words and pseudowords from the DMwW test

Real words
apache, assign, automatic, batch, category, characteristic, core, correlate, cycle, default, deteriorate, distribute, effective, explicit, gradient, identify, indicate, infrared, interface, kernel, majority, manual, mode, multifilter, outperform, preprocess, probably, procedure, reveal, ripper, sensitive, sophisticated, specific, structure, technical, tedious, template, trivial, typical, visualizer
Pseudowords
accur, algory, annear, bayer, bioinform, circumstall, finate, frustruct, identiment, libline, majorithm, normat, opervise, optide, passifier, peptron, predical, selete, specialogy, stochanism

demonstrate a strong similarity between the AEFL pseudowords and those used in the original EFL tests. This, in turn, supports the use of selection criteria to determine the type of pseudowords required. If the pseudowords had been less well suited to the test, the difference in false alarm scores between the control test and the AEFL test could have been much more substantial. Although not within the scope of this research, another series of tests could be conducted with different pseudowords to test this theory.

Secondly, the results of the pilot and main study found that, for the majority of tests and participants, Meara's original EFL test and the generated AEFL test showed no distinguishable difference, both in terms of hits and false alarms. Given these results, the AEFL tests could be used in place of Meara's original EFL Vocabulary Test, not just with the original wordlists, but with language or domain-specific wordlists, resulting in automatically generated domain-specific vocabulary tests.

5.7 The DMwW vocabulary test

Assuming researchers have access to domain-specific wordlists, the AEFL algorithm can be used to generate domain-specific versions of the EFL Vocabulary Test. To illustrate this, I have used the *DMwW wordlist* (described in Section 3.7) to generate the *DMwW Vocabulary Test*. Table 5.17 illustrates the real words and pseudowords present in the test, while Appendix C.3 contains the formatted

Table 5.18 An excerpt from Meara's original scoring matrix

FA / H	0	1	2	3	4	5
40	100	95	90	85	80	75
39	98	92	87	82	76	71
38	95	90	84	78	73	67
37	93	87	81	75	69	63
36	90	84	78	72	65	59
35	88	81	75	68	62	55

DMwW Vocabulary Test. The CGCA algorithm was used to generate the pseudowords, using 3-grams and a WSM no greater than 0.85.

5.8 Scoring the EFL

There has been some debate around Meara's original method for scoring the EFL Vocabulary Test. In his 2010 revision of the test, he acknowledged that there was an unresolved issue with the scoring system. The scoring formula relied on mathematical assumptions that were incorrect, resulting in learners being scored harshly in some situations, particularly when their "yes" responses were low (Meara, 2010). Since the original 1992 EFL Vocabulary Test was published, there have been multiple attempts at correcting the scoring bias (Beeckmans et al., 2001; Huibregtse et al., 2002; Mochida & Harrington, 2006). However, Meara has stated that none were entirely satisfactory. In his 2010 revision, he spoke of creating a Bayesian model that would "settle this issue once and for all" (Meara, 2010, p. 2). However, to my knowledge, there has been no revision since. In this section, we will look at both Meara's original scoring method and those that have been suggested by others.

5.8.1 Meara's Matrix (Δm)

Meara's original scoring method involves using a scoring matrix to convert hit and false alarm counts into a final percentage vocabulary score. Table 5.18 illustrates an excerpt from Meara's original scoring matrix, while Appendix D shows the full table. The figures in the matrix are derived from the equation shown below.

$$\Delta m = \frac{h - f}{1 - f} - \frac{f}{h}$$

where *h* is the observed hit rate and *f* is the observed false alarm rate. The hit and false alarm rates are values between 0 and 1, corresponding to the percentage of real words that were indicated as known, and the percentage of pseudowords indicated as known.

This scoring measure is derived from Signal Detection Theory, which handles sophisticated guessing based on the assumption that learners choose the most plausible alternative, rather than the all-or-nothing approach used in blind guessing (Huibregtse et al., 2002; Nunnally & Bernstein, 1994). It is a transformation of a formula described by Grier (1971) that uses the area under a ROC curve to determine a learner's overall vocabulary score.

5.8.2 The *h–f* method

The simplest way to measure a learner's performance would be to count all correct responses – all real words that were indicated as known (hits) and all pseudowords that were indicated as not known (correct rejections). The first is believed to indicated vocabulary knowledge, while the second is believed to reflect sophisticated guessing (Huibregtse et al., 2002). This difference suggests that treating the two as equal would be problematic (Mochida & Harrington, 2006). Instead, Huibregtse et al. (2002) recommends a form of correction for guessing. The simplest of which would be to calculate the hit rate minus the false alarm rate, as shown below.

$$P(h) = h - f$$

where *h* is the hit rate and *f* is the false alarm rate. However, Huibregtse et al. (2002) concede that it may underestimate vocabulary knowledge if the false alarm rate is low.

5.8.3 Correction for guessing

Meara and Buxton*'s* (1987) Yes/No Vocabulary test uses a form of correction for guessing (cfg) which takes into account, not just hits and false alarms, but also the proportion of pseudowords that were correctly identified as not known (correct rejection). The formula is shown below.

$$cfg = \frac{h - f}{1 - f}$$

where *h* is the hit rate, *f* is the false alarm rate, and 1-*f* is the correct rejection rate. However, this formula assumes that vocabulary knowledge is categorical (Huibregtse et al., 2002), that a learner either knows a word or is guessing at random (Mochida & Harrington, 2006). It does not consider a learner's particular response style.

5.8.4 I_{SDT}

I_{SDT} uses the proportions of hits and false alarms, includes correction for guessing, and considers participant's response style. The formula is shown below.

$$I_{SDT} = 1 - \frac{4h(1 - f) - 2(h - f)(1 + h - f)}{4h(1 - f) - (h - f)(1 + h - f)}$$

where *h* is the hit rate and *f* is the false alarm rate. Huibregtse et al. (2002) propose that by calculating the intersection of the ROC curves for a learner, a test score that is corrected for response bias can be obtained. The above formula has been obtained by (1) applying a linear transformation to (2) the formula for calculating the intersection of a ROC curve (3) which contains a derivation of Grier's formula for the area under a ROC curve (Grier, 1971, p. 425; Huibregtse et al., 2002, p. 238), resulting in a vocabulary score that ranges between 0 and 1.

5.9 Calculating scores

The hit and false alarm counts, from both the pilot study described in Section 5.4, and the main study described in Section 5.5, have been used to calculate vocabulary scores for the control and AEFL tests, using four scoring methods: *h–f*, correction for guessing (cfg), Meara's scoring matrix (Δm), and I_{SDT}. Proportionate hit rates are used as the control (hits).

5.9.1 Mean vocabulary scores

Table 5.19 shows the mean vocabulary scores from the pilot and main study, for each scoring method.

Table 5.19 Scoring: Mean vocabulary scores for each scoring method

	Pilot study				
	Hits	*h–f*	cfg	Δm	I_{STD}
Control	0.928	0.839	0.923	0.827	0.916
AEFL	0.930	0.863	0.919	0.838	0.928
	Main study				
	Hits	*h–f*	cfg	Δm	I_{STD}
Control	0.996	0.893	0.996	0.892	0.959
AEFL	0.992	0.961	0.992	0.960	0.979

Table 5.20 Scoring: the number of tests with no vocabulary score difference

	Pilot study				
	Hits	*h–f*	cfg	Δm	I_{STD}
Count	41	28	41	24	28
Total	49	49	49	49	49
Percentage	84%	57%	84%	49%	57%
	Main study				
	Hits	*h–f*	cfg	Δm	I_{STD}
Count	92	46	91	45	47
Total	97	97	97	97	97
Percentage	95%	47%	94%	46%	48%

5.9.2 Statistical analysis

A series of two-tailed z-tests were also conducted, to compare vocabulary scores for the AEFL and control tests, for each scoring method. The null hypothesis, $H_{0,}$ was that the two population proportions were equal. In this case, rejecting the null hypothesis meant there was a statistically significant difference between the vocabulary scores for the control and AEFL tests.

For the pilot study, 245 z-tests were conducted in total, i.e. 49 for each of the 5 scoring methods. Table 5.20 shows the number of tests with no vocabulary score difference. Of the five scoring methods, proportionate hit rates (Hits) and correction for guessing (cfg) resulted in the highest number of tests with no statistically significant difference (both 84%), I_{SDT} and *h–f* were in the middle (both 57%), and Meara's original scoring matrix (Δm) was the lowest (49%). For the

Table 5.21 Scoring: valid tests with no statistically significant difference

Pilot study					
	Hits	*h–f*	cfg	Δm	I_{STD}
Count	39	27	39	23	27
Total	44	44	44	44	44
Percentage	89%	61%	89%	52%	61%
Main study					
	Hits	*h–f*	cfg	Δm	I_{STD}
Count	79	45	78	44	46
Total	83	83	83	83	83
Percentage	95%	54%	94%	53%	55%

main study, 485 z-tests were conducted, i.e. 97 for each of the 5 scoring methods. Again, Table 5.20 shows the number of tests with no vocabulary score difference. Like the pilot study, hit rates (Hits) and correction for guessing (cfg) were the highest (95% and 94%), I_{SDT} and *h–f* were in the middle (48% and 47%), and Meara's original scoring matrix (Δm) was the lowest (46%).

If two tests showed no statistically significant difference in hit and false alarm counts, they should also show no statistically significant difference in their resulting vocabulary scores. Table 5.21 shows the percentage of "valid" tests that showed no statistically significant difference in vocabulary scores. Valid tests are those that showed no statistically significant difference in both hit and false alarm counts. For example, in the pilot study, there were 44 tests that did not show a difference in hit and false alarm counts. Of those 44 tests, 89% did not show a statistically significant difference for the proportionate hit rate (hits) and correction for guessing (cfg) scoring methods. In fact, the proportional hit rate (hits) and correction for guessing (cfg) have substantially higher percentages for both the pilot study (89% versus 61%-52%), and the main study (95% and 94% versus 55%-53%).

5.9.3 Score comparisons

Huibregtse et al. (2002) estimated vocabulary scores for four scoring methods – *h–f*, cfg, Δm, and I_{SDT} – based on proportions of hits and false alarms. Although they

Table 5.22 Comparing vocabulary score relationships

Source	Relationship
Huibregtse	$cfg > I_{SDT} \geq h - f > \Delta m$
Pilot control	$cfg(0.923) > I_{SDT}(0.916) > h - f(0.839) > \Delta m(0.827)$
Pilot AEFL	$I_{SDT}(0.928) > cfg(0.912) > h - f(0.863) > \Delta m(0.838)$
Main control	$cfg(0.996) > I_{SDT}(0.959) > h - f(0.893) > \Delta m(0.892)$
Main AEFL	$cfg(0.992) > I_{SDT}(0.979) > h - f(0.961) > \Delta m(0.960)$

did not provide any precise figures, their estimates found that Meara's original scoring matrix always yielded an underestimation, while correction for guessing gave an overestimation when hit proportions were large. Conversely, they found that *h–f* was often comparable, if not identical, to $I_{SDT,}$ unless hit and false alarm rates were very low or very high, then *h–f* was lower. I have interpreted this as the following relationship.

$$cfg > I_{SDT} \geq h - f > \Delta m$$

Table 5.22 shows the relationship between mean vocabulary scores, for the pilot and main study, for the control and AEFL tests, in comparison to the relationship described by Huibregtse et al. (2002). The mean vocabulary scores shown in Table 5.22 are derived from Table 5.19. All four results (except the pilot AEFL, where I_{STD} was higher than cfg, but only by 0.009) showed the same relationship as Huibregtse et al. (2002), in turn supporting their findings.

Mochida and Harrington (2006) used performance results from a Yes/No test to predict scores for the Vocabulary Levels Test (Nation, 1990), a multiple choice receptive vocabulary test. They used identical items for both tests, resulting in a direct comparison between learners' vocabulary scores for each. Mochida and Harrington (2006) evaluated the same four scoring methods as Huibregtse et al. (2002) – *h–f*, cfg, Δm, and I_{SDT} – plus a fifth scoring method – the proportional hit rate. Mochida and Harrington (2006) aimed to determine which method gave the best predictor of vocabulary knowledge, in relation to a learner's scores on the Vocabulary Levels Test. Their results found that all scoring methods were strong predictors of a learner's performance in the Vocabulary Levels Test, but that the hit rate itself was the strongest predictor overall. As shown in Table 5.23, Mochida and Harrington (2006) recorded the Vocabulary Levels Test (VLT) having a mean

Table 5.23 Mean vocabulary scores

VLT	Hits	*h–f*	cfg	Δm	I_{SDT}
0.83	0.82	0.78	0.81	0.76	0.81

Table 5.24 Comparing mean vocabulary scores with the pilot and main study

	M & H	Pilot control	Pilot AEFL	Main control	Main AEFL
Hits	0.82	0.928	0.930	0.996	0.992
cfg	0.81	0.923	0.919	0.996	0.992
I_{SDT}	0.81	0.916	0.928	0.959	0.979
h–f	0.78	0.839	0.863	0.893	0.961
Δm	0.76	0.827	0.838	0.892	0.960

proportion score of 0.83. The closest scoring method to this was the hit rate (0.82), followed by I_{SDT}, correction for guessing (cfg) (both 0.81), and *h–f* (0.78). Meara's original scoring method (Δm) resulted in the lowest proportion (0.76).

Table 5.24 shows the mean vocabulary scores from the control and AEFL tests, in comparison with Mochida and Harrington (2006). Both the pilot and main study resulted in scores higher than those of Mochida and Harrington (2006). However, the results support their findings. Hits have the highest score, while *h- f* and Meara's original matric (Δm) have the lowest. All results showed the same ordering as those seen in Huibregtse et al. (2002), except for the AEFL, where I_{STD} was higher than cfg, but only by 0.009. The control and AEFL scoring results support the findings from both Mochida and Harrington (2006) and Huibregtse et al. (2002), for both the pilot and main study.

5.10 Implications of scoring

The second half of this chapter investigated methods for scoring the EFL Vocabulary Test. Researchers have debated Meara's original scoring method and suggested several alternatives. However, Meara (2010) claims that none are entirely satisfactory. I have used the results from the pilot and main studies to (1) determine which scoring method best evaluates the EFL Vocabulary Test, and (2) compare the results with those produced by other researchers (Huibregtse et al., 2002; Mochida & Harrington, 2006).

For the first, the hypothesis was that, if two tests showed no statistically significant difference in hit and false alarm counts, they should also show no statistically significant difference in their resulting vocabulary scores. For the pilot study, 89% of tests that had no difference in hit and false alarm counts also showed no difference in vocabulary scores for both the proportionate hit rate (Hits) and the correction for guessing (cfg) methods. Likewise, for the main study, 95% of tests with no difference in hit and false alarm counts also showed no difference in vocabulary scores for Hits, and 94% for cfg. This suggests that the proportionate hit rate (Hits) and correction for guessing (cfg) methods are most suited to the EFL Vocabulary Test.

For the second, Huibregtse et al. (2002) evaluated four scoring methods, ranking Meara's original method and three alternatives. They found that Meara's method produced the lowest mean vocabulary score, underestimating learners' abilities, and that correction for guessing (cfg) produced the highest, creating an overestimation. However, Mochida and Harrington (2006) used EFL vocabulary scores to predict scores for the Vocabulary Levels Test, and although Huibregtse et al. (2002) suggest that correction for guessing gives an overestimation, Mochida and Harrington (2006) found that the correction for guessing and proportionate hit rate methods were the best predictors of Vocabulary Levels Test scores. The results from the pilot and main study, shown in Section 5.9.3, support their findings.

Both of the above findings support proportional hit rate and correction for guessing as the preferred scoring methods. However, calculating a vocabulary score based on the hit proportions alone does not take false alarms into account. Likewise, correction for guessing (cfg) does not consider false alarms when learners select all real words correctly; for example, a learner who has 40 hits and 0 false alarms will receive the same score as someone with 40 hits and 20 false alarms. This supports Meara's claims that none of the alternative scoring methods are entirely satisfactory, and suggests that there is need for more research in this area.

This concludes the study of vocabulary testing and generating domain-specific tests. The next chapter begins investigating the third challenge, integrating language resources into online courses.

Chapter 6

Designing an integrated system

Integrate /ˈɪntɪgreɪt/

(verb) (1) to form into one whole; to make entire; to complete; to renew; to restore; to perfect. (adjective) (2) composed and coordinated to form a whole.

(Wiktionary, 2019)

In teaching and learning vocabulary, three things must be considered: (1) the learners existing level of knowledge, (2) the genre of text that best suits them, and (3) growing and expanding their vocabulary as their skills develop. The first requires vocabulary testing, and the second is determined by their purpose in language learning, both of which were explored in the first half of this thesis. The third involves exposure to new and different vocabulary.

Vocabulary learning often focuses on initial vocabulary acquisition and assumes little prior knowledge of the language. This is not enough for online learners who wish to extend their existing vocabulary. The sheer quantity of vocabulary that learners need to acquire requires constant exposure. Even advanced learners continually develop and improve their reading ability by acquiring and consolidating new vocabulary. This chapter outlines the planning and design

considerations for an application that integrates language resources into online courses. It answers four questions: (1) are online courses suited to language learning? (2) Do applications already exist that work on top of online courses to provide language resources? (3) What type of vocabulary should be supported? (4) What type of language resources should be provided?

6.1 Content-based language learning

Content-based language learning, described in Section 2.6, is the dual-learning concept of learning a subject through a foreign language, and learning the foreign language by studying the subject (Marsh, 2002). There is no "one" approach to content-based learning. In fact, Stoller and Grabe (1997) identify eight.

1. The Centre for Applied Linguistics (CAL) approach
2. The English for Academic Purposes (EAP) instruction
3. The university-level foreign language approach
4. The discoursal knowledge structures approach
5. The genre-based approach to K-12 literacy instruction
6. The immersion programs approach
7. The cognitive academic language learning approach
8. The whole language instruction approach

Each approach is centred on different aspects of learning. The first focuses on integrating content and language, language comprehension problems, and assessing vocabulary and content knowledge (Crandall, 1992; Short, 1991, 1993, 1994). The second focuses on sheltered instructions, adjunct instruction, and theme based instruction (Adamson, 1993; Brinton & Holten, 1989; Snow, 1993; Snow, Met, & Genesee, 1989). The third focuses on organizing language instruction around cultural, historical, geographical, political, and literary themes (Krueger & Ryan, 1993; Musumeci, 1993; Straight, 1994; Wesche, 1993) or providing additional topic-based resources in a foreign language (Jurasek, 1993; Sudermann & Cisar, 1992); and so on. However, Stoller and Grabe (1997) suggest that each overlaps much more than former research indicates, and propose a new approach based on this, the Six-T's Approach, which can be defined by six curricular components: Themes, Texts, Topics, Threads, Tasks, and Transitions.

Themes are the central ideas that major curricular units are organized around, for example, *insects*, *the solar system*, *demography*, or *historical*

monuments. Normally a class explores multiple themes throughout a course. *Texts* refer to the written and aural content used by a class, and should be selected based on: interest, relevance and instruction, length, and coherence and connection. *Topics* are sub-categories of themes and explore more specific elements, for example, *the solar system* could be broken into topics such as *humans in space, technology in space* and *research in space,* or *Earth, Venus* and *Mercury. Threads* link themes together to form major curricular units and are often abstract concepts, for example *responsibility, ethics,* or *contrasts. Tasks* are instructional units for content, language and strategy, and are planned based on how texts are used, for example, language acquisition activities, interactive communication, and discourse organization. Finally, *transitions* are explicit actions that link topics with themes and link themes with major curricular units. Some transitions are used to shift emphasis (topical transitions), while others are used to navigate through tasks (task transitions).

Stoller and Grabe (1997) suggest that this approach can be used to develop coherent content-based curriculum. The Six-T's approach views content as the driving force for learning, opposed to that of structural, communicative, or task-based approaches. In fact, this content-based approach fits well with the structure of online courses. Course content tends to be broken into weekly chunks (themes), each of which are further categorized into steps (topics). Each step contains some type of content or exercise, for example, articles and videos (texts) or discussions and quizzes (tasks). Finally, the entire course is tied together (threads) to form an over-arching subject, and learners are encouraged to advance from one step to the next (transitions).

6.2 Online language applications

Online support for content-based language learning is limited. Applications exists that aid teachers to integrate language into classroom curricula (CARLA, 2019; Tedick & Cammarata, 2010); software exists that provides learners with online language resources, such as dictionaries and glosses ("Oxford English Dictionary," 2018); and courses exist that provide learners with online topic-based content (FutureLearn, 2019b); but to my knowledge, the three have not yet been combined to provide content-based language courses freely or easily available online.

Even so, there are a variety of computer assisted language learning applications, some of which provide language resources for online written content. Computer-assisted language learning (CALL) is a term used to describe computer-based applications for teaching or learning a language. CALL includes a wide range of applications, from virtual learning environments and web-based distance learning to corpora and concordance software. The following section introduces six existing CALL applications that provide language resources for online written content.

6.2.1 Web-based systems

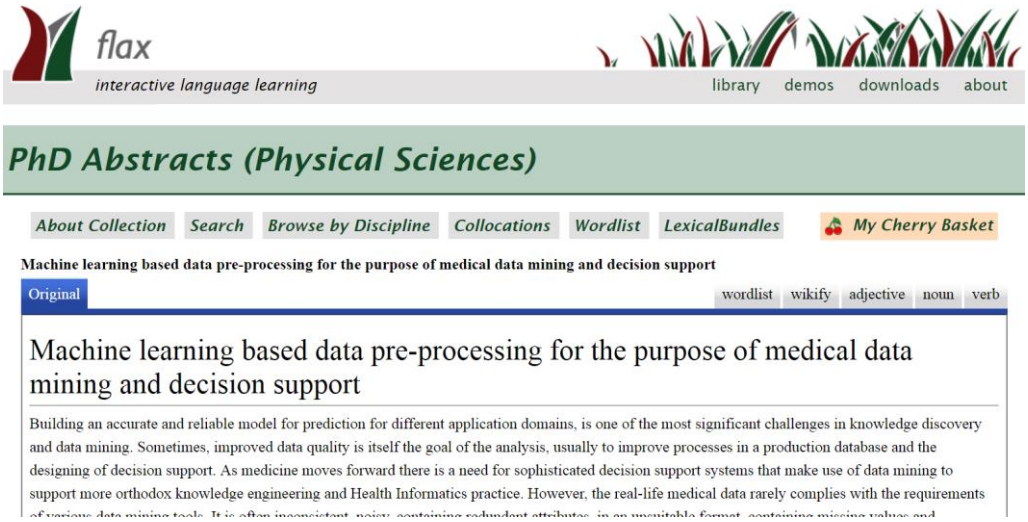
This section introduces three web-based applications: *FLAX*, *LingQ*, and *ParallelText.io*.

6.2.1.1 FLAX

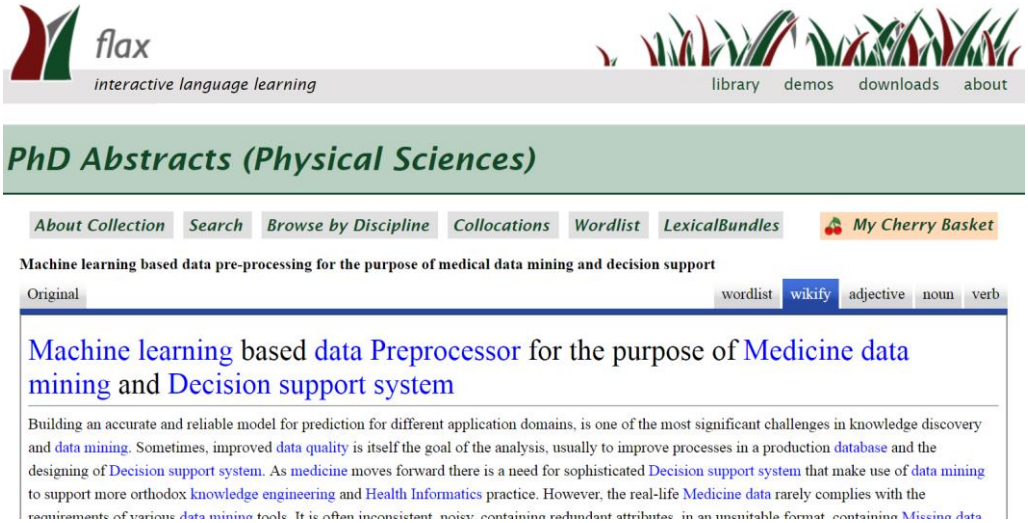
FLAX is an online language learning application. It contains collections of digitized text from a variety of sources, including PhD abstracts from the British Library's E-Theses Online Service, online course material from edX and Coursera, and a selection of texts created by their users. A PhD abstract from the British Library's Physical Sciences collection is shown in Figure 6.1a.

FLAX highlights words, collocation, and "wikified" terms within text. Words are highlighted based on whether they are in the first or second 1000 most frequent words according to the General Service List (West, 1953), academic words according to the Academic Word List (Coxhead, 1998), topic-specific words, or keywords. Collocations are highlighted based on their part of speech, for example, collocations including adjectives, nouns, or verbs. Finally, both single and multi-word terms that exist as articles in Wikipedia are highlighted, for example *machine learning*, *data*, and *pre-processor*, as shown in Figure 6.1b.

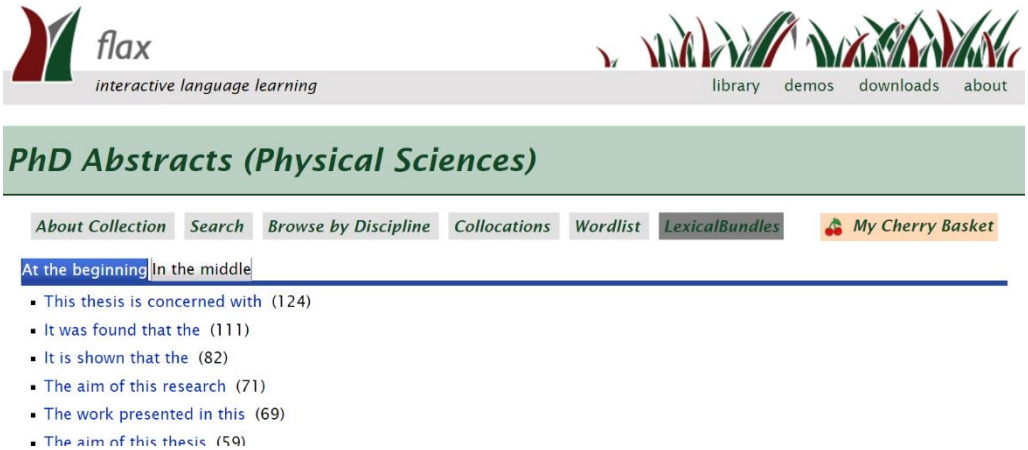
FLAX allows learners to search through all the text in a collection, returning sentences, paragraphs, or entire articles that contain a target word or phrase. It also provides lists of the words, collocations, and lexical bundles (shown in Figure 6.1c) that appear within a collection, and displays descriptions from Wikipedia for "wikified" terms.



a) A PhD abstract from the Physical Sciences collection



b) Highlighting Wikipedia concepts in text



c) Lexical bundles that appear in the Physical Sciences collection

Figure 6.1 Online learning application: FLAX

6.2.1.2 LingQ

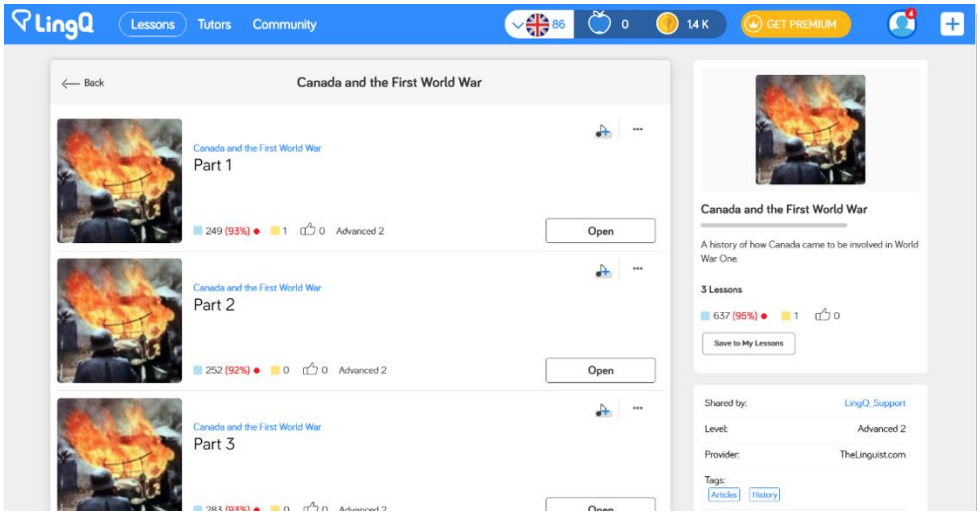
LingQ is a web-based platform where learners can develop their vocabulary by reading content that interests them. It allows learners to select a collection of categories (travel, history, leisure, food, etc.) and estimate their language competency (beginner, intermediate, advanced), providing them with a selection of articles that match. Content is either provided by LingQ itself, obtained from an external source, such as the Huffington Post, or provided by their users. Articles can be completed in recommended lesson steps, where one article leads on to another, or in guided courses, where there are multiple articles on one topic, such as the "Canada and the First World War" course shown in Figure 6.2a.

Articles are displayed as pages, where learners can view small portions of text at a time, clicking through pages to see more. Words that have not been seen before are highlighted blue (Figure 6.2b) and can be clicked to retrieve dictionary definitions, example sentences, or translations. Each are retrieved either from user-specified information, or from an external dictionary application, which LingQ provides a selection of. However, this means that any additional linguistic information is only as good as the external application used, and is never content specific.

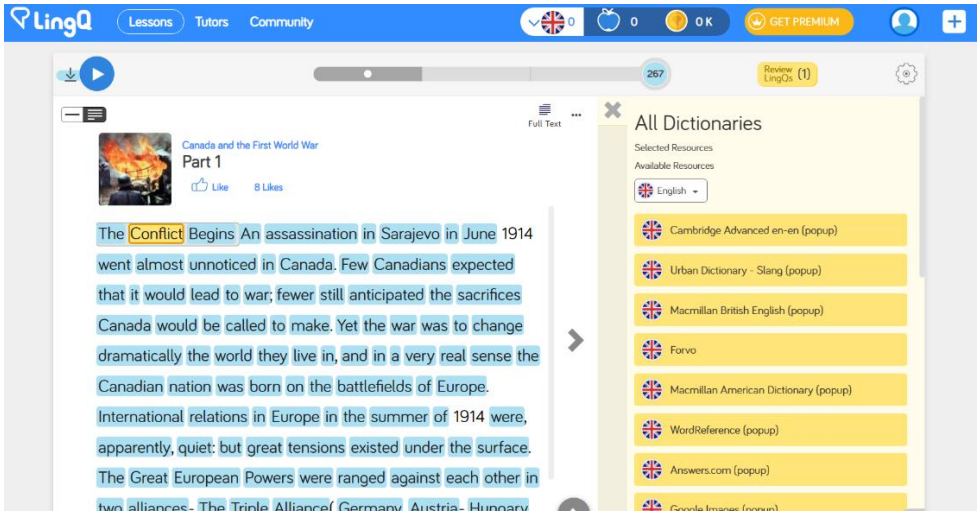
LingQ also highlights phrases, as shown in Figure 6.2c. Once a word has been clicked, the phrase containing it is highlighted grey. This phrase can then be looked-up in the same external dictionary applications that are provided for individual words. LingQ provides learners with lists of words and phrases that they have interacted with, and language games to help them learn. However, the online application itself can at times be difficult to interact with, and the consequences of clicking certain items is often unclear.

6.2.1.3 ParallelText.io

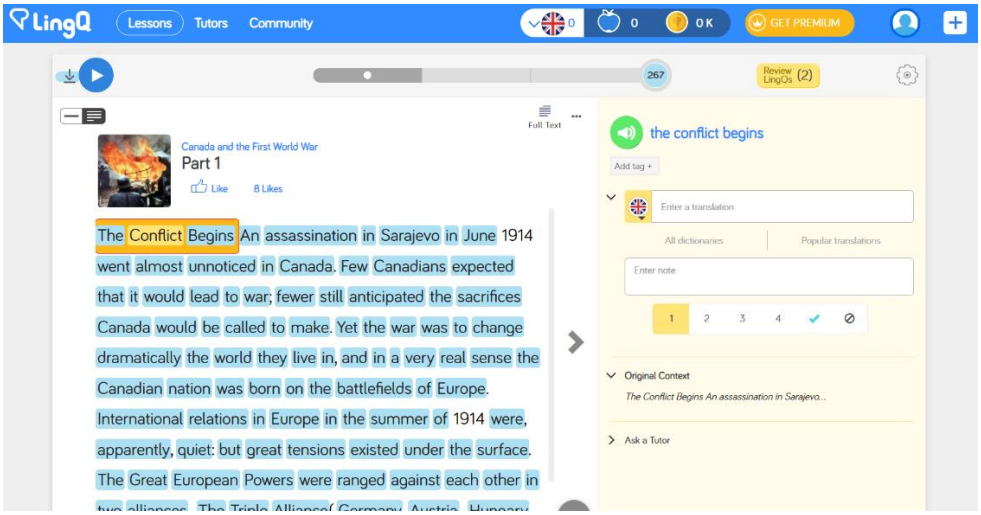
ParallelText.io is a website that helps learn a language by reading books online. It provides learners with reading material in the form of well-known novels such as *Jane Eyre* by Charlotte Bronte or *Tom Sawyer* by Mark Twain. ParallellText.io displays text, either solely in the language that learners are trying to learn (Figure 6.3a), or in a split view, displaying the same text in both L1 and L2 language (Figure 6.3b). It allows learners to translate paragraphs of text (Figure 6.3c) and reads the text out loud in either language.



a) Guided courses



b) Highlighting words within text

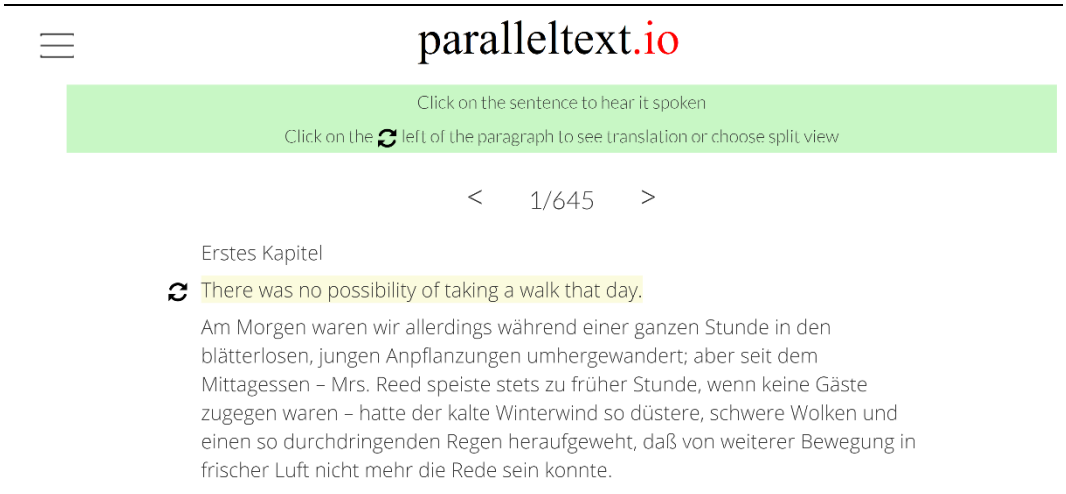


c) Highlighting phrases within text

Figure 6.2 Online learning application: LingQ

a) Compact view

b) Split view



c) Translating text

Figure 6.3 Online learning application: ParallelText.io

ParallelText.io focuses on using reading as a learning resource rather than augmenting text with additional language resources. It provides translations but does not support dictionary definitions, example sentences, or any other language resources.

6.2.2 Browser extensions

This section introduces three browser extensions: *ReadLang*, *BeFluent*, and *KnowbleReader*, each of which are available on Google Chrome, but are not available for Firefox or MS Edge.

6.2.2.1 ReadLang

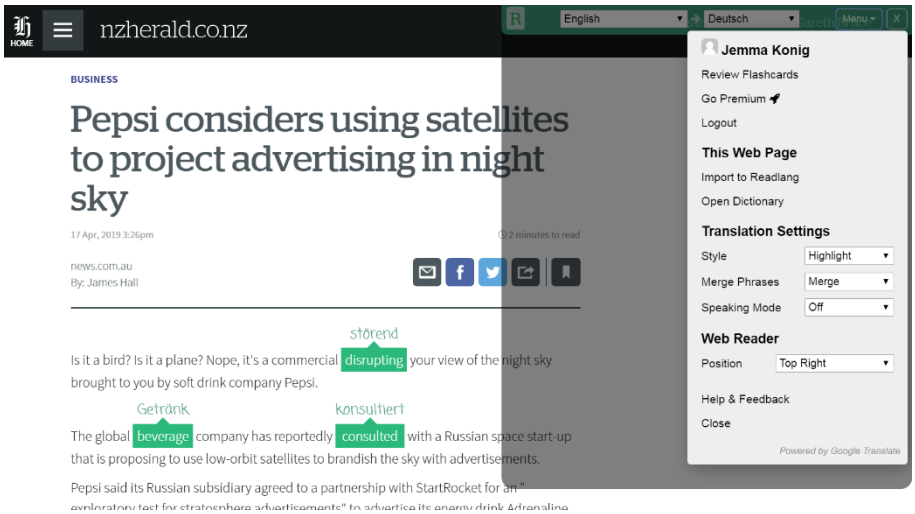
ReadLang is a Chrome extension that works on top of existing websites. It embeds event handlers into the words on a page, allowing learners to click words to view translations. ReadLang can be installed using the Chrome web store and works on most webpages. It does not work on interactive pages like Facebook, but is targeted towards news, blog, and short story sites.

ReadLang does not change any of the content on the page until after words are clicked, as shown in Figure 6.4a which illustrates an article on the NZ Herald site after ReadLang has been installed. When a word is clicked, ReadLang highlights it and overlays a translation in the language specified by the learner. Figure 6.4b shows an article after a learner has clicked on three words: *disrupting*, *beverage* and *consulted*.

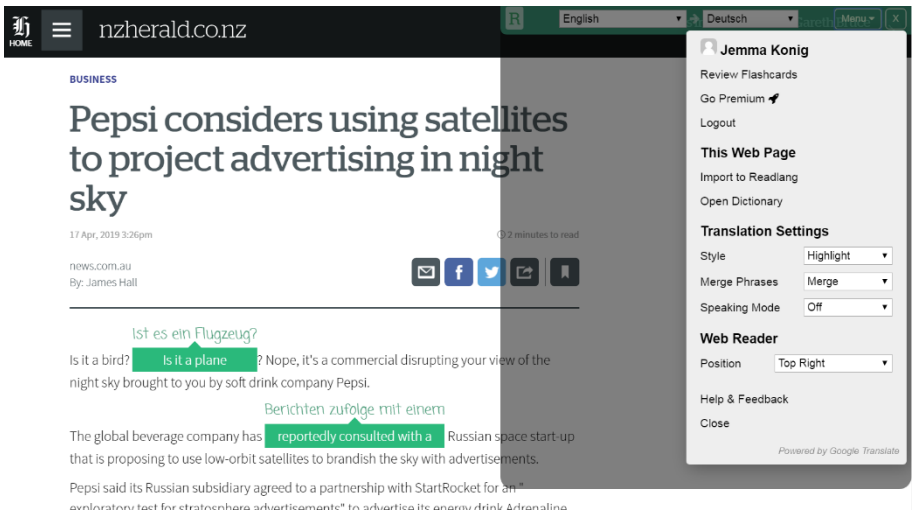
Phrases can be selected by clicking on two or more consecutive words. However, these are not special phrases like collocations or lexical bundles, but rather a collection of consecutive words selected by the user. There is no constraint to force users to select well-formed phrases, they can select any consecutive words that they wish (up to six words long). Figure 6.4c shows the phrases *is it a plane* and *reportedly consulted with a*. Allowing learners to specify phrases allows them to translate a section of text. However, it should be noted that these phrases are necessarily not well suitable for language productivity.

Finally, learners can opt to “Open Dictionary” from the ReadLang menu. Doing so opens an external dictionary application that navigates to an entry for the last word or phrase that was clicked. Using an external dictionary means that any additional linguistic information is only as good as the dictionary being used and is

a) A NZ Herald article with ReadLang installed



b) Using ReadLang to click on words



c) Using ReadLang to select phrases

Figure 6.4 Online learning application: ReadLang

never content specific. The external dictionary applications also tend not to find helpful information for phrases. This is understandable since the phrases are defined by the user, rather than pre-defined collocations or lexical bundles.

6.2.2.2 BeFluent

BeFluent is another Chrome extension. Like ReadLang, it uses content on existing webpages as a language resource, allowing learners to click on words within the text. Installing BeFluent does not change any of the content on the page, as shown in Figure 6.5a, even after learners click on words. Instead, a dialog is opened that contains additional lexical information, as illustrated in Figure 6.5b.

The BeFluent dialog provides learners with synonyms, a dictionary definition, collocations, derivations, and a translation. It also provides learners with the original sentence, which when expanded, shows collocations, for example the noun phrase *soft drink* and the phrasal verb *follow through* shown in Figure 6.5c. It also provides users with external dictionary details, by clicking "details" at the bottom of the dialog. However, the link appends the desired search term, for example "`/Dictionary/en/pitch/?pos=noun`" for *pitch*, to the URL of the current page, for example "`https://www.nzherald.co.nz/Dictionary/en/company/?pos=noun`", which does not result in a successful link.

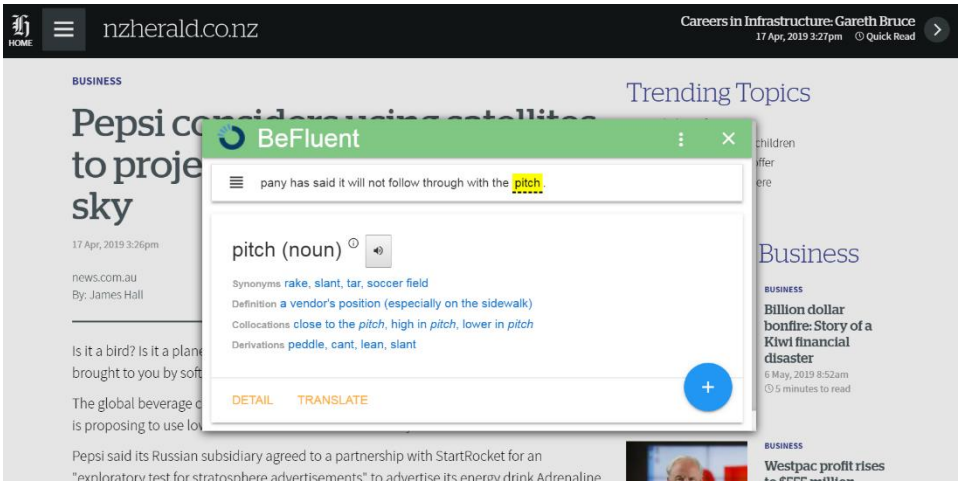
Finally, BeFluent allows learners to save words to a vocabulary list. This requires learners to register for an account. However, once they have, they are able to click on the blue floating action button, in the BeFluent dialog, to add words to their vocabulary.

6.2.2.3 KnowbleReader

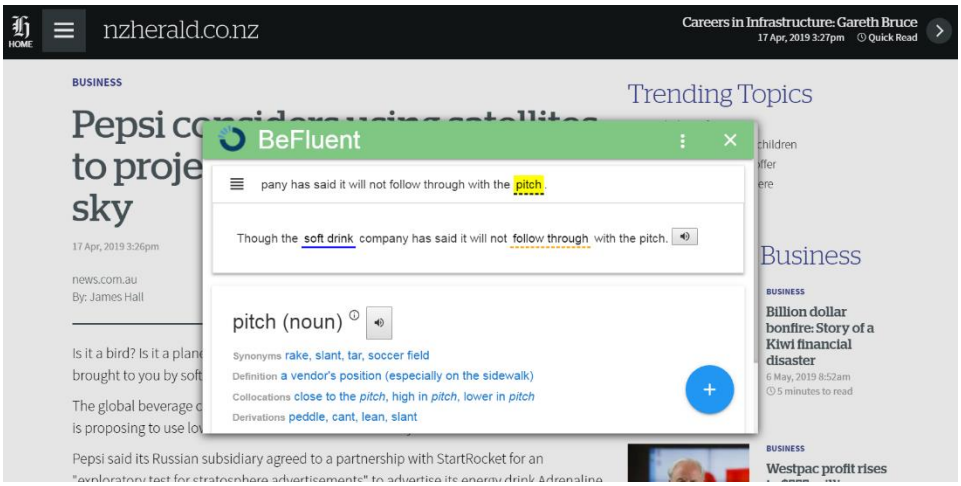
Finally, KnowbleReader is a third Chrome extension. It too is available from the Google web store and works on top of existing content. However, rather than working on most webpages, KnowbleReader is tailored towards a selection of news sites. It allows learners to pick from a selection of news articles, navigates to the corresponding site, and underlines keywords within the text. Figure 6.6a shows a news article from the BBC before KnowbleReader was installed, while Figure 6.6b shows how KnowbleReader underlines keywords in text in the same article.

Keywords are underline orange, and are highlighted when they are clicked. Figure 6.6b shows the highlighted word *geological*, which has been clicked, and

a) A webpage once BeFluent has been installed



b) Using BeFluent to click on words

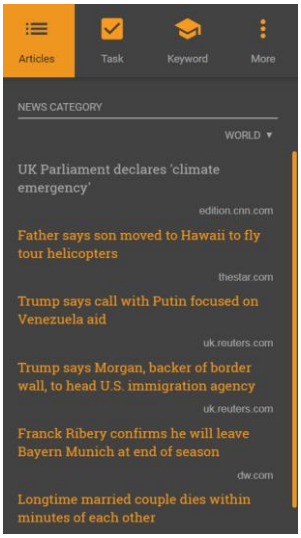


c) Expanding the dialog to view highlighted phrases

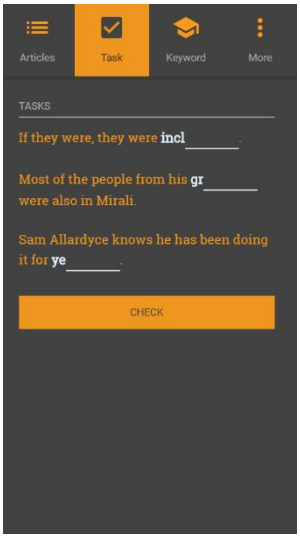
Figure 6.5 Online learning application: BeFluent

a) News article without KnowbleReader installed

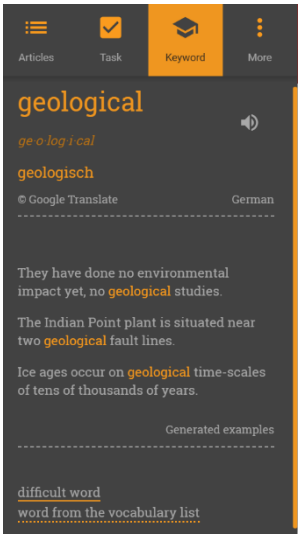
b) Highlighting words and selecting an article



c) News article lists



d) Fill-in-the-blank tasks



e) Lexical information

Figure 6.6 Online learning application: KnowbleReader

keywords like *newly* and *dinosaur*, which are underlined. Once a word has been clicked, its translation and a set of example sentences are shown in the KnowbleReader panel, as illustrated in Figure 6.6b. Underlined words can be selected by using a single click. However, any word can be selected by double clicking, regardless of whether it has been identified and underlined as a keyword.

The KnowbleReader panel contains three tabs: *articles*, *task*, and *keyword*. The *articles* tab provides learners with a list of articles that are supported by KnowbleReader, as shown in Figure 6.6c. They can be filtered by categories such as world, business, entertainment, and sport. Learners click on articles from this list to load them in the webpage and start reading. The *task* tab provides learners with fill-in-the-blank tasks that relate to the current article, as shown in Figure 6.6d. The *keyword* tab displays lexical information about clicked words, as described above and shown in Figure 6.6e. However, it does not support dictionary definitions or phrases like collocations and lexical bundles. It is also limited to a very restricted set of pre-defined news articles and does not work on any other type of webpage.

6.3 Feature comparison

A feature comparison was conducted between the six systems reviewed above, outlining the type of language support existing applications provide.

6.3.1 Methodology

The comparison explored eleven features, divided into three categories: text enhancement, lexical item support, and language resources, based on the language support provided by the existing systems. The full list of features is as follows.

Text enhancement:

1. Does the system integrate itself into the content on a page?
2. Does the system highlight items within the text?

Lexical item support:

3. Does the system provide support for words?
4. Does it provide support for collocations?
5. Does it provide support for lexical bundles?
6. Does the system disambiguate single and/or multi-word lexical items?

Language resources:

7. Does the system provide dictionary definitions?

8. Does is provide example sentences?
9. Does it provide related collocations?
10. Does it provide descriptions for disambiguated terms?
11. Does the system provide translations?

6.3.2 Results

Table 6.1 displays the results of the feature comparison. The first, FLAX, scored 8 out of 11. It highlights items in text, supports words, collocations, lexical bundles, and disambiguates terms, and provides example sentences, related collocations, and descriptions for disambiguated terms. It does not integrate itself into existing content and does not provide definitions or translations.

LingQ scored 5 out of 11. It highlights items in text and supports words, dictionary definitions, example sentences, and translations. It does not integrate itself into existing content, and does not support collocations, lexical bundles, disambiguated terms, related collocations, or descriptions for disambiguated terms. LingQ does highlight phrases. However, these are not specialised phrases like collocations and lexical bundles, rather they are phrases based on sections of text. For example, it highlights the phrase *in a country a long way off*.

ParallelText.io scored 2 out of 11. It highlights items in text and supports translation but does not meet any of the other features. It does not support words, collocations, lexical bundles, or disambiguated terms, and it does not provide definitions, example sentences, related collocations, or descriptions for disambiguated terms. Instead, it provides audio and visual translations for sections of text.

ReadLang scored 4 out of 11. It integrates itself into existing content, supports words, and provides definitions and translations. It does not highlight items in text, support collocations, lexical bundles, or disambiguated terms, and does not provide example sentences, related collocations, or descriptions for disambiguated terms. Although ReadLang does not support collocations or lexical bundles, it does support the selection of less specific phrases. It allows learners to select a string of consecutive words and provides definitions and translations for them, where it can.

BeFluent scored 6 out of 11. It integrates itself into existing content, supports words and collocations, and provides definitions, related collocations and

Table 6.1 Feature Comparison

		FLAX (web)	LingQ (web)	ParallelText (web)	ReadLang (extension)	BeFluent (extension)	Knowble (extension)
Text enhancement	Integrated into content				✓	✓	✓
	Highlights items in text	✓	✓	✓			✓
Lexical item support	Words	✓	✓		✓	✓	✓
	Collocations	✓				✓	
	Lexical bundles	✓					
	Disambiguated terms	✓					
Language resources	Dictionary definitions		✓		✓	✓	✓
	Example sentences	✓	✓				✓
	Related collocations	✓				✓	
	Disambiguated descriptions	✓					
	Translation		✓	✓	✓	✓	✓

translations. It does not highlight items in text, support lexical bundles or disambiguated terms, and does not provide example sentences or descriptions for disambiguated terms. Although it supports collocations, it does not identify them until a word has been clicked. Even then, it only identifies them in the current sentence.

KnowbleReader scored 6 out of 11. It integrates itself into existing content, highlights items within text, supports words, and provides definitions, example sentences and translations. It does not support collocations, lexical bundles or disambiguated terms, and does not provide related collocations or descriptions for disambiguated terms. Although it integrates itself into existing content, it only works on top of a set of pre-defined news websites.

6.3.3 Discussion

The purpose of this comparison was to sample the language features provided by existing online language applications, and in turn, determine which language features would be included in the integrated system. None of the existing applications met all eleven features, but FLAX was the closest with eight. This was followed by BeFluent and KnowbleReader, which both scored six. Only FLAX supported disambiguation in any way. However, all but FLAX supported translation.

As a result of this feature comparison, the following two areas are investigated below: (1) different types of vocabulary – words, collocations, lexical bundles, and disambiguated terms – and (2) different types of language resources – definitions, example sentences, related collocations, and disambiguated descriptions.

6.4 Supporting vocabulary acquisition

Vocabulary acquisition has long been accepted as a crucial component in learning a second language (Coady & Huckin, 1997). However, vocabulary is not just words, but an entire gamut of items – words, collocations, lexical bundles. To achieve lexical competence, learners must master a significantly larger collection of vocabulary knowledge than just individual words.

6.4.1 Learning words

Great importance has been placed on the most frequent 2000 words, which cover approximately 80% of English writing, suggesting that their successful acquisition will be sufficient for a basic understanding of the English language (Milton, 2009; Nation, 2001). This is often achieved by learning wordlists, which, with only 2000 words, seems a reasonable approach. Yet in order to cover 99% of the English language in text, one needs to know approximately 44,000 words (Nation, 2001). Learning single-word items now seems like a daunting task. However, it is one that can be achieved through content-based language learning.

6.4.2 Learning phrases

A learner's lexical competence relies not just on individual words but also on multi-word items (phrases). Learners who are familiar with phrases can express ideas simply and precisely, thereby communicating more effectively. There are several types, but this thesis focuses on two: collocations and lexical bundles.

1. *Collocations* (described in Section 2.1.2) are sequences of two or more words that occur together more frequently than by chance, and that hold semantic meaning (Nation, 2001; Nattinger & DeCarrico, 1992).
2. *Lexical bundles* (described in Section 2.1.2) are recurrent multi-word items that carry meta-discourse functions; the understanding of which improves learners' accuracy and fluency (Li, 2016).

6.4.2.1 Learning collocations

Second language learners use collocations to increase their fluency. For example, consider the different uses for the words 'make' and 'do'. You *make* a cup of tea, and *do* the laundry, you do not *do* a cup of tea and *make* the laundry. Although both are grammatically correct, and can be understood, the first is much more salient.

Research into the measurement of collocations is scarce. However, one study has shown that collocation knowledge can be learnt incidentally from reading, and that the rate of acquisition is similar to that of single-word lexical items (Pellicer-Sánchez, 2017).

6.4.2.2 Learning lexical bundles

Lexical bundles can be defined by three characteristics: they are frequently occurring, are not idiomatic in meaning and not perceptually salient, and are often not structurally complete (Biber & Barbieri, 2007), for example, the phrase *in the case of* is categorized as a lexical bundle because it is frequent and widely distributed across texts (Li, 2016), does not have meaning and is not salient when standing alone, and is not a complete grammatical sentence.

Research into the measurement of lexical bundles is even scarcer than collocations. However, one study places great importance on lexical bundles and indicates that their acquisition significantly helps students' writing ability (Kazemi, Katiraei, & Rasekh, 2014).

6.4.3 Disambiguation

Both single and multi-word lexical items can be ambiguous when seen outside of context. The word *Weka* can refer to the flightless New Zealand bird, or a suite of machine learning software. The phrase *Big Data* can refer to large collections of data, or an electronic music project. Researchers place great importance on context in vocabulary learning, because the meaning of a word often changes dramatically depending on the context in which it is used. This highlights limitations in studying decontextualized vocabulary, such as rote learning wordlists or memorizing definitions. Researchers have shown that context is important for learners, at times allowing them to disambiguate words based solely on the surrounding context (Nagy, 1995).

6.5 Providing language resources

There is a plethora of language resources available to learners, in both electronic and physical form. Learners can use dictionary definitions to retrieve word meanings, example sentences to see them used within context, related phrases to see how their language can be expanded, and disambiguated descriptions to retrieve their meaning based on context.

6.5.1 Definitions

Dictionary definitions provide learners with descriptions for single words. They often include multiple meanings based on different parts of speech and can include

etymology, derived forms, synonyms, and antonyms.

A study was conducted in 1994 that explored the use of dictionary definitions for vocabulary acquisition. It found that a significant number of new words were learnt through reading alone, but that students who used dictionaries learned more new words than those that did not (Knight, 1994).

6.5.2 Example sentences

Example sentences show learners how vocabulary is used within context. The sentence *the aim of the course is to dispel the mystery that surrounds data mining* shows the use of the single word dispel, while *this course introduces you to practical data mining using the Weka workbench* shows use of the collocation *data mining*.

A 1991 study exploring the use of example sentences for vocabulary acquisition demonstrated that information processed at a semantic level increases information retention (Brown & Perry Jr, 1991), supporting the use of example sentences for vocabulary acquisition. Another study showed that using example sentences allows learners to explore the appropriateness of lexical bundles in multiple contexts before using them in their own speech or writing (Li, 2016).

6.5.3 Related collocations

Related collocations can be used to expand and enrich learners' vocabulary, for example, the collocations *data mining*, *data types*, and *data structures* illustrate how the word *data* can be used in a phrase. This is not restricted to single words. Collocations like *data mining algorithms*, *data mining techniques*, and *data mining models* can be used to illustrate how the collocation *data mining* can be expanded.

Wu (2010) explored the use of related collocations for vocabulary acquisition when reading, using related collocations to expand and enrich a learner's knowledge. As a result, she developed a collocation learning system that highlights collocations within text and provides learners with lexical information relating to them.

6.5.4 Disambiguated descriptions

Dictionary definitions have been shown to assist vocabulary acquisition. However, they often include multiple descriptions and do not disambiguate words based on

their context. They also often do not support phrases. Disambiguated descriptions allow learners to view information on both words and phrases, providing information that is context dependent.

The Wikipedia Miner toolkit, developed by Milne (2010) at the University of Waikato, is an open source system that can be used to augment text with Wikipedia information. It identifies and disambiguates words and phrases within text, referred to as *concepts*, and retrieves Wikipedia content and concepts that relate to them (Milne & Witten, 2013). Wikipedia Miner can be used to provide learners with disambiguated descriptions from Wikipedia, in turn aiding them in their vocabulary acquisition.

6.6 An integrated language system

Content-based language learning is a technique that integrates language teaching into courses without disturbing the original content. Learners acquire vocabulary from the natural, contextualised language of the topic, with emphasis on the acquisition of both single-word and multi-word lexical items. This section first explores the "noticing hypothesis", which is used in language teaching to emphasise the features of a language, then outlines a design summary for F-Lingo, the integrated language system. F-Lingo provides resources for content-based language learning, integrating itself into online courses without disturbing the original content.

6.6.1 The noticing hypothesis

One notion in language acquisition is that of "noticing", that noticing the features of a language facilitates learning them (Uggen, 2012). The "noticing hypothesis", proposed by Schmidt (1990), claims that noticing through conscious attention is a necessary pre-requisite for second language acquisition. Second language learners cannot learn the features of a language if they do not consciously notice them; and only what has been noticed can be acquired (Schmidt, 1990; Uggen, 2012). The necessity of noticing has received criticism from some researchers (S. E. Carroll, 1999; Gass, 2017; Truscott, 1998). Some have produced L2 data that dispute it (Izumi, 2002; Leow, 2001), while others have produced L2 data that support it, but as a facilitator rather than a necessary pre-requisite (Alanen, 1995; Jourdenais, Ota, Stauffer, Boyson, & Doughty, 1995).

Schmidt (2001) developed a weaker version of the noticing hypothesis in response to critiques. Thus, the common understanding today is that noticing and awareness facilitate, but are not essential to, language learning (Gass, Svetics, & Lemelin, 2003; Izumi, 2002; Uggen, 2012).

Noticing in written language involves visual input enhancement, by manipulating the appearance of text through italics, bolding, or highlighting. Alanen (1995) investigated how enhancement affected the acquisition of locative suffixes and consonant alternations in Finnish. She used visual highlighting through italics to focus learners' attention on linguistic forms, and found that it had a facilitating effect on their recall and use. Jourdenais et al. (1995) investigated whether input enhancement through textual modification made L2 forms more noticeable. They highlighted simple past tenses and imperfect verb forms in a sample Spanish text, and found that learners who were exposed to the highlighted text used more of the target forms than those who were not. This has led us to explore the notion of enhancing online course content, highlighting vocabulary within the text that learners are reading.

6.6.2 Design summary

F-Lingo has been designed based on the areas investigated in this chapter, and includes four main features, as shown in Figure 6.7: identifying vocabulary, highlighting vocabulary in text, providing language resources, and integrating modal dialogs.

6.6.2.1 Identifying vocabulary

First, based on the vocabulary items supported by existing systems, and outlined in the feature comparison (Table 6.1), F-Lingo identifies words, phrases (collocations and lexical bundles), and concepts (disambiguated single and multi-word items) in the text on a page. The following chapter describes how they are identified (Section 7.2).

6.6.2.2 Highlighting vocabulary in text

Once they have been identified, F-Lingo highlights words, phrases, and concepts within the page. The noticing hypothesis supports the notion of enhancing content

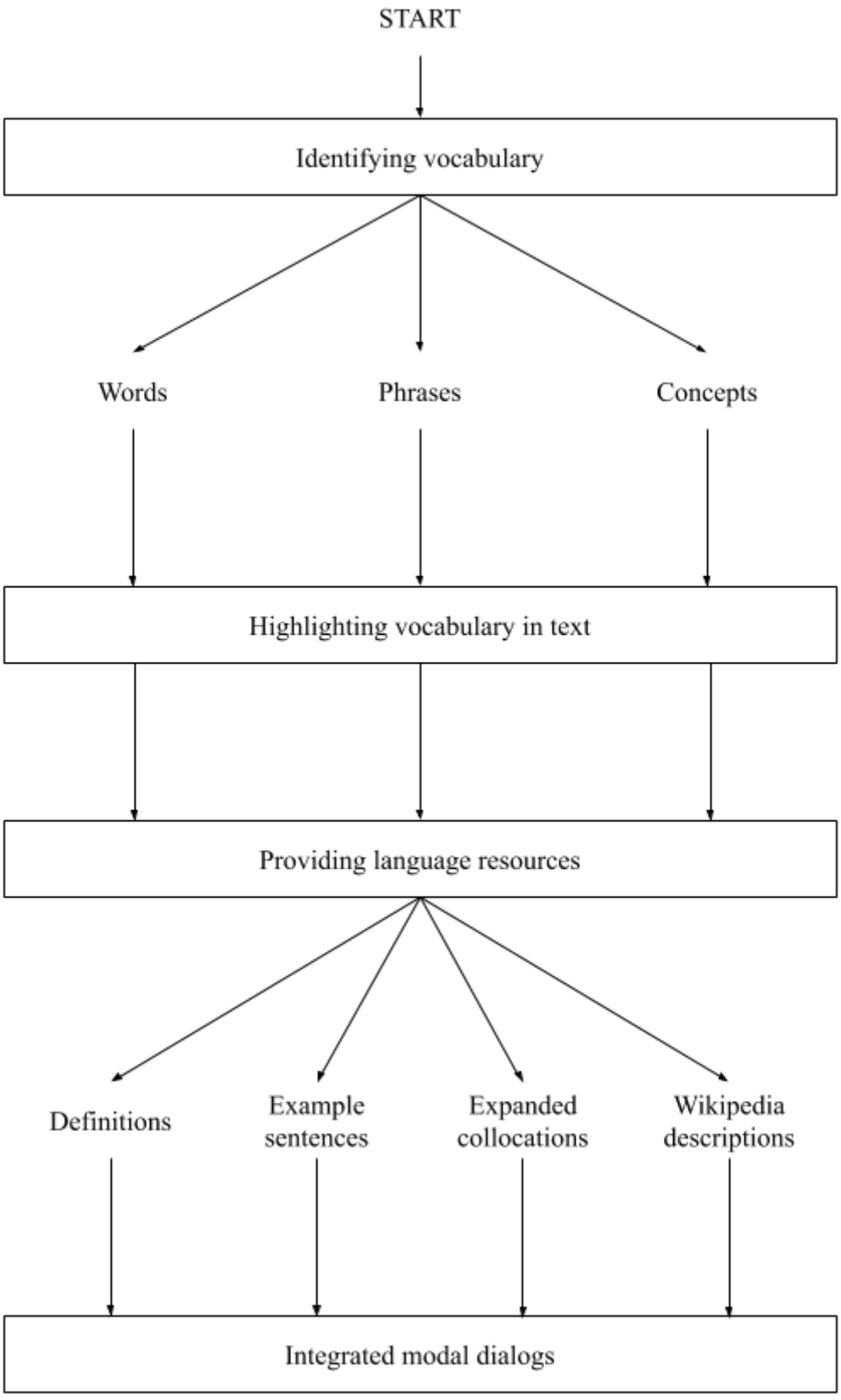


Figure 6.7 Design summary for F-Lingo

by highlighting vocabulary within text, as is done in four out of the six systems evaluated above. Section 7.3 gives details of how this is implemented.

6.6.2.3 Providing language resources

F-Lingo provides four language resources, based on those provided by the existing systems: dictionary definitions, example sentences, expanded collocations, and disambiguated descriptions (Section 7.4). Although most of the evaluated systems also provide translations, they are not included in the integrated system at this stage, but would make a worthwhile future addition.

6.6.2.4 Integrated modal dialogs

Learners can view language resources by clicking on the highlighted items within text. This will open a modal dialog that floats above the original page content. Using modal dialogs gives learners additional lexical information without disrupting the original content on the page.

6.6.3 Additional features

As well as its four main features, F-Lingo also provides learners with three additional features: page-specific wordlists, course-specific vocabulary tests, and seamless integration into an online MOOC platform.

6.6.3.1 Page-specific wordlists

Items that have been highlighted within the text can also be listed down the side of each page, providing learners with a summary of the keywords, phrases, or concepts that appear on the page (Section 7.5). Learners can use these as target wordlists or as quick access to highlighted items.

6.6.3.2 Course-specific vocabulary tests

F-Lingo provides learners with the ability to take vocabulary tests. The tests are built specifically for each course, using vocabulary that appears within them, and can be done as many times as the learner likes, and at any time during their learning process. The development of domain-specific vocabulary tests was covered in Chapter 5, and their integration into F-Lingo is outlined in Section 7.6.

6.6.3.3 Platform-specific integration

Finally, F-Lingo has been designed to be as unobtrusive as possible. As shown in Section 7.1, the only constant change it makes to the original page is a small banner along the top. F-Lingo takes advantage of the CSS style sheets employed by FutureLearn, ensuring that any additional lexical information (i.e. modal dialogs and summary wordlists) fit nicely into the design of the existing platform. This allows F-Lingo to work seamlessly with the original content, enhancing the learners' experience, rather than distracting them from the content that they are trying to learn.

Unlike other applications, F-Lingo has been designed specifically to work with existing online courses, taking advantage of their text for content-based language learning. Although it has the potential to be integrated into other online course platforms, it has been developed to work on top of FutureLearn, the online MOOC platform described in Sections 2.7, 3.3, and 3.4. The system has been named F-Lingo, where the "F" in F-Lingo acknowledges its integration into the FutureLearn platform, and "Lingo" represents its development for course-specific vocabulary. Just as FutureLearn can be seen as the future of online learning, the aim is for F-Lingo (Future-Lingo) to be considered the future of integrated content-based language learning. The next chapter outlines its implementation, while Chapter 8 evaluates its usability and functionality.

Chapter 7

Implementing F-Lingo

Lingo /ˈlɪŋ.gəʊ/

(noun) (1) language, especially language peculiar to a particular group, field, or region; jargon or a dialect.

(Wiktionary, 2019)

Content-based language learning involves non-language-based subjects, such as science, being learned in a second or foreign language, where language knowledge is essential for learning the subject. It is used by language teachers, integrating language into subjects such as engineering, mathematics, and geography. However, this is usually done in the classroom, rather than online. To my knowledge, no one has integrated content-based language learning to online courses.

This chapter describes the implementation of F-Lingo, a Chrome extension that highlights words, phrases, and concepts within online course content, and provides definitions, example sentences, related collocations, and additional information from Wikipedia. F-Lingo draws learners' attention to new and different vocabulary on the page, supporting content-based language learning by providing

additional lexical information to aid in vocabulary acquisition and reading comprehension.

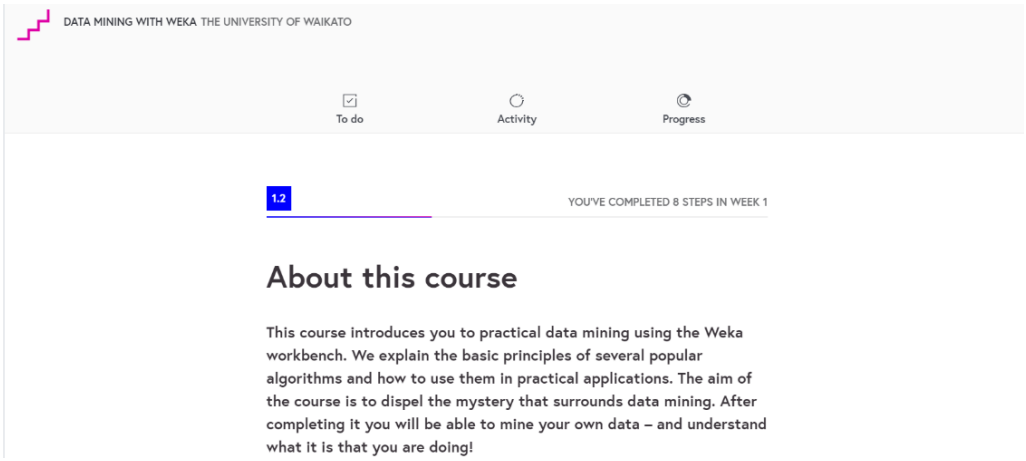
7.1 Introducing F-Lingo

F-Lingo is a Chrome extension that was developed for this thesis as an aid for second language acquisition. As mentioned in Section 6.6, although there is a plethora of existing computer assisted language software, F-Lingo is the first to explicitly take advantage of an online MOOC platform. It has been developed to work with FutureLearn, an online MOOC platform that provides learners with free online courses from leading universities, specialist organisations, and institutions around the world.

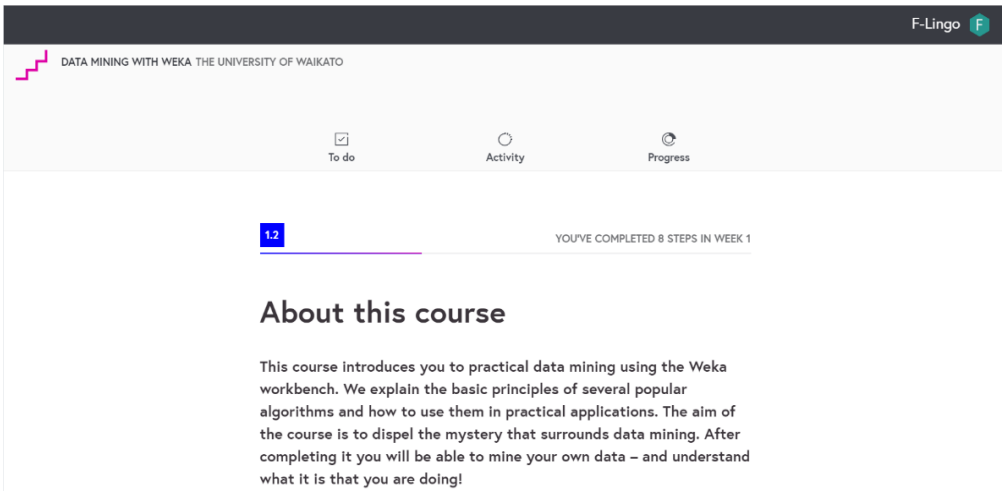
FutureLearn has hundreds of active courses, three of which are run by the University of Waikato: *Data Mining with Weka*, *More Data Mining with Weka*, and *Advanced Data Mining with Weka*. The first – Data Mining with Weka – is made up of 95 pages of textual content that total 25,000 running words, plus over 30,000 running words from video transcripts. Using F-Lingo on top of FutureLearn, learners can view text augmented with additional lexical features, taking advantage of online courses for content-based language learning.

F-Lingo supports the noticing hypothesis by highlighting words, phrases, and concepts in the text, and provides language resources. For words, it allows learners to view definitions and example sentences; for phrases, learners can view expanded phrases and example sentences; and for concepts, learners can view information about disambiguated single and multiword lexical items defined on Wikipedia. Finally, it provides learners with lists of lexical items highlighted in text as a summary of the key vocabulary on each page.

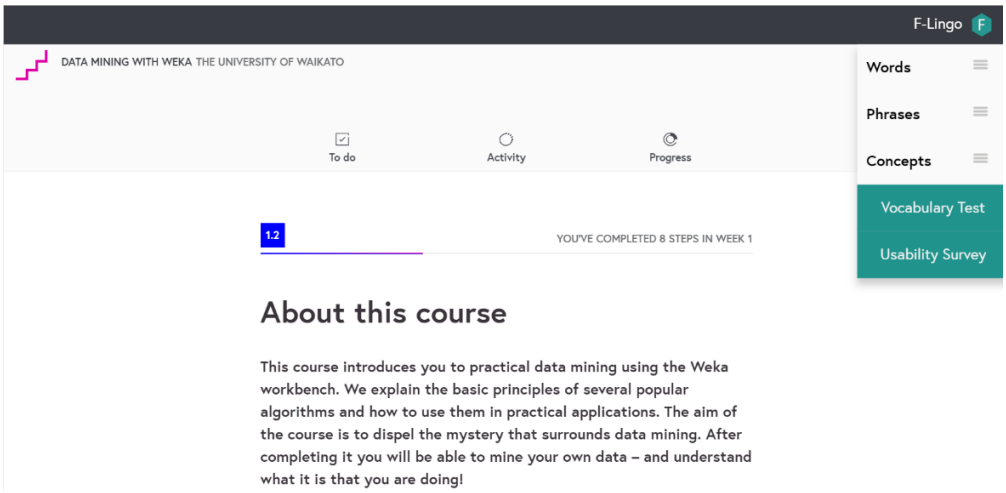
F-Lingo has been designed to be as unobtrusive as possible, fitting into an existing FutureLearn course without changing the layout or style. The only visual difference between a FutureLearn course with and without F-Lingo installed is a navigation bar that appears at the top of the page. Learners click on the navigation bar to open the F-Lingo menu, then select whether they want to highlight words, phrases, or concepts. Figure 7.1a shows a page from a FutureLearn course without F-Lingo installed, while Figures 7.1b and 7.1c show the same page after F-Lingo has been downloaded.



a) A FutureLearn page without F-Lingo installed



b) A FutureLearn page with F-Lingo installed



c) A FutureLearn page with F-Lingo installed and the drop-down menu open

Figure 7.1 F-Lingo: integrating F-Lingo into a FutureLearn page

In Figure 7.1b the black F-Lingo navigation bar is closed, while in Figure 7.1c it is open, showing the menu items learners can select to highlight words, phrases, and concepts.

7.2 Pre-processing

Before FutureLearn courses can be made available for use with F-Lingo, their content must be processed to extract sentences and identify words, phrases, and concepts. A Python script has been developed to perform the task, taking advantage of three external resources, the Natural Language Toolkit (NLTK), FLAX, and Wikipedia Miner. Pre-processing involves six steps: (1) downloading course content, (2) extracting sentences, (3) identifying words, (4) identifying phrases, (5) identifying concepts, and (6) caching the results in a database.

7.2.1 Downloading content

Section 3.5.1 discussed the development of a web-crawling Chrome extension that recursively follows links within the Data Mining with Weka courses, traversing content and extracting both the spoken and written text. There, it was used to gather course content, creating the *Data Mining with Weka* corpus. However, it can also be used to gather course content for F-Lingo's pre-processing. The text can then be passed through to the Python script, which extracts sentences and identifies words, phrases, and concepts.

7.2.2 Extracting sentences

One of the ways in which F-Lingo supports vocabulary acquisition is by providing learners with example sentences that demonstrate how words and phrases are used within the course content they are currently learning. In order to do this, F-Lingo first needs to split the content into individual sentences.

Identifying sentence boundaries can be challenging. Punctuation is often ambiguous. A period may denote a sentence boundary, but it may also denote a job title (*Dr.*), abbreviation (*etc.*), a decimal point (*5.2*), or an ellipse (*...*). Question marks and exclamation marks often appear within quotation marks (*she said "pardon?", but he didn't hear her.*), or inside parentheses. Sentence disambiguation is a natural language processing problem, and is accomplished in many natural language processing libraries that support it – such as the Natural

Code Example 7.1 NLTK: PunktSentenceTokenizer coding example

```
import nltk
para  = "Now you will learn how to use Weka's popular J48
         classifier, which builds decision trees. J48 is a
         reimplementation of a classic classifier algorithm
         called C4.5."
sents = nltk.sent_tokenize(para)
print(sents)
```

```
Output:
[   "Now you will learn how to use Weka's popular J48
classifier,
         which builds decision trees.",
    "J48 is a reimplementation of a classic classifier algorithm
         called C4.5."  ]
```

Language Toolkit (NLTK) (Bird & Tan, 2018), a leading platform for building Python programs that support natural language processing and text processing. It provides a suit of libraries for text classification, tokenization, stemming, and tagging.

F-Lingo uses the NLTK *PunktSentenceTokenizer* (Bird & Tan, 2018), which uses an unsupervised machine learning algorithm to build a model for sentences, using abbreviations, collocations, and sentence-initial words to find sentence boundaries. It takes a string and returns a list of sentences, as shown in Code Example 7.1. F-Lingo's pre-processing script assigns a sentence ID, relating to the course it was extracted from, and its position in the text. For the first course, *Data Mining with Weka*, this resulted in a total of 3303 sentences.

7.2.3 Identifying words

It is recommended that learners understand 95%–98% of the words on a page (Nation, 2001). However, some online learners may know less. Milton (2009) suggests that learners need to know at least the most frequent 2000 words in English before they can begin to comprehend more general ideas, and Nation (2001) stresses the importance of their acquisition for comprehension of vocabulary. We therefore assume that the online learners are familiar with at least the first 2000 English words, but do not assume any greater knowledge.

F-Lingo identifies words that are not in the most frequent 2000 English words, according to the General Service List (GSL) (West, 1953), which contains

Code Example 7.2 NLTK: TreebankWordTokenizer coding example

```
import nltk
para  = "Now you will learn how to use Weka's popular J48
         classifier, which builds decision trees."
sents = nltk.sent_tokenize(para)
words = nltk.tokenize.TreebankWordTokenizer().tokenize(sents[0])
print(words)
```

```
Output:
['Now', 'you', 'will', 'learn', 'how', 'to', 'use', 'Weka',
"'s", 'popular', 'J48', 'classifier', ',', 'which', 'builds',
'decision', 'trees', '.']
```

2000 headwords taken from a written corpus, representing words of the greatest 'general' use in English, described in Section 0.

Identifying words within course content begins with splitting text into sentences. The next step involves breaking sentences into individual words, referred to as *tokens*. This is achieved with the Natural Language Toolkit's *TreebankWordTokenizer* (Bird & Tan, 2018), which uses regular expressions to tokenize text, splitting sentences into words, punctuation, and contractions, as illustrated in Code Example 7.2. Once the content has been tokenized, word types, excluding punctuation, are extracted and stored in a list to prevent handling the same word more than once. The corresponding sentence ID for each occurrence of a type is also noted in the list. Next, each type is compared against the GSL, and discarded if it appears there. Finally, Wiktionary is used to determine whether the remaining types are words in the English language. A GET request is sent to Wiktionary, passing the type, and retrieving a JSON file. Two JSON attributes are used to determine whether there is an English definition for a type, the *language* and *part of speech*. Language is determined by an HTML ID tag, and part-of speech is determined by corresponding POS tags (noun, verb, etc.). Any types that do not return an English definition are discarded. The remaining types are identified as words.

7.2.4 Identifying phrases

F-Lingo identifies and provides additional lexical information for two types of phrases: collocations (described in Sections 2.1.2.1 and 6.4.2.1) and lexical bundles (described in Sections 2.1.2.2 and 6.4.2.2).

Table 7.1 Collocation patterns

Pattern	**Example**
Adjective + adjective + noun(s)	Extensive practical knowledge
Adjective + and/but + adjective + noun(s)	Present and future knowledge
Adjective + noun + noun	A practical data structure
Adjective + preposition + verb	Thorough in their learning
Adjective + to + verb	Conductive to learning
Adjective(s) + noun(s)	Practical knowledge
Adverb + adjective	Highly practical
Adverb + verb	Currently learning
Gerund verb + noun	The underlying data
Noun + noun	Data model
Noun + of + noun	Types of data
Noun + preposition + noun	Access to data
Noun + to + verb	Data to identify
Verb + adjective + noun(s)	Learning practical skills
Verb + adverb	Learning quickly
Verb + noun + noun	Learning data mining
Verb + noun(s)	Learning environment
Verb + preposition + noun(s)	Learning from experience
Verb + pronoun + adjective	Made it practical
Verb + to + verb	Learning to use

7.2.4.1 Collocations

To identify collocations, researchers often suggest taking a syntax-oriented approach (Firth, 1951; Nation, 2001; Nattinger & DeCarrico, 1992; Nesselhauf, 2004; Sinclair, 1991; Wu, Li, Witten, & Yu, 2016), emphasising a collocation's grammatical structure. By using this approach, collocations can be identified by their syntactic patterns. Wu et al. (2016) defines a set of twenty patterns shown in Table 7.1, which FLAX uses to identify collocations within a Wikipedia corpus.

F-Lingo uses FLAX to identify collocations. Given the course content as input, FLAX's *CollocationExtractor* returns a list of collocations that match the collocation patterns in Table 7.1. FLAX's identification process has three steps:

1) Split text into sentences

Table 7.2 A sample of sentence-initial lexical bundles

as a matter of	as can be seen
as a result of	as far as the
as a result, it	as is shown in
as a result, the	as one of the

2) Tag words with part of speech
3) Match the part of speech in the text with the collocation patterns

FLAX uses OpenNLP to extract sentences and add part of speech to text. Like the Natural Language Toolkit (NLTK), OpenNLP is a machine learning based library for natural language processing and text processing.

Although F-Lingo uses FLAX to identify collocations, it still needs to link them with corresponding sentence IDs. It uses Python's built in search capabilities to find collocations within course content, and generates a list of all collocations found, plus any corresponding sentence IDs. Collocations that only occur in one sentence are excluded.

7.2.4.2 Lexical bundles

F-Lingo identifies lexical bundles using a pre-defined list, recommended for use in academic writing by Li (2016). She developed a list of 160 sentence-initial bundles, taken from four postgraduate thesis corpora: the Chinese master's corpus, Chinese PhD corpus, New Zealand masters corpus, and New Zealand PhD corpus. Topic specific bundles and bundles containing chapter titles, method names, or proper names were excluded. Table 7.2 shows eight sentence-initial lexical bundles identified by (Li, 2016), ordered alphabetically, while Appendix E shows the full list.

F-Lingo uses Python's built in search capabilities to identify lexical bundles. It uses the pre-defined bundles to search for any that appear within course content, including sentence IDs for each. Lexical bundle identification has the potential to be expanded to include non-initial bundles, and to identify bundles without the use of a pre-defined list.

7.2.5 Identifying concepts

F-Lingo uses Wikipedia Miner to identify and disambiguate concepts within course

Code Example 7.3 Wikipedia Miner: annotation example

```
>> annotate 0.5 0 xxx

>> Now you will learn how to use Weka's popular J48 classifier,
which builds decision trees.
>> Xxx
Now you will learn how to use [Weka | Weka (machine learning)]'s
popular J48 [classifier], which builds decision trees.
```

Code Example 7.4 Wikipedia Miner: lead and outlinks example

```
>> a weka (machine learning)
12345
>> first 12345
Waikato Environment for Knowledge Analysis (Weka) is a suite of
[machine learning] software written in [Java | Java (programming
language)] …
>> outlinks 12345
12346/12347/12348/12349
>> p 12346
Machine Learning
```

content. Wikipedia Miner is an information retrieval toolkit that can be used to examine and obtain usable data from Wikipedia. It includes a database of summarised Wikipedia content, and an API to access it (Milne & Witten, 2013).

F-Lingo uses Wikipedia Miner to pre-process course content, identifying and disambiguating concepts within text, for example, depending on the context, *Weka*, could be identified as either *Weka (the bird)* or *Weka (machine Learning)*. Wikipedia miner identifies and disambiguates both single and multi-word items, for example, depending on the context *big data* could be identified and disambiguated as either *big data (large collections of data)* or *big data (the band)*. Wikipedia Miner performs a range of tasks, one of which is annotation. The annotation package includes:

1) Topic detection: detecting terms, both singular and multi-word phrases
2) Disambiguation: resolving ambiguous terms based on context
3) Link detection: determining the salience of topics
4) Document tagging: adding document mark-up to tag terms within text

F-Lingo's pre-processing Python script creates a Telnet connection that accesses Wikipedia Miner's annotation package. It passes the content of each course page through to Wikipedia Miner, which identifies and disambiguates concepts within text, and returns the page, including document mark-up. Code Example 7.3

illustrates using Wikipedia Miner's *annotate* command to identify and disambiguate concepts within text.

To provide additional Wikipedia content to learners, F-Lingo retrieves the first paragraph from a Wikipedia page, and a list of related concepts. Wikipedia Miner provides online APIs to achieve this. However, since the toolkit is no longer monitored regularly, the service is not always available. For this reason, F-Lingo's pre-processing script uses Wikipedia Miner to both identify and disambiguate concepts, and retrieve the additional information.

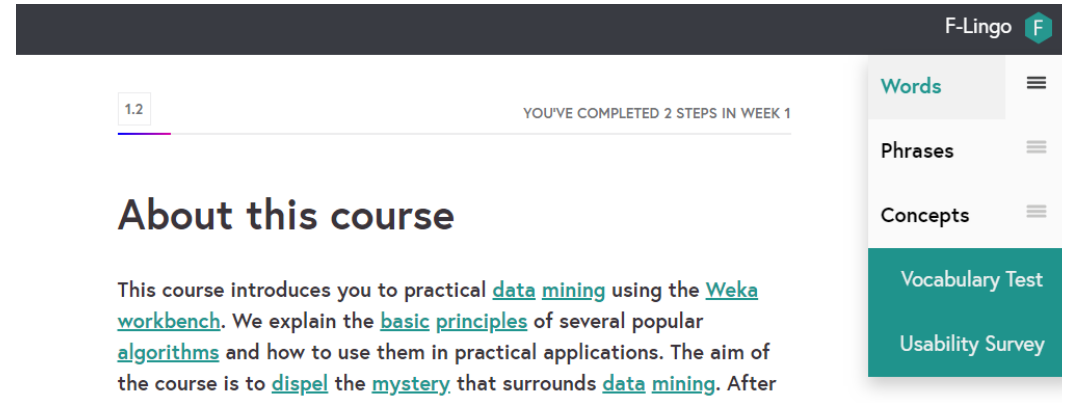
Code Example 7.4 demonstrates using Wikipedia Miner to retrieve the first paragraph from a Wikipedia page. Given the title of a concept, it first uses the article ("a") command to retrieve the concept's corresponding article ID. The article ID is then used, with the "first" command, to retrieve the first paragraph from the Wikipedia page. The next line uses the "outlinks" command to retrieve a list of IDs for articles that are included as links in the Wikipedia page. The page ("p") command is then used, with each related article ID, to retrieve the titles of related concepts.

7.2.6 Database caching

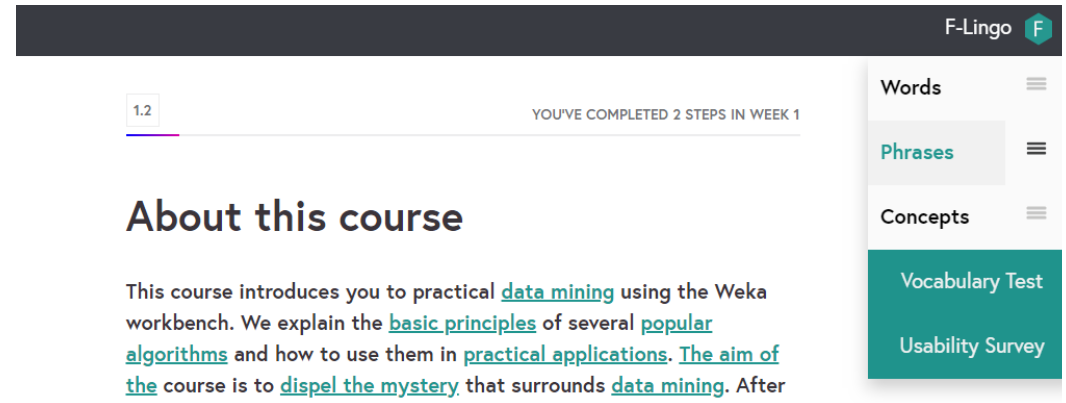
Once pre-processing has been completed, the corresponding data is cached in a database for later use by F-Lingo. The database holds:

1. Sentences, and their sentence IDs
2. Words, and the corresponding sentence IDs
3. Phrases, whether the phrase is a collocation or a lexical bundle, and the sentence IDs
4. The course content, by page, tagged with concepts
5. Concepts, the first paragraph of their corresponding Wikipedia page, and related concepts

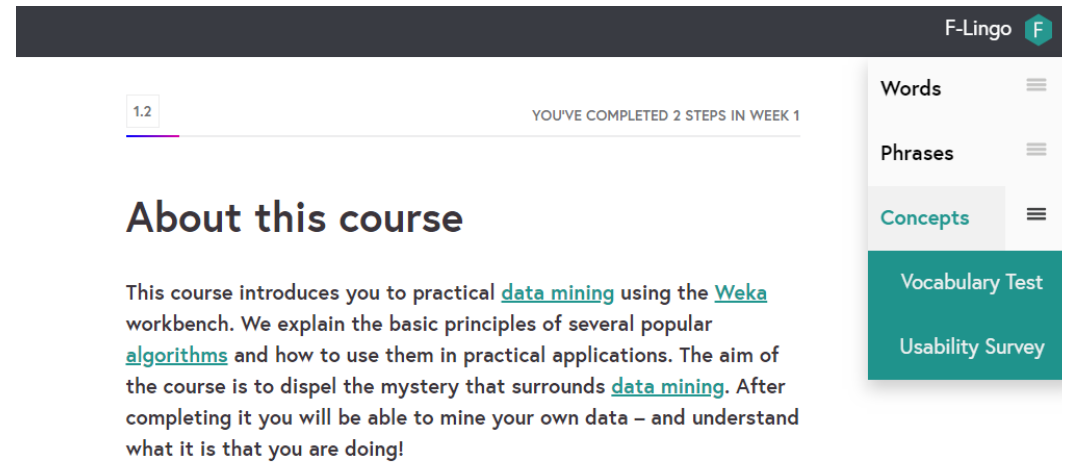
Section 7.3 shows how the data from points [2], [3] and [4] are used to highlight words, phrases, and concepts within FutureLearn, enhancing the content on a page, while Section 7.4 demonstrates using sentences [1] and Wikipedia information [5] to provide learners with additional lexical information, enhancing language acquisition.



a) Highlighting words



b) Highlighting phrases



c) Highlighting concepts

Figure 7.2 F-Lingo: highlighting words, phrases, and concepts

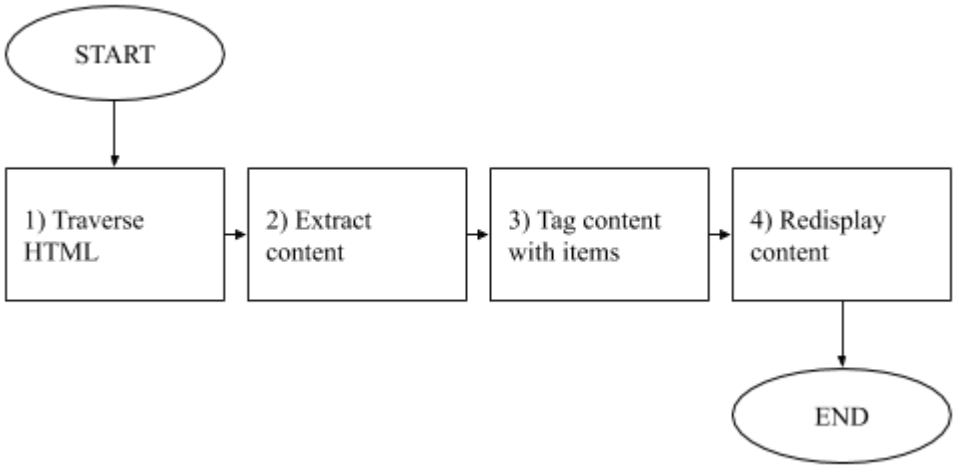


Figure 7.3 F-Lingo: traversing and highlighting content

7.3 Text enrichment

The F-Lingo Chrome extension works on top of FutureLearn to highlight words, phrases, and concepts within a course, as illustrated in Figures 7.2a, b, and c. Figure 7.3 outlines the process: first, F-Lingo traverses the HTML elements on a page and extracts textual content, then it tags the text with words, phrases, and concepts and re-displays the tagged content on the page. Learners can select whether they want to view the content tagged with words, phrases, or concepts, each of which results in a selection of lexical items being highlighted and made into clickable links that provide additional lexical information.

7.3.1 Traversing MOOC content

Not all content on a FutureLearn page can be tagged with lexical items. Some elements – such as buttons, internal links, or external links – are already clickable, and cannot be tagged to re-display clickable links without altering the original content on the page.

Figure 7.4 shows an excerpt from the *Data Mining with Weka* course on FutureLearn in which the term *Weka Software* (highlighted pink) links to an external site for downloading and installing Weka. Although the term *Weka* could be tagged as both a word and a concept, doing so here would disturb the original content on the page. F-Lingo works on top of FutureLearn, enhancing its content without disturbing it.

Figure 7.4 Content from the FutureLearn *Data Mining with Weka* course.

F-Lingo traverses the HTML on a page, extracting only text nodes that are not linked to clickable elements, for example, in Figure 7.4, F-Lingo would extract the content before the highlighted link: *You will download and install the free* and the content after the highlighted link: *during Week 1. It runs on … around the world* but not the link itself. F-Lingo tags each section of content separately, ensuring that, when re-displayed, it occupies its original position in the document.

7.3.2 Highlighting words

Section 7.2 explained how words were identified in text and cached in a database. When F-Lingo extracts content from a page, it sends it to the F-Lingo server, which uses the pre-identified list from the database to add mark-up to any words that occur in the text. Code Example 7.5 illustrates mark-up for the word *Weka*, while Figure 7.2a shows words highlighted on FutureLearn.

7.3.3 Highlighting phrases

F-Lingo provides learners with the ability to highlight phrases within text. Although it is not specified to learners, this results in two specific types of phrases being highlighted: collocations and lexical bundles. Identifying collocations and lexical bundles, and caching them in the database, is discussed in Section 7.2.

F-Lingo sends the textual content to its server, which then uses the pre-identified list of lexical bundles and collocations from the database, marking up any that occur in the text. Code Example 7.6 illustrates mark-up for the collocation *decision trees*, Code Example 7.7 illustrates mark-up for the sentence-initial lexical bundle *In the case of*, and Figure 7.2b shows collocations and lexical bundles highlighted on FutureLearn.

Code Example 7.5 F-Lingo: mark-up for words

```
<span>
      <a class="f-highlight"
      onclick="onclickWord(this)">Weka</a>
</span>
```

Code Example 7.6 F-Lingo: mark-up for collocations

```
<span>
      <a class="f-highlight" onclick="onclickPhrase(this,
      collo)">decision trees</a>
</span>
```

Code Example 7.7 F-Lingo: mark-up for lexical bundles

```
<span>
      <a class="f-highlight" onclick="onclickPhrase(this,
      bund)">In the case of</a>
</span>
```

Code Example 7.8 F-Lingo: mark-up for Wikipedia concepts

```
<span>
      <a class="f-highlight" onclick="onclickWiki('1579244',
      'statistical classification',
      'classifier')">classifier</a>
</span>
```

7.3.4 Highlighting concepts

Both words and phrases are highlighted using a list of pre-processed items from the database. However, the same cannot be done for concepts. Section 7.2.5 showed how Wikipedia Miner is used to identify and disambiguate concepts within text. If F-Lingo highlighted concepts based on the pre-processed list, rather than the context that surrounds them, this disambiguation would be lost. Section 7.2.6 showed that full pages tagged with Wikipedia concepts were cached in the database. They are used to add HTML mark-up to course content, highlighting concepts within text.

Wikipedia Miner tags concepts in two ways. If the name of the Wikipedia page matches the words in the text, i.e. there is no ambiguity, the concept is simply surrounded by double square brackets. For example, for the concept *data mining* the tag is as follows.

Code Example 7.9 F-Lingo: matching plain text with Wikipedia tags

Plain text

```
J48 is a reimplementation of a classic classifier algorithm
called C4.5.
```

Regular expression

```
(\[\[)*J48.*?(\]\])* (\[\[)*is.*?(\]\])* (\[\[)*a.*?(\]\])*
(\[\[)*reimplementation.*?(\]\])* (\[\[)*of.*?(\]\])*
(\[\[)*a.*?(\]\])* (\[\[)*classic.*?(\]\])*
(\[\[)*classifier.*?(\]\])* (\[\[)*algorithm.*?(\]\])*
(\[\[)*called.*?(\]\])* (\[\[)*C4\.5.*?(\]\])*
```

Text tagged with Wikipedia concepts

```
J48 is a reimplementation of a classic [[classifier |
statistical classification]] [[algorithm]] called [[C4.5 | C4.5
algorithm]].
```

$$[[\ data\ mining\ ]]$$

Otherwise, if a concept can have multiple meanings, and Wikipedia Miner has disambiguated it, the concept is returned with its disambiguated term, surrounded in double square brackets, for example, for *Weka (machine learning)*, the tag is as follows.

$$[[\ Weka\ |\ Weka\ (machine\ learning)\ ]]$$

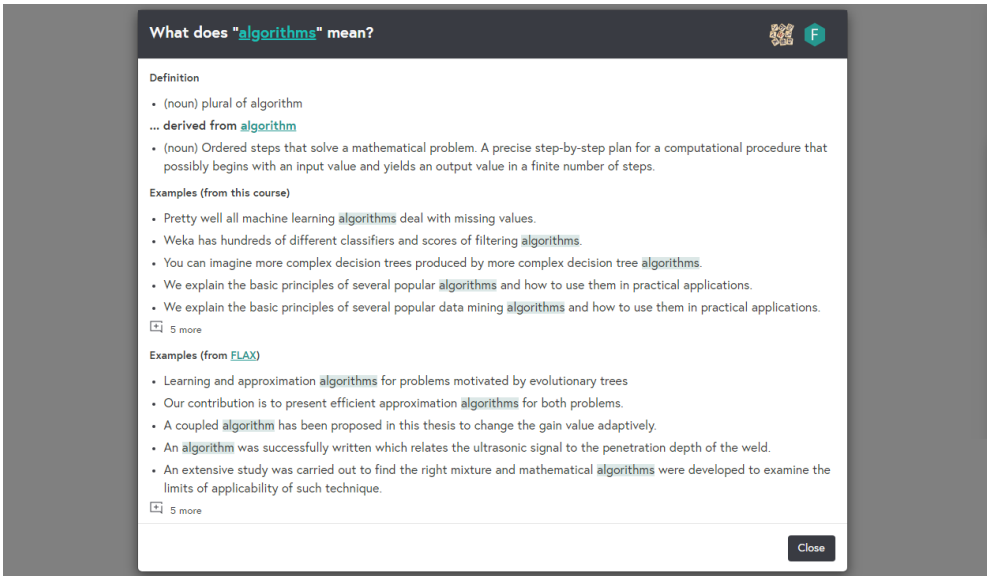
F-Lingo uses regular expressions to match the plain content extracted from a FutureLearn page with a portion of the tagged Wikipedia content. It breaks the plain content into tokens and surrounds each one with the following regular expression.

$$(\backslash[\backslash])*<token>.*?(\backslash]\backslash])*$$

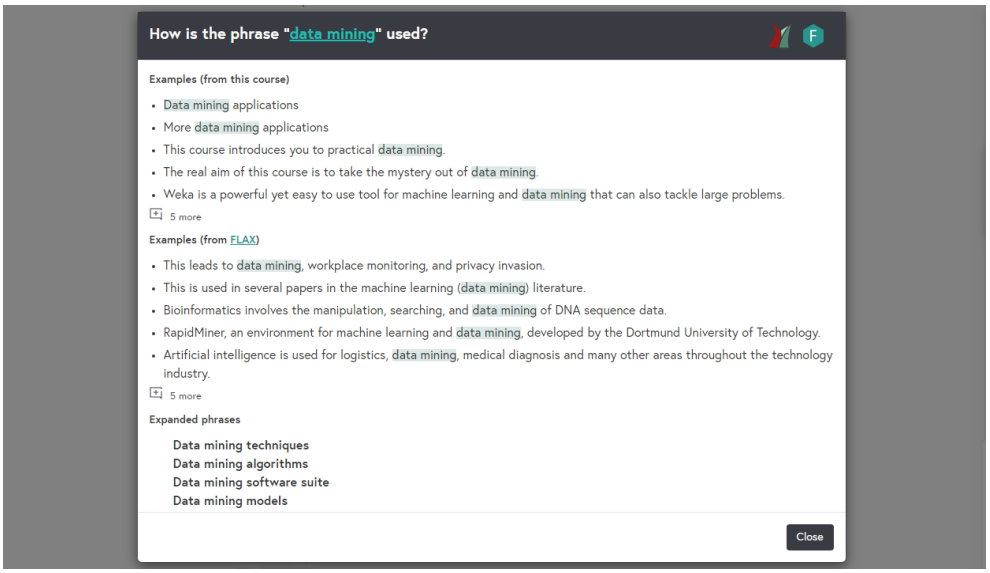
Code Example 7.9 illustrates formatting plain text into a regular expression that matches text tagged with Wikipedia concepts. Once F-Lingo has found the tagged Wikipedia content that matches the plain text extracted from a FutureLearn page, it can replace the Wikipedia tags with HTML mark-up. Code Example 7.8 illustrates mark-up for the concept *classifier*, while Figure 7.2c shows concepts highlighted on FutureLearn.

7.4 Language resources

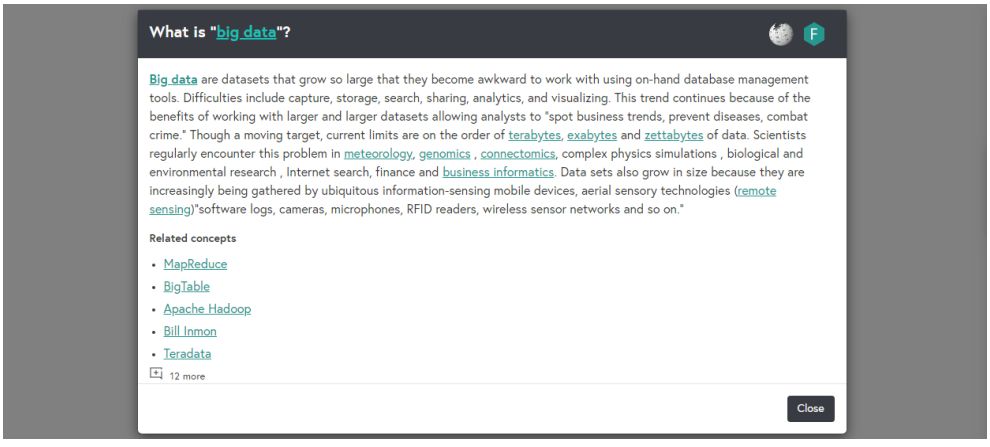
Once words, phrases, and concepts have been highlighted, they can be clicked to view additional lexical information, as illustrated in Figures 7.5a, b and c.



a) Modal dialog for words



b) Modal dialog for phrases



c) Modal dialog for concepts

Figure 7.5 F-Lingo: dialogs for words, phrases, and concepts

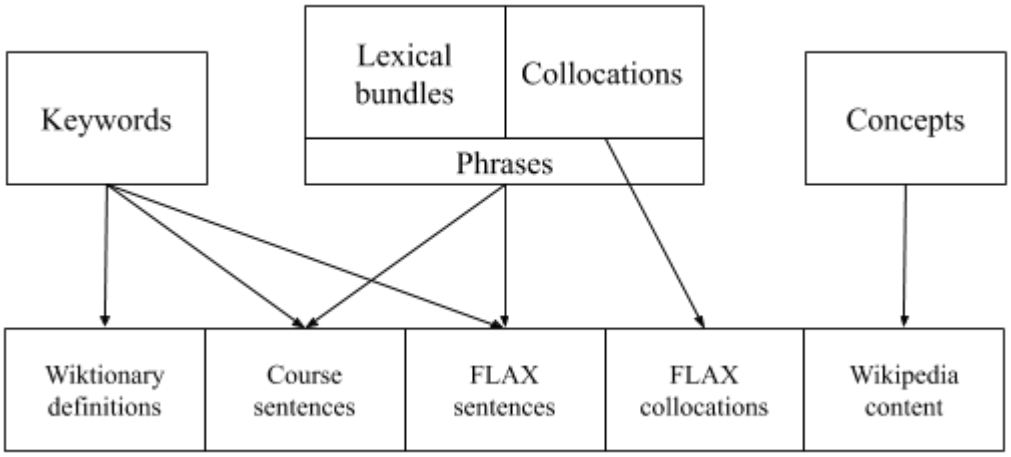


Figure 7.6 F-Lingo: language resources

Figure 7.6 illustrates the language resources available. F-Lingo provides Wiktionary definitions for words; sample sentences from within the course and from FLAX, for words, collocations, and lexical bundles; expanded collocations (discussed in Section 7.4.3) for collocations; and content from Wikipedia for disambiguated concepts.

7.4.1 Wiktionary definitions

F-Lingo retrieves dictionary definitions from Wiktionary. It sends a GET request, passing it the word, and retrieving a JSON file that contains the HTML for a page. F-Lingo also determines whether the word is derived from another, by searching the HTML for the *form-of-definition* tag. If it exists, a second GET request is sent, passing the derived word, and retrieving its JSON file, extracting the HTML for the page. F-Lingo uses this to extract each part-of-speech and dictionary definition. Definitions that are marked as obsolete, archaic, vulgar, or slang are excluded.

Once a definition has been retrieved from Wiktionary and processed, it is re-formatted with HTML tags and returned to the F-Lingo Chrome extension, where it is displayed in a modal dialog. If the word was derived from another, F-Lingo retrieves both definitions and display them in the dialog. Figure 7.5a shows the definition for the word *algorithm* and its derived form, *algorithms*, retrieved from Wiktionary and displayed by F-Lingo.

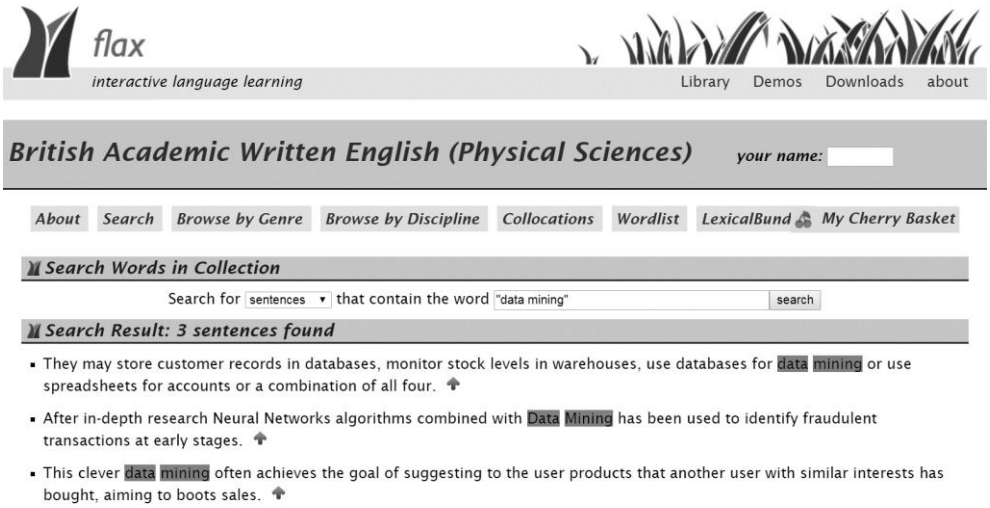


Figure 7.7 FLAX: search example relating to *data mining*

7.4.2 Example sentences

F-Lingo provides learners with example sentences, demonstrating how words and phrases are used within context. It provides learners with sentences from within the course, and from an external source (FLAX).

7.4.2.1 Course sentences

F-Lingo provides learners with example sentences for words and phrases (collocations and lexical bundles) from the course they are studying. This allows learners to see how single and multi-word lexical items can be used within their current context.

When learners click on words or phrases, F-Lingo uses a database selection query to retrieve sentence IDs that relate to the item. It then uses the sentence IDs to retrieve sentence content. F-Lingo orders the returned sentences by their level of simplicity. Sentences with less punctuation are displayed first, while sentences with more punctuation, numeric figures, and acronyms are displayed last.

F-Lingo returns the top ten sentences, displaying five and providing the option to expand the rest. Figure 7.5a shows example sentences, from the *Data Mining with Weka* course, for the word *algorithms*, while Figure 7.5b shows example sentences for the collocation *data mining.*

7.4.2.2 FLAX sentences

As well as providing sentences from within the course, F-Lingo also provides

learners with example sentences from an external source, FLAX. This allows them to see how words and phrases are used within the course and outside of it, and any differences that may appear between the two.

FLAX is an online tool that includes collections of digital text from a variety of sources including the British Library's Open Access toolkit, which contains a selection of PhD abstracts from four categories; Arts and Humanities, Life Sciences, Physical Sciences, and Social Sciences. FLAX allows learners to search its content for words and phrases, returning example sentences that contain them.

Figure 7.7 demonstrates using FLAX to search for the phrase *data mining*. F-Lingo uses FLAX's word search utility to retrieve sentences from the Physical Sciences collection that contain a given word or phrase. It sends a GET request, passing the word or phrase, and retrieves an XML file containing the relevant data. F-Lingo extracts example sentences from the XML file and orders them by form simplicity, as was done for the course sentences. It then displays five of the top ten sentences and provides the option to expand the rest. Figure 7.5a shows example sentences, from FLAX, for the word *algorithms*, while Figure 7.5b shows FLAX sentences for the collocation *data mining.*

7.4.3 Expanded Collocations

FLAX provides extensive information about collocations. The *FLAX Learning Collocation* collection displays collocations from Wikipedia, organized by syntactic patterns and frequency, and linked back to the original text (Wu et al., 2016). Figure 7.7 shows an example for the collocation *data mining*.

F-Lingo uses FLAX to retrieve collocations. It sends a GET request, specifying a target collocation, and receives an XML document with any collocations that relate to it. Some are expansions of the original collocation: for example, data mining algorithms and data mining models are collocations that expand the target collocation data mining. However, others only contain one word from the target collocation, for example, data structures and data types are collocations that contain the word data from the target collocation data mining. F-Lingo displays expanded collocations, so collocations that only contain one word from the target (i.e. data structures and data types) are discarded. Figure 7.5b shows expanded collocations for the target collocation data mining.

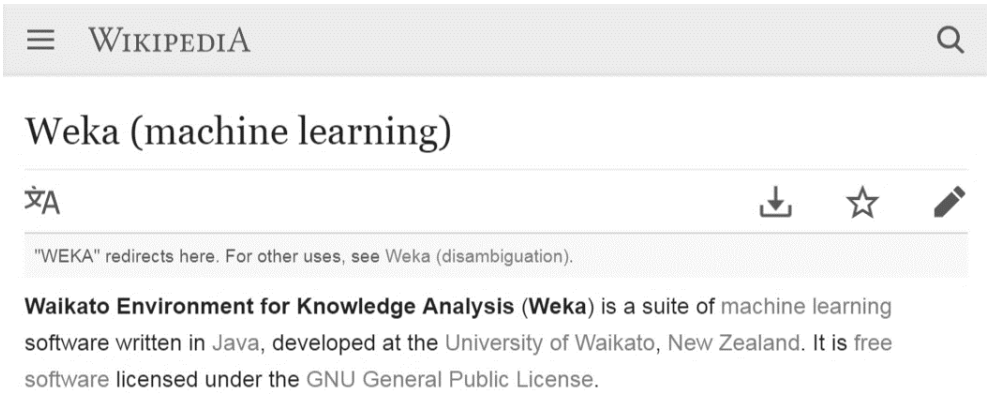


Figure 7.8 Wikipedia: the article for the concept *Weka*

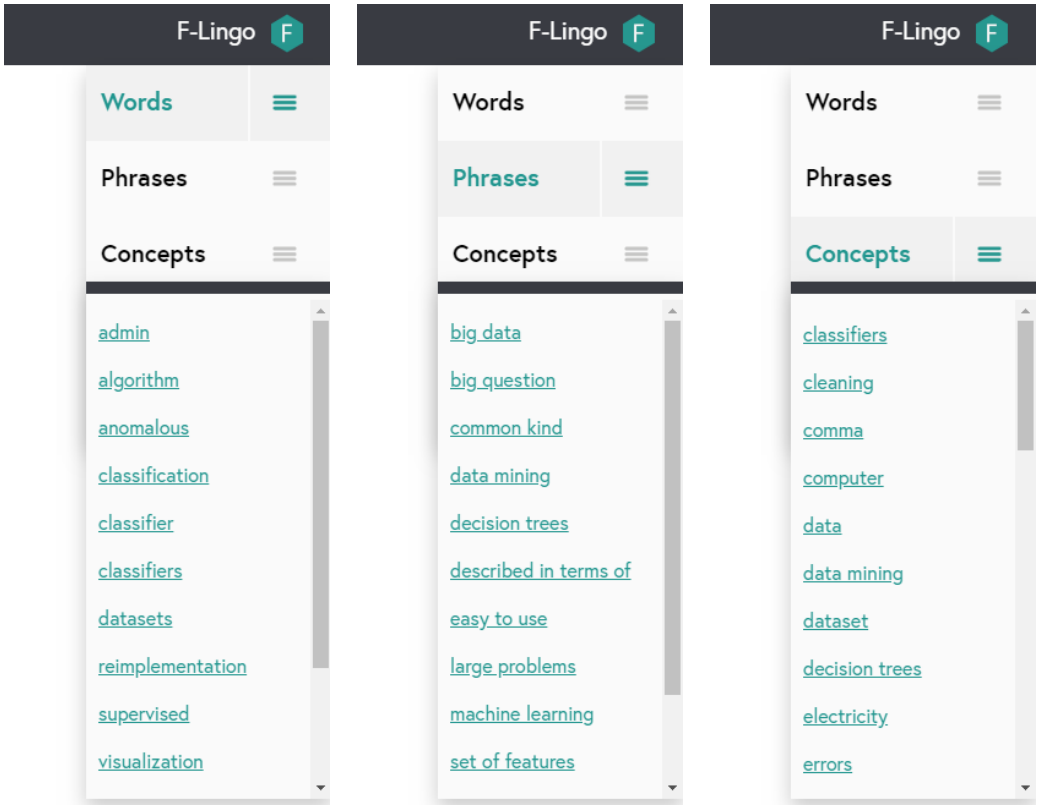
7.4.4 Wikipedia content

Wikipedia concepts are mapped to their corresponding Wikipedia page so that F-Lingo can provide learners with the first paragraph from the page, referred to as the *lead*, and titles of any Wikipedia articles that relate, referred to as *outlinks*. The lead section of a Wikipedia page is the introduction. It summarises the most important information in the article (Wikipedia, 2019) and is found at the top of the page, before the logo, table of contents, and article body. Outlinks are Wikipedia articles that are shown as links within a Wikipedia page. Figure 7.8 shows the lead text of the Wikipedia article for the machine learning concept *Weka*, and the outlinks to other articles, for example *machine learning*, *Java*, *University of Waikato*, *New Zealand*, *free software*, and *GNU General Public License*.

F-Lingo uses Wikipedia Miner, during pre-processing, to retrieve the Wikipedia article ID, the first paragraph (lead), and any related articles (outlinks), described in Section 7.2.5. When learners click on concepts, F-Lingo uses the concept title to access the Wikipedia lead and outlinks through a database selection query. It then formats it, adding HTML tags to show the lead text and the first five related articles. Figure 7.5c shows Wikipedia content for the concept *Weka*.

7.5 Content-specific wordlists

F-Lingo provides leaners with a list of highlighted words, phrases, and concepts per page. Each time a page is loaded, F-Lingo communicates with the server, tagging the content that it traverses. At the same time, it returns a list of the items that have been tagged. Learners can then select the list icon (≡) next to the *words*, *phrases*,



a) Words b) Phrases c) Concepts

Figure 7.9 F-Lingo: lists of words, phrases, and concepts

and *concepts* menu items to view a list of the items that have been highlighted on the page. Figures 7.90a, 7.90b and 7.90c show an example of each.

7.6 Vocabulary testing

Finally, F-Lingo provides learners with domain-specific vocabulary tests, based on those described in Section 5.7. This is currently tailored specifically for the *Data Mining with Weka* courses but could easily be expanded for other courses on FutureLearn. The *DMwW wordlist* has been used to generate 100 domain-specific pseudowords, using 3-grams and a word similarity metric no greater than 0.85, as described in Section 5.3.3. The option to complete a vocabulary test has been included in F-Lingo's menu items. Each time learners click on it, a random set of 40 real words are selected from the DMwW wordlist, and a random set of 20 pseudowords are selected from those that were generated. This allows learners to take the vocabulary test multiple times with a different selection of words each time. Figure 7.10 illustrates the test.

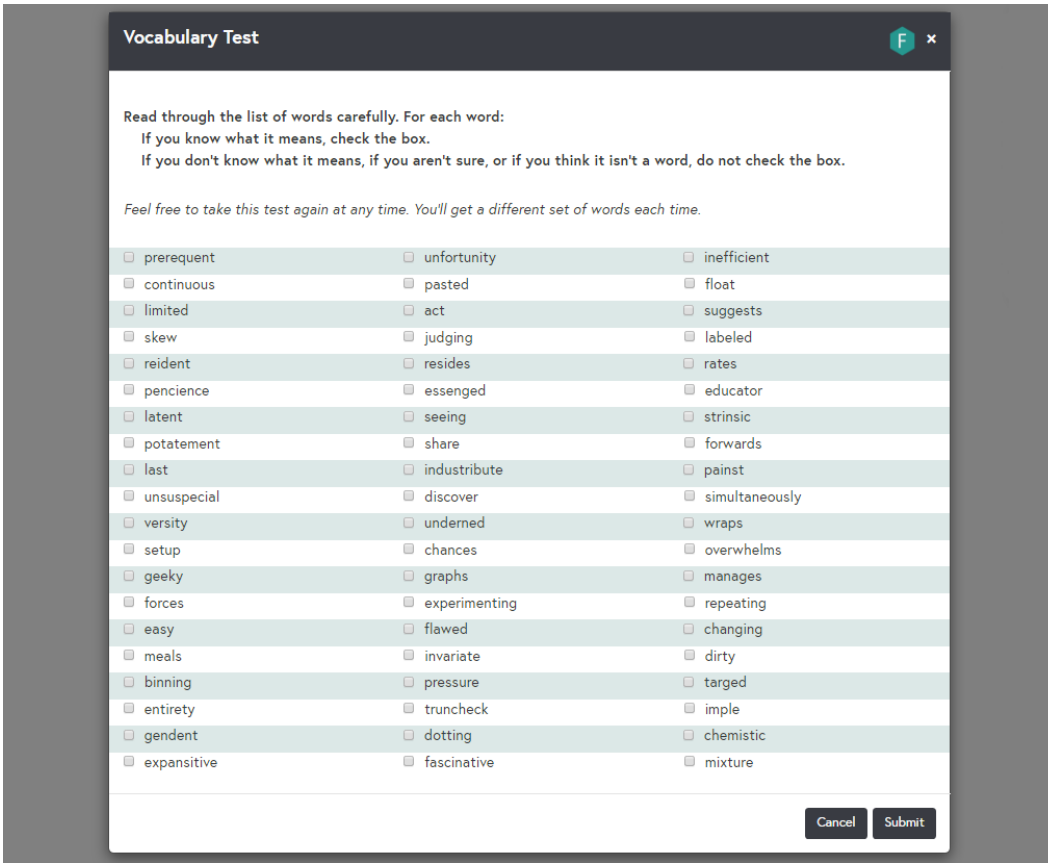


Figure 7.10 F-Lingo: vocabulary testing

7.7 Integrating F-Lingo into courses

The F-Lingo Chrome extension is currently available for three FutureLearn courses: *Data Mining with Weka*, *More Data Mining with Weka*, and *Advanced Data Mining with Weka*. However, it can be applied to any FutureLearn course. This involves four steps:

1. Download course content
2. Pre-process course content
3. Cache data in the F-Lingo database
4. Add the course title to the database

F-Lingo highlights words, phrases, and concepts each time a page is loaded and provides additional lexical information each time an item is clicked. However, providing these facilities first requires some pre-processing. For example, providing learners with example sentences from within the course, particularly from pages in the course that they have not yet visited, cannot be achieved without first processing the course content to extract sentences. This means that, before F-Lingo can be used

on a course, it needs to have access to the content in its entirety. This can either be achieved using the *CourseCrawler* described in Section 3.5.1, or by an educator providing the course content themselves.

Once the content is available, F-Lingo's pre-processing can begin, splitting the course into sentences and indexing them, identifying words, phrases (using FLAX), and concepts within the text, and retrieving concept leads and related articles using Wikipedia Miner. This results in a set of text files that contain SQL statements that can be loaded into the F-Lingo database. Finally, F-Lingo checks the database for active courses before it integrates itself into a FutureLearn course. Adding a courses title to the database allows it to be recognized by F-Lingo.

Chapter 8

Evaluating F-Lingo

Evaluate /ɨˈvaljʊeɪt/

(verb) (1) to draw conclusions from examining, to assess (2) to compute or determine the value of.

(Wiktionary, 2019)

This chapter evaluates the F-Lingo Chrome extension in two ways: an expert heuristic evaluation, and a learner-data based evaluation. The first is used to determine whether F-Lingo meets design and usability standards and integrates language resources into online courses without disturbing the original content, using Nielsen's (1994) ten heuristics as the evaluation criteria. The second is used to interpret learner behaviour in relation to their use of the F-Lingo Chrome extension. Learners' interaction with words, phrases, and concepts is tracked, both from within a page's content, and from the summary lists, to draw inferences about learner behaviour. The learner-data evaluation can be broken into four parts: (1) a participant-based analysis, which summarises the distribution of participant statistics, (2) an interaction-based analysis, which focuses on the type of interactions that occurred (clicking words, phrases, or concepts; clicking in the text

or in summary lists), (3) a time-based analysis, which investigates the duration of time learners spent reading the language resources, and (4) a content-based analysis, which focuses more closely on the actual words, phrases, and concepts that were clicked.

8.1 Expert heuristic evaluation

The F-Lingo Chrome extension was subjected to an expert heuristic evaluation, conducted by a professor from the University of Waikato. The aim of the evaluation was to highlight any shortfalls in F-Lingo's design with a view to overcoming them and determining whether F-Lingo integrated language resources into online courses without disturbing the original content.

8.1.1 Methodology

The expert heuristic evaluation was conducted in the form of a cognitive walkthrough. The walk through started with navigating to the Chrome Web Store and installing F-Lingo, then navigating to the *Data Mining with Weka* course on FutureLearn in order to test each of F-Lingo's features.

The results of the evaluation have been structured around a set of ten heuristics, developed by Nielsen (1994), that are used in testing the design and usability of a user interface. They are as follows.

1. *Visibility of the system status:* feedback should be provided to the user in order to keep them informed on the status of the system.
2. *Match between the system and the real world:* the system should use language that is familiar to the user, and information should appear in a natural and logical order.
3. *User control and freedom:* the system should provide users with an 'emergency exit' for when they enter any states accidentally.
4. *Consistency and standards:* the system should not use different words, situations or actions to mean the same thing.
5. *Error prevention:* prevent errors from occurring where possible. Provide users with a confirmation option before entering any situation where an error may occur.

6. *Recognition rather than recall:* make options, actions and values visible to the user. They should not be expected to remember information from one part of the system to another.
7. *Flexibility and efficiency of use:* accelerators and shortcuts should be provided for the advanced user.
8. *Aesthetic and minimalist design:* the system should not include information that is irrelevant. Any additional unnecessary information will take away from the importance of the necessary information.
9. *Help users recognize, diagnose, and recover from errors:* provide the user with error messages in plain language indicating the problem and suggesting a solution.
10. *Help and documentation:* the user should be able to understand the system without help or documentation. However, if it is necessary, ensure that it is clear, concise and easy to use.

8.1.2 Results

Shortfalls in the design and functionality of F-Lingo have been recorded here, according to the expert heuristic evaluation, ordered by Nielson's ten heuristics.

8.1.2.1 Heuristic 1: Visibility of the system status

Two shortfalls were identified relating to visibility. First, by default, when a Chrome extension is installed, a message pops up instructing users to click the icon in the browser search bar to start the extension. Contrary to this message, F-Lingo is not started by clicking on this icon. Instead, it becomes available once the user refreshes the page. However, no information was provided to suggest this. Second, when a learner clicks on a word, phrase, or concept, a dialog opens with additional lexical information. If the item has been clicked before, the information has been cached and retrieval is fast. However, if not, F-Lingo retrieves information from external sources, such as Wiktionary for definitions and FLAX for example sentences. While this content is retrieved, a dialog provides users with a loading symbol. However, this symbol resembles the infinity sign and may communicate that the loading process will never end.

8.1.2.2 Heuristic 2: Match between system and the real world

Two shortfalls were identified here, both of which had to do with tooltip wording. F-Lingo provides tooltips when learners hover over different features within the system. The tooltips for words, phrases, and concepts each start with "Show selected", for example: "show selected words", "show selected phrases", or "show selected concepts". First, "show" suggests clicking on the menu item will make words, phrases, or concepts appear on the page, as if they are not currently visible. This is not the case; rather, relative items become highlighted on the page. Second, "selected" suggests learners should physically select the text on the page, when it should be referring to the fact that not all words on the page are highlighted.

8.1.2.3 Heuristic 3: User control and freedom

One shortfall was identified where users do not have full control over the system. F-Lingo is controlled through its menu items. Clicking the black F-Lingo bar once opens the drop-down menu. Clicking it a second time closes it and removes highlighting from the page. If learners want to uninstall F-Lingo, they can right click on the F-Lingo icon in the browser search bar and select "Remove from Chrome", which is Chrome's default behaviour. However, once users click one of *Words, Phrases,* or *Concepts* in the drop-down menu, they cannot click it a second time to disable it. Instead, they either have to click on another list item, highlighting a different lexical item, or click on the black bar to close F-Lingo completely.

8.1.2.4 Heuristic 4: Consistency and standards

Two shortfalls were identified in terms of consistency. First, F-Lingo refers to words, phrases, and concepts consistently throughout the interface. However, one occurrence was identified, in a tooltip, where the word "term" was used instead of "concept". Second, F-Lingo contains several links to three external resources: Wiktionary for definitions, FLAX for example sentences, and Wikipedia for related concepts. Wiktionary and Wikipedia both use secure https, while FLAX uses http.

8.1.2.5 Heuristic 5: Error prevention

No shortfalls were identified in relation to error prevention. There are two situations where error prevention is required. However, F-Lingo has been developed to handle them. First, external links that could cause errors are opened in new tabs, preventing them from affecting F-Lingo or FutureLearn. Second, if an error occurs when

retrieving external content (e.g. from Wiktionary), F-Lingo tracks this and displays a dialog window informing users that the relevant information was unavailable.

8.1.2.6 Heuristic 6: Recognition rather than recall

The expert heuristic evaluation identified two shortfalls in recognition rather than recall, both of which were in reference to the external resource FLAX. First, F-Lingo uses FLAX to retrieve expanded phrases and example sentences. However, with the exception of the FLAX logo, there is no reference to it, either in the tooltips or the F-Lingo interface. Users should not be expected to recognize the FLAX logo without further textual reinforcement. Second, where example sentences were displayed, F-Lingo included the text “from here”, which when clicked, redirected the user to FLAX. However, again, there was no textual content to make this explicitly clear.

8.1.2.7 Heuristic 7: Flexibility and efficiency of use

This heuristic suggests that a system should include accelerators or shortcuts for advanced users. F-Lingo does not support either. However, it could be considered for future development.

8.1.2.8 Heuristic 8: Aesthetic and minimalist design

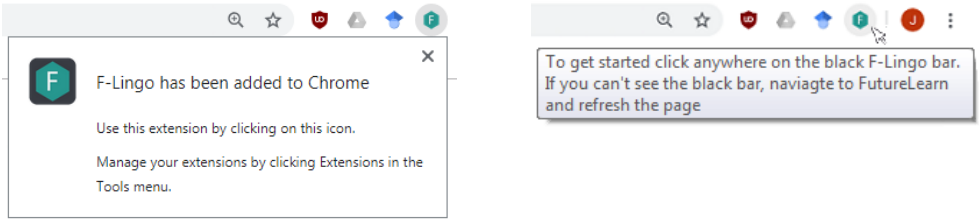
This heuristic suggests that the system should not include information that is irrelevant. Although shortfalls were identified in terms of inconsistent language (consistency and standards), and ambiguous language (match between system and the real world), no shortfalls were identified in terms of irrelevant information. Rectifying the shortfalls in inconsistent and ambiguous language will further cement the aesthetic and minimalist design of F-Lingo.

8.1.2.9 Heuristic 9: Help users recognize, diagnose, and recover from errors

No shortfalls were identified when helping users recognize, diagnose and recover from errors. F-Lingo provides plain language error messages indicating the problem and suggesting a solution. For example, if an error occurs when retrieving information from the database or an external resource, it displays a dialog window informing users of the error and suggesting that they try again later.

8.1.2.10 Heuristic 10: Help and documentation

One shortfall was identified in terms of help and documentation. Words, phrases,



a) Chrome's pop-up b) F-Lingo's pop-up

Figure 8.1 F-Lingo: installing the Chrome extension

a) Original loading symbol b) New loading symbol

Figure 8.2 F-Lingo: loading symbols

and concepts that have been highlighted on the page do not provide any help information. The results of the expert heuristic evaluation suggested adding tooltips to each highlighted item, informing the user, when they hover over them, that clicking them retrieves additional lexical information. However, this has the potential to conflict with the eighth heuristic, relating to aesthetic and minimalist design. Any other shortfalls for help and documentation have been identified in previous sections, in terms of inconsistent and ambiguous language.

8.1.3 Discussion

Five changes have been made to F-Lingo based on the results of this evaluation.

1. The first heuristic showed a shortfall in how users were informed to start F-Lingo. Chrome's default pop-up gives users incorrect information but its content cannot be changed and it cannot be disabled (Figure 8.1a). However, a second popup has been added which becomes visible when users hover over the F-Lingo icon. This pop-up provides users with the correct instructions for starting F-Lingo (Figure 8.1b).
2. The first heuristic also showed a shortfall in the design of F-Lingo's loading

symbol (Figure 8.2a). This has since been changed to reduce confusion (Figure 8.2b).

3. The second and fourth heuristics highlighted inconsistencies and ambiguities in some of the wording used by F-Lingo. The word “show” has been changed to “highlight”, and “selected” has been changed to “key”, for example, “highlight key words” rather than “show selected words”. The word “term” has also been changed to “concept”.
4. The third heuristic identified a shortfall in drop-down menu behaviour. This has since been rectified. Now, the first time a menu item is clicked, it is selected, and the second time it is clicked, it is deselected, returning control to the user and allowing them to have the drop-down menu open but nothing highlighted on the page.
5. The sixth heuristic identified a lack of recognition for FLAX, one of the external resources used by F-Lingo. FLAX is now mentioned in the tooltips for words and phrases, and “from here” has been changed to “from FLAX”.

Finally, two shortfalls were identified but not rectified.

1. The fourth heuristic identified an inconsistency in external links. Two external sources use https, while the third uses http. The preferred solution would be to use https with all three sources. However, the third source does not support it. The other alternative would be to use http for all sources. However, this would result in a less secure system. Instead, no changes have been made, but could be in the future, should circumstances change.
2. The tenth heuristic identified a shortfall in help documentation for words, phrases, and concepts highlighted on the page, and suggested that tooltips could be added that inform users when they hover over them. However, as mentioned above, this has the potential to conflict with the eighth heuristic, relating to aesthetic and minimalist design, so no change has been made.

Although two shortfalls could not be rectified, the first relates more to security than usability, and the second conflicts with the minimalist design of F-Lingo. Rectifying the other five shortfalls has resulted in a design that meets all ten heuristics, including the eighth heuristic *aesthetic and minimalist design.* This suggests that F-Lingo’s interface meets design and usability standards, and in turn, due to its design, does not interfere with FutureLearn’s original content.

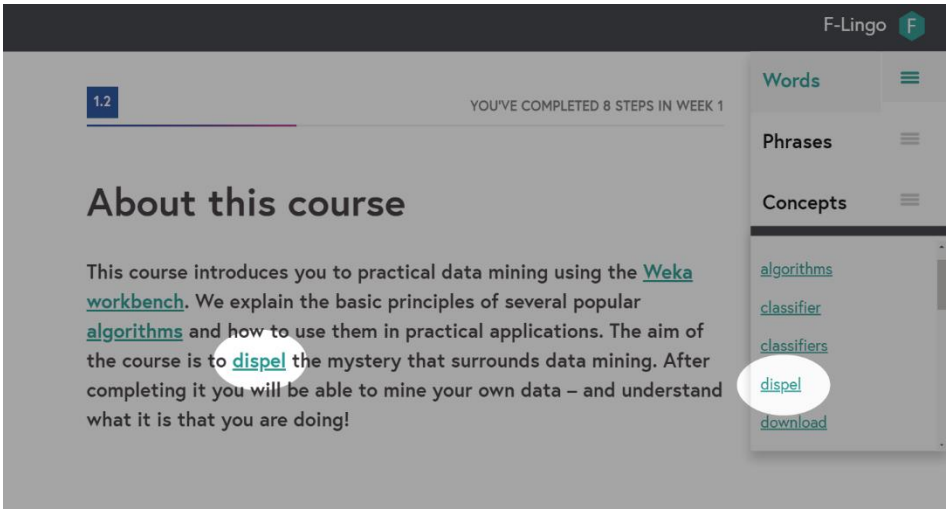


Figure 8.3 F-Lingo: clicking in text versus clicking in the summary list

8.2 Collecting learner data

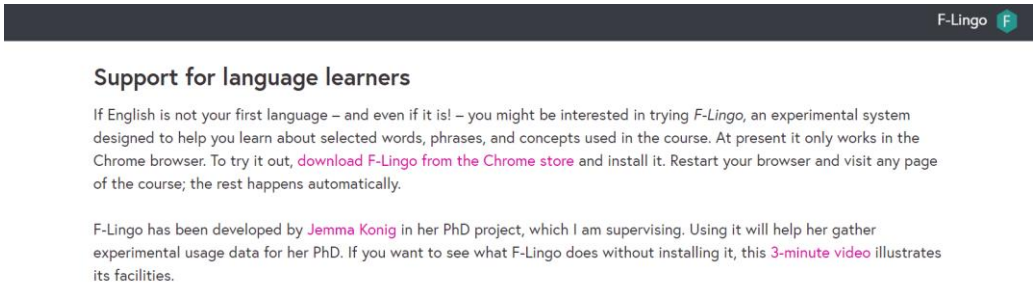
An evaluation has been conducted on user data logged by F-Lingo, analysing learner behaviour in relation to their use of the F-Lingo Chrome extension. F-Lingo has been made available for three FutureLearn courses, *Data Mining with Weka*, *More Data Mining with Weka*, and *Advanced Data Mining with Weka*. The first article in each course includes a section describing F-Lingo and informing learners where they can download it, as shown in Figure 8.4a. Ethical consent was applied for and approved prior to the start date for the study, as shown in Appendix F.3.

8.2.1 Participants

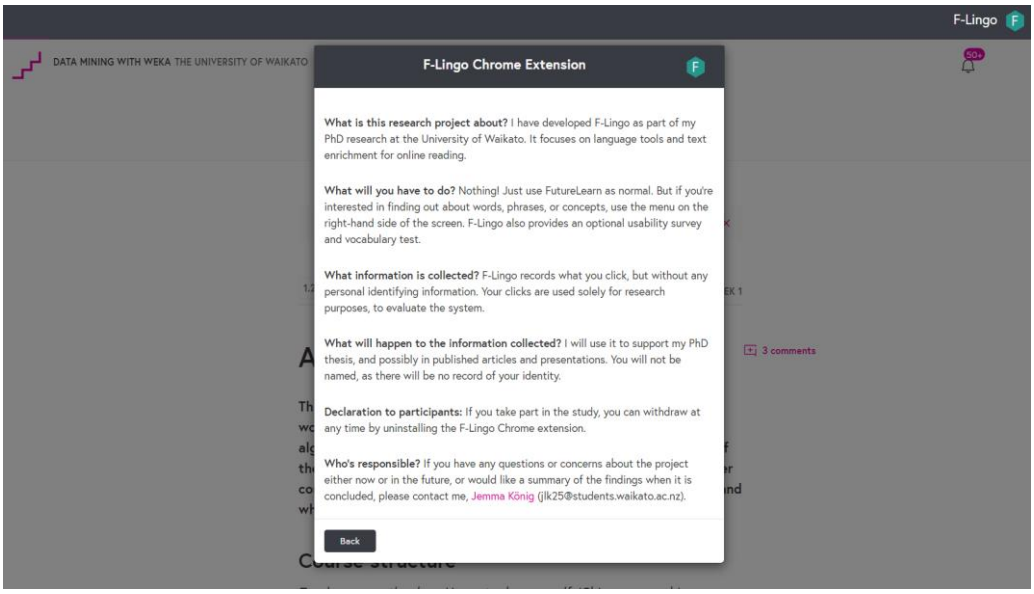
The participant pool for this study was comprised of learners enrolled in one of three Data Mining courses on FutureLearn. Once a learner downloaded F-Lingo, they were given a prompt with information about the study (Figure 8.4b) and the option to join (Figure 8.4c). Learners who consented to F-Lingo logging their data were also asked to enter their first language (L1), and any other languages that they know (L2) (Figure 8.4c). Once a learner gave their consent, they were marked as a *consenting user* and any future interactions with F-Lingo were logged. There are 251 consenting users in this study.

8.2.2 Methodology

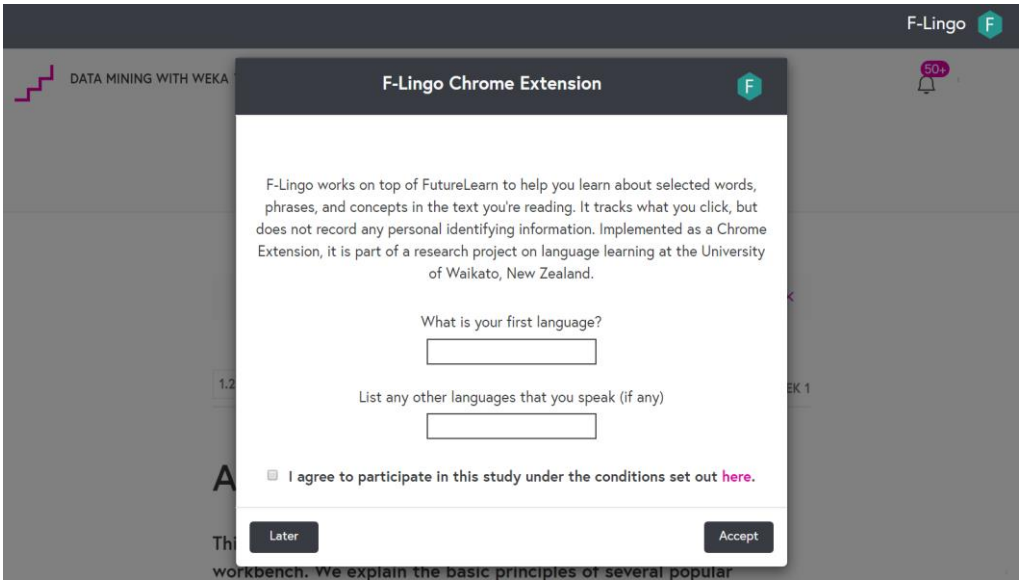
Once a learner has given their consent, F-Lingo begins logging any interaction they



a) Support for language learners



b) F-Lingo information dialog



c) F-Lingo consent dialog

Figure 8.4 F-Lingo: participant recruitment and consent

Table 8.1 Participant-based: participant distribution

	L1 English	L2 English	Both
Installed F-lingo	58	193	251
Used F-Lingo	20	89	109
Percentage	35%	46%	43%

Table 8.2 Participant-based: the number of language spoken by active users

Languages	One	Two	Three	Four	Five
Users	12 (11%)	76 (70%)	12 (11%)	6 (5%)	3 (3%)

have with it. This includes clicking words, phrases, and concepts, both highlighted within the content on the page and within the summary list down the side of the page (shown in Figure 8.3). This results in six types of interaction: clicking a (1) word, (2) phrase, or (3) concept within the course text, or clicking a (4) word, (5) phrase, or (6) concept within the summary list. Each time a learner clicks on a word, phrase, or concept, F-Lingo opens a dialog with additional lexical information. Once the learner has closed this dialog, the following is logged in a database.

$$userID, item, type, location, duration, course, step, datetime$$

where *userID* is a unique ID given to the user, *item* is the item clicked, *type* is the type of item (word, phrase, or concept), *location* is whether it was clicked within the course content (in text) or in the summary list (in list), *duration* is the amount of time that the dialog window remained open, *course* is the course the learner was on when they clicked the item, *step* is the course step they were on, and *datetime* is the date and time when the interaction occurred.

8.3 Participant-based analysis

Table 8.1 shows the number of participants that downloaded F-Lingo, and the number who used it. Of the 251 participants who downloaded it, 109 used it to click words, phrases, or concepts within the text. These *active users* are the focus of the rest of this evaluation. Table 8.1 also shows the distribution of users whose first language (L1) is English, versus those whose second language (L2) is English. Of the 109 active users, 18% were L1 English speakers and 82% were L2.

Table 8.3 Participant-based: distribution of languages

Afrikaans	Dutch (6)	Indonesian	Nepali	Slovenian
Arabic (8)	English (20)	Italian (2)	Odia	Spanish (16)
Bengali	Farsi	Japanese	Persian (3)	Tamil
Brazilian	French	Korean	polish (2)	Turkish (3)
Cantonese	German (3)	Malay	Portuguese (7)	Ukrainian (3)
Chinese (9)	Hindi (2)	Marathi	Russian (5)	Uzbek
Danish	Hungarian	Nederland	Serbian	Vietnamese

Table 8.4 Participant-based: distribution of L2 languages

Catalan (3)	English (90)	Korean	Romanian	Spanish (5)
Croatian (2)	French (11)	Marathi	Russian (2)	Ukrainian
Czech	German (9)	Polish	Serbian	

Table 8.2 shows the number of languages spoken by each user. Only 11% speak one language. This is not surprising, as most of these users are those who speak English as their first language and do not speak any others. 70% speak two languages, and 19% speak three or more.

8.3.1 Languages

Table 8.3 and Table 8.4 show the distribution of L1 and L2 languages. There were 35 L1 languages in total, the most common of which, excluding English, were Spanish, Chinese, Arabic, and Portuguese. L2 language variation was smaller, with only 14 different L2 languages spoken. However, of those, English was by far the most common, with 83% speaking it as a second language. This was followed by French, German, and Spanish. It is not surprising that English was the most common L2 language, given that F-Lingo is directed towards L2 English learners.

8.3.2 Courses

F-Lingo documents which course a learner is in when words, phrases, or concepts are clicked. As shown in Table 8.5, most learners used F- Lingo while participating in *Data Mining with Weka*, followed by *More Data Mining with Weka*, and *Advanced Data Mining with Weka*. However, there were also a significantly

Table 8.5 Participant-based: course distribution

	DMwW	MDMwW	ADMwW
Active F-Lingo users	92 (of 109)	12 (of 109)	9 (of 109)
Enrolled in course	3542	966	647
Percentage	2.6%	1.2%	1.3%

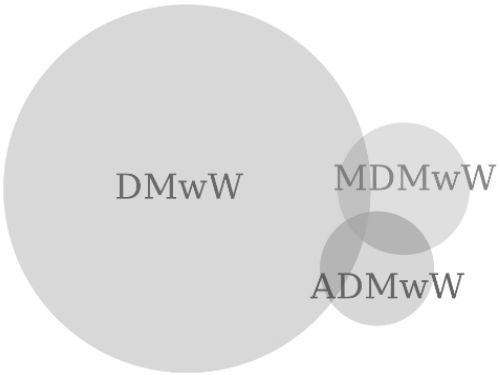


Figure 8.5 Participant-based: overlap between courses

larger number of learners enrolled in Data Mining than the other two courses. Of those enrolled in each course, 2.6% became active users of F-Lingo in Data Mining, 1.2% in More Data Mining, and 1.2% in Advanced Data Mining.

Some learners interacted with F-Lingo in more than one course, shown in Figure 8.5. Two used F-Lingo in both Data Mining and More Data Mining, two used it in Data Mining and Advanced Data Mining, two used it in More Data Mining and Advanced Data Mining, and one used F-Lingo in all three courses.

8.4 Interaction-based analysis

Each time a word, phrase, or concept is clicked, F-Lingo logs that interaction. Each interaction is stored with (1) whether the user clicked a word, phrase, or concept, and (2) whether the item was clicked within the text on the page, or in the summary list on the side of the page (shown earlier in Figure 8.3).

8.4.1 Words, phrases, and concepts

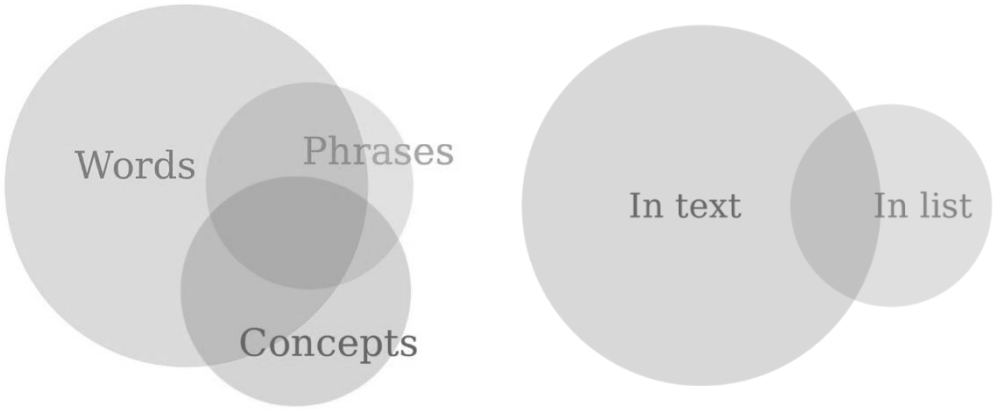
Table 8.6 shows the total number of interactions that were logged, and whether each interaction was with a word, phrase, or concept (row 1). There were 519 interactions

Table 8.6 Interaction-based: words, phrases, and concepts

	Words	Phrases	Concepts
Interactions	308 (519) (59%)	83 (519) (16%)	128 (519) (26%)
F-Lingo users	92 (109) (84%)	30 (109) (28%)	37 (109) (34%)

Table 8.7 Interaction-based: in text and in list

	In text	In list
Interactions	444 (519) (86%)	71 (519) (14%)
F-Lingo users	92 (109) (84%)	29 (109) (27%)



a) Words, phrases, and concepts b) In text, versus in lists

Figure 8.6 Interaction-based: overlap in interactions

in total. Words were clicked the most (58%), followed by concepts (26%), and phrases (16%). Table 8.6 also shows the number of users who interacted with words, phrases, and concepts (row 2). More users interacted with words (84%) than concepts (34%) or phrases (28%). However, as with course distribution, some users interacted with a combination of each. As shown in Figure 8.6a. 22% of users interacted with words and phrases, 20% with words and concepts, 13% with concepts and phrases, and 10% interacted with all three.

8.4.2 In text, versus in lists

Table 8.7 shows the number of times items were clicked, either within the course text or in the summary list provided on the side of each page (row 1). In-text clicks were significantly higher (86%) then in-list clicks (14%). Table 8.7 also shows the

Table 8.8 Time-based: time-based analysis for words, phrases, and concepts

	Words	Phrases	Concepts
Mean duration	22.1 s	20.3 s	41.4 s
Mean word count	164	134	94
Words per minute	445 wpm	398 wpm	136 wpm

Table 8.9 Time-based: time-based analysis for in-text and in-list interactions

	In text	In list
Mean duration	25.7 s	28.7 s
Mean word count	147	148
Words per minute	343 wpm	310 wpm

number of users who interacted with items, either by clicking in the course text, or in the summary lists (row 2). More users (84%) interacted with in-text items than (27%) in-list items. However, some users interacted with both, as shown in Figure 8.6b. 10% of users interacted with items by clicking on them within the course text and using the summary list.

8.5 Time-based analysis

Each time a learner clicks a word, phrase, or concept, whether in the text or in the summary list, a dialog window opens that provides them with additional information. After a learner clicks an item, F-Lingo keeps track of how long the corresponding dialog window was open.

8.5.1 Words, phrases, and concepts

Table 8.8 shows the mean duration that learners kept dialog windows open, for words, phrases, and concepts. Extreme outliers were removed using the lower and upper quartiles +– three times the inner-quartile range. Concept-dialogs stayed open for the longest period of time, followed by word-dialogs, then phrase-dialogs. This table also shows the mean word counts for dialogs opened by learners. Although concept-dialogs were open longest, they also contained significantly less words then both word-dialogs and phrase-dialogs. To conduct a fair comparison between them, I have calculated the average words-per-minute, as follows.

$$wpm = \frac{wordcount}{duration} \times 60$$

Although I have calculated the words-per-minute (wpm), I do not assume that users always, or ever, read every word in the dialog. However, producing wpm rates gives an indication of how long users spend looking at the dialog, in relation to how large, or how many words, were shown.

L2 reading speeds can vary considerably, but the average reading speed for a native speaker (L1) is between 200-250 wpm. All, except one, of the dialogs resulted in a wpm rate approximately one and a half to two times faster than this, suggesting that learners rarely read all the text in a dialog. The one exception to this was concept-dialogs, which had a rate of 136 wpm. This suggests that learners spend more time reading concept-dialogs than word-dialogs or phrase-dialogs.

8.5.2 In text, versus in lists

Like the word, phrase, concept comparison, Table 8.9 shows the mean duration that learners kept dialog windows open for in-text and in-list interactions. Both have similar word counts, which is not surprising since they both include a selection of word-dialogs, phrase-dialogs, and concept-dialogs. However, both also have similar durations, and as a result, have similar wpm rates. This suggests that the amount of time learners spend reading dialogs does not change based on how they are accessing them, i.e. within text or summary lists.

8.6 Content-based analysis

As mentioned in Section 8.2, each time a word, phrase, or concept is clicked, F-Lingo logs information about the interaction, including data relating to the content of the item clicked (i.e. the word *algorithms*) and its type (word, phrase, or concept). This section looks at the content of the clicked words, phrases, and concepts. For example, what ten words were clicked most often?

8.6.1 Unique items

Table 8.10 shows the number of unique words, phrases, and concepts clicked, in relation to the total number of items clicked (in brackets), and the number of duplicates. *Words* showed the least variety, with 66% of clicked words being

Table 8.10 Content-based: unique words, phrases, and concepts clicked

	Words	Phrases	Concepts
Unique items	102 (304)	43 (83)	53 (128)
Duplicates	202 (66%)	40 (48%)	75 (59%)

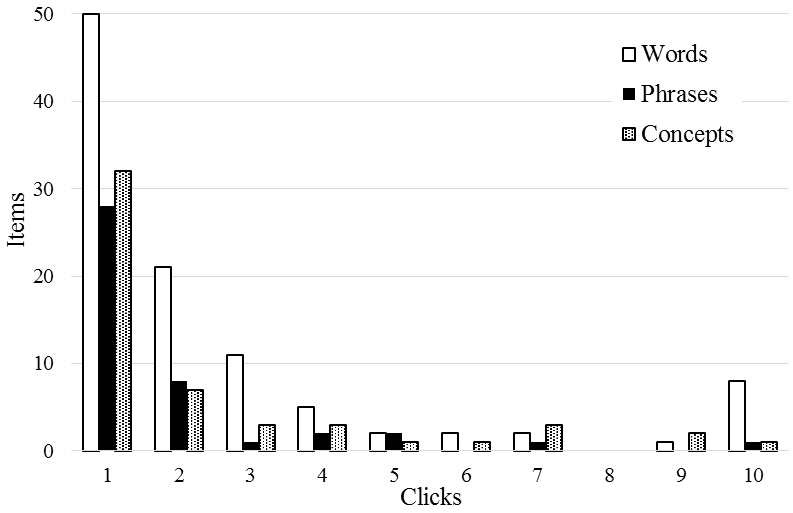


Figure 8.7 Content-based: words, phrases, and concepts, ordered by clicks

Table 8.11 Content-based: 10 most clicked words, phrases, and concepts

Words			
Item	Clicks	Item	Clicks
Download	18	Weka	15
Supervise	17	Algorithm	13
Classifier	17	Unsupervised	12
Phrases			
Item	Clicks	Item	Clicks
Data mining	10	Big data	5
Attribute selection	7	Visualize data	4
Sort of data	5	Image classification	3
Concepts			
Item	Clicks	Item	Clicks
Data mining	11	Weka	7
Dataset	9	Machine learning	7
Big data	9	Decision tree	7

duplicates, followed by *concepts* (59%). *Phrases* had the most variety, with less than half (48%) of clicked phrases being duplicates. This is not a drastic difference, but it is noticeable. This shows that learners are more likely to click the same word then they are phrases, or concepts.

8.6.2 Frequent items

Figure 8.7 shows the number of words, phrases, and concepts that were clicked once, twice, three times, and so on. As mentioned earlier, and shown in Table 8.10, 102 unique words were clicked in total. Of those, 50 were only clicked once, 21 were clicked twice, and 11 were clicked three times. The majority of words were clicked less than 5 times, but some were clicked more than 10, some up to 18 times.

Like words, 43 unique phrases were clicked in total. Of those, 28 were only clicked once and 8 were clicked twice. The majority of phrases were clicked no more than twice, and only one was clicked 10 times. Finally, 53 unique concepts were clicked in total, and of those, 32 were clicked once, 7 were clicked twice, and 3 were clicked three times. The majority of concepts were clicked less than 4 times. Table 8.11 shows the 10 most frequently clicked words, phrases, and concepts.

8.7 Discussion

The four learner-data based analyses: participant-based, interaction-based, time-based, and content-based (described in Sections 8.3 to 8.6), focus on learner behaviour in relation to their use of F-Lingo. These evaluations were limited to usability-based results, rather than considering the effectiveness of F-Lingo for language acquisition, nevertheless we can draw several conclusions from the analysis.

1. Learners interested in using F-Lingo are more likely to be L2 English speakers than L1 English speakers.
2. Of the three courses available with F-Lingo, *Data Mining with Weka* may be best suited to providing additional language resources to learners, even if only because of its higher enrolment numbers.
3. Learners have been shown to use F-Lingo to obtain additional lexical information for words, phrases, and concepts, both within course content, and through the use of the summary lists.
4. Learners interact with words more often than phrases or concepts.

5. Learners interact with items within course content more often than summary lists.
6. Learners spend more time looking at additional lexical information for concepts than for words or phrases.
7. Learners spend similar amounts of time looking at additional lexical information when it is obtained from within course content as when it is obtained from summary lists.
8. Learners are more likely to click the same word then they are the same phrase or concept.

Chapter 9
Conclusion

This thesis examines the untapped potential of using online courses for content-based language learning, and investigates three related challenges. First, online courses offer subjects in a variety of domains, but supporting the language in different domains requires knowledge of the vocabulary present in each. Second, in order to determine whether content-based learning has been successful, each learner's vocabulary growth within the domain needs to be measured, but domain-specific vocabulary tests are few and far between. Third, in order for content-based learning to be successful, the integration of language teaching into non-language subjects must be done well. Not only that, but vocabulary support should include not just single words, but also multi-word lexical items. This raises the challenge of developing a system that integrates itself into online courses without disturbing

the original content, but while still providing an adequate level of language learning support.

9.1 Revisiting the thesis statement

This thesis addressed three challenges: 1) identifying domain-specific vocabulary 2) automating a receptive vocabulary test, and 3) integrating language resources into online courses. The following section discusses and summarise the results of these investigations.

9.1.1 Identifying domain-specific vocabulary

The first challenge was to use online course content and a set of automated processes to create domain-specific corpora and wordlists entirely automatically.

1. *Developing a set of automated processes that use online course content to build domain-specific corpora and wordlists.*

This was met through the work outlined in Chapter 3. First, a Chrome extension, the *CourseCrawler*, was developed to traverse online courses, extracting both written and spoken course content. Next, a Python application, the *CourseCorpusBuilder*, was developed to process the course content, building and annotating a corpus from it. Finally, a second Python application, the *CourseWordlistBuilder*, was developed to generate domain-specific wordlists, where word selection is based on a set of frequency and range criteria used in the creation of the Academic Word List.

Combining the three allows linguists and language teachers to build domain-specific corpora and wordlists, entirely automatically, using online course content from FutureLearn. The corpus building applications holds one important advantage over others: they can be applied to any FutureLearn course. This allows researchers and language teachers to build corpora from a vast selection of topics whose content is known to be of high quality. The wordlist generation application has another important advantage: it is generalised. Whereas criteria for including words in wordlists usually depend on their purpose, and therefore change with each individual case, this technique can be applied to any corpus built from online course content to create a wordlist of vocabulary specific to that course's domain.

Chapter 3 illustrates this by applying the automated processes to three practical data mining courses, *Data Mining with Weka*, *More Data Mining with*

Weka, and *Advanced Data Mining with Weka*, on the FutureLearn MOOC consortium. This resulted in the automatic creation of the 200,000 word *DMwW corpus*, and the 571 word *DMwW wordlist*.

9.1.2 Automating a receptive vocabulary test

The second challenge can be broken into two areas: generating domain-specific pseudowords, and using those pseudowords to generate domain-specific versions of the EFL Vocabulary Test.

2. *Recreating an existing vocabulary test automatically using domain-specific vocabulary.*

This challenge was met through the work outlined in Chapters 4 and 5. First, Chapter 4 introduced a new pseudoword generation technique, chaining character-grams to form pseudowords, and described the development of a Python application, the *CGCA algorithm*, that uses it. The technique holds two main advantages (1) it does not require any knowledge of the language, thereby facilitating the generation of pseudowords in any language, and (2) the pseudowords reflect the wordlist used to create them, thereby facilitating the generation of pseudowords specific to a certain domain. Chapter 4 illustrates this by applying the CGCA algorithm to a wordlist to generate a set of 800 pseudowords using character-grams that vary in size between 2-grams and 8-grams.

The second half of Chapter 4 evaluated the character-gram chaining technique, introducing a set of criteria for evaluating pseudowords, both in terms of their orthographic fit in the target language, and their suitability for use in lexical processing and language teaching. The evaluation criteria were used to provide a comparison with other current pseudoword lists. The results of the legal evaluation found that using this technique resulted in higher counts of orthographically legal pseudowords than other techniques and illustrated that there is a need for post-production criteria for evaluating pseudowords (Section 4.7.2). The results of the suitability evaluation found that not all pseudowords are formed the same, and that some pseudowords may be better suited to certain lexical tasks than others. This illustrates that the evaluation criteria could be used to determine which pseudowords most closely match others (Section 4.7.3).

Chapter 5 introduced a new technique for generating vocabulary tests entirely automatically, based on the structure of the well-founded EFL Vocabulary

Test, and using pseudowords generated by the CGCA algorithm. The chapter also described the development of a Python application, the *AEFL algorithm* that uses it. This new technique holds two main advantages (1) it does not require any knowledge of the language, thereby facilitating the automatic generation of vocabulary tests, and (2) the tests reflect the wordlist used to create them, thereby facilitating the generation of vocabulary tests for certain domains. Chapter 5 illustrates this by applying the AEFL algorithm to the same wordlists that were used to create the original EFL test. Two studies were conducted where participants were asked to complete both the original EFL and the automatically generated AEFL vocabulary tests. Results from both studies found that there was no statistically significant difference between the EFL and AEFL tests for majority of participants (Sections 5.4.6 and 5.5.4).

The second half of Chapter 5 evaluated the way in which the EFL Vocabulary Test is scored. Researchers have debated the original scoring method and have suggested several alternatives. This chapter investigated which scoring methods best evaluate the EFL Vocabulary Test, based on the results of two studies, and compared the results with those produced by other researchers. The results found that *proportionate hit rate* and *correction for guessing* were the scoring methods best suited to the EFL Vocabulary Test (Section 5.9.2), and that the results from the two studies supported patterns put forward by other researchers (Section 5.9.3). However, none of the alternative scoring methods were found to be entirely satisfactory, supporting Meara's claims and suggesting that there is need for more research in this area (Section 5.10).

9.1.3 Integrating language resources into online courses

Finally, the third challenge was to integrate language resources into online courses without disturbing the original content.

3. *Integrating language resources into online courses without disturbing the original content.*

This challenge was met through the work outlined in Chapters 6, 7 and 8. First, Chapter 6 investigated content-based language learning and reviewed six existing online language applications, It identified four vocabulary items (words, collocations, lexical bundles, and disambiguated terms), and four language resources (definitions, example sentences, related collocations, and disambiguated

descriptions), based on the results obtained by comparing the features present in the existing systems. Finally, it investigated the pedagogy for acquiring these items using the resources listed, and outlined the design considerations for an integrated language system called F-Lingo.

Chapter 7 described the development of F-Lingo, a Chrome extension that works on top of FutureLearn to support content-based language learning. Section 7.2 described pre-processing course content; identifying words excluding those present in the General Service List; identifying collocations by their syntactic patterns, identifying lexical bundles using a predefined list; and disambiguating concepts using Wikipedia Miner. Section 7.3 illustrated supporting the noticing hypothesis by traversing online course content and highlighting words, phrases, and concepts within the text, and Section 7.4 illustrated using external resources to provide learners with definitions from Wiktionary, example sentences from FLAX, and disambiguated descriptions from Wikipedia. Each of the above were illustrated using examples from F-Lingo, applied to *Data Mining with Weka*, *More Data Mining with Weka*, and *Advanced Data Mining with Weka*, three FutureLearn courses

Chapter 8 evaluated the F-Lingo Chrome extension in two ways: an expert heuristic evaluation, and a data-based study. First, an expert heuristic evaluation was conducted, in the form of a cognitive walkthrough, and the aim of the evaluation was to determine whether F-Lingo integrated language resources into online courses without disturbing the original content. The results of the evaluation were structured around ten heuristics that test the design and usability of a user interface. Five shortfalls were identified and rectified based on the expert heuristic evaluation. A further two minor shortfalls were identified and acknowledge but not rectified. The first related to the format of external URLs and was outside of my control. The second contradicted another heuristic so was excluded. The final results for the F-Lingo Chrome extension met all ten heuristics, including the eighth heuristic *aesthetic and minimalist design.* This suggests that F-Lingo's interface meets design and usability standards, and in turn does not disturb the original content.

Second, an evaluation was conducted using data that was logged when learners used F-Lingo. This data was analysed to interpret learner behaviour in relation to their use of the F-Lingo Chrome extension. Learners enrolled in *Data*

Mining with Weka, *More Data Mining with Weka*, and *Advanced Data Mining with Weka* were given the opportunity to install F-Lingo and take part in the evaluation. Information about their L1 and L2 languages (if any), the words, phrases, and concepts clicked, and the duration that they looked at definitions, example sentences, and so on, was logged in a database. As illustrated in Section 8.7, eight behavioural observations were noted based on the data gathered. (1) Learners interested in using F-Lingo were more likely to be L2 English speakers than L1 English speakers. (2) Of the three courses available with F-Lingo, *Data Mining with Weka* may be best suited to providing additional language resources to learners, even if only because of its higher enrolment numbers. (3) Learners have been shown to use F-Lingo to obtain additional lexical information for words, phrases, and concepts, both within course content, and using summary lists. (4) Learners interact with words more often than phrases or concepts. (5) Learners interact with items within course content more often than summary lists. (6) Learners spend longer periods of time looking at additional lexical information for concepts than words or phrases. (7) Learners spend similar periods of time looking at additional lexical information when it is obtained from summary lists and when it is obtained from within course content. (8) Learners are more likely to click the same word then they are the same phrase or concept.

9.2 Limitations of this work

The work outlined in this thesis resulted in the creation of several language-based software applications that automate what could otherwise be an arduous task that requires extensive language knowledge. However, it should be noted that these software applications have been created as an aid for researchers, applied linguistics and language teachers, rather than a replacement for them. As such, although the tools are automated, there may still be times when human input is required.

First, the CourseCorpusBuilder annotates corpora with headwords, and the CourseWordlistBuilder uses these annotations to build wordlists from corpora. Headword annotation involves two steps: (1) tagging word types with their headwords according to Nation et al. (2002), and (2) lemmatizing any word types that were not recognised as belonging to a word family. However, word types exist that neither belong to a word family nor are lemmatized by NLTK, even though their headword may seem instinctive to us. Once such example of this is *updateable*,

which according to Nation et al. (2002) does not belong to the *update* word family, and is not lemmatized down to *update* by NLTK. In cases like these, it is up to the researcher to complete a final check on their wordlists, ensuring that any untagged word types are either included or excluded based on their own requirements. This is discussed further in Section 3.5.3 (annotating a corpus) and Section 3.6 (automating wordlists).

Next, Chapter 5 discussed the implementation of the AEFL algorithm, an automated software application that generates domain-specific versions of the EFL Vocabulary Test using the words present in a domain-specific wordlist. Before the EFL was chosen as the base vocabulary test for the AEFL algorithm, several other existing and well-founded matching-based and checklist-based vocabulary tests were considered. The EFL was selected based on four criteria (Section 5.1) that outline the minimum features required for this work: (1) the test needed to be automatable, (2) it needed to provide at least a rough estimate of vocabulary, (3) full and detailed instruction on the test needed to be available, and (4) a full copy of the test needed to be available. These criteria led to the selection of a checklist-based test, specifically the EFL. Here, the second criteria should be highlighted, "it needed to provide at least a rough estimate of vocabulary". As mentioned in Section 5.1, match-based tests inarguably measure more in-depth knowledge than checklists. However, the criteria only required a rough estimate of vocabulary, and given their use of extensive language knowledge, match-based tests cannot be easily automated. Although this resulted in the selection of the EFL Vocabulary Test, the difference in measureable language knowledge between the two test types should still be noted. In his second edition of the EFL Vocabulary Test, Meara (2010) states that learners "should NOT use these tests in situations where very high levels of accuracy are required, and important decisions, which might have serious legal consequences, should not be based on the results of the tests" (Meara, 2010). The same should be said for the AEFL algorithm, which can be used to create a "rough" lexical profile, but should not be used in cases where high levels of accuracy are required.

Finally, the design, implementation, and evaluation of the F-Lingo Chrome extension was discussed in Chapters 6, 7, and 8. F-Lingo was developed based on several content-based language learning concepts (Section 6.1), and the inclusion of certain features was determined by both a feature comparison (Section

6.3) and based on exiting research in language teaching, specifically in relation to techniques behind learning words, phrases, and concepts (Sections 6.4 and 6.5). F-Lingo was evaluated in two ways: an expert evaluation, and a learner-data based evaluation, the culmination of which evaluated the ability to develop and utilise a software approach to content-based language learning, and the usability of this software. However, these evaluations were limited to usability-based results, and did not consider the value of the software for language learning itself (similarly expressed in Section 8.7). This thesis focused on the development of these applications, the creation of which has created a catalyst for more learner-based evaluations in the future, as discussed in the following section about future work.

9.3 Future work

Chapter 3 illustrated using automated processes to create the *DMwW corpus*, which although small (200,000 running words), is remarkably well balanced between spoken and written English (100,000 each). An interesting future endeavour would be to apply the automated processes, described in Chapter 3Chapter 2, to other FutureLearn courses, collecting course content and building corpora from other FutureLearn courses. Are other courses as well suited to becoming corpora, in terms of balance between spoken and written English, or are the *Data Mining with Weka* courses unique?

A plethora of Python based software applications have been introduced throughout this thesis. The *CourseCorpusBuilder* and *CourseWordlistBuilder* in Chapter 3, the *CGCA algorithm* in Chapter 4, the *AEFL algorithm* in Chapter 5, and the F-Lingo pre-processing scripts in Chapter 7. Although Python is an ideal language for text processing and natural language processing, it does limit those who are able to run and use it. Redeveloping each of these Python applications into web-based solutions would allow a larger audience, including linguists and language teachers, to take advantage of the practical applications that were developed for this thesis.

Most importantly, Chapter 8 illustrates how the F-Lingo Chrome extension has been evaluated. It has been subjected to evaluations for determining whether it meets design and usability standards without disturbing existing content, and whether (and how) learners use it. However, I have not tested whether it influences language acquisition. The next step would be to study whether long-term use of an

enriched course improves learners' vocabulary. This could be evaluated in extensive longitudinal studies. F-Lingo could be applied to a course that has high L2 learner enrolments. Their interaction with F-Lingo, along with regular domain-specific vocabulary tests, and perhaps short and long answer written assessments, could be used to measure vocabulary growth. Comparing the vocabulary growth of learners who use F-Lingo with those, in the same course, who do not, could provide some insight into whether enriching online courses with language resources influences language acquisition. Although it does not fall within the scope of this thesis, it is hoped that this work will serve as a catalyst, allowing others to apply F-Lingo to courses and undergo appropriate evaluations.

Finally, F-Lingo is currently only available for use with the FutureLearn consortium. Although it can be applied to any FutureLearn course with very little effort, it cannot currently be applied to other online course platforms. Some aspects of its design have been tailored specifically to FutureLearn. It has been developed to only highlight words, phrases, and concepts on particular pages, such as articles and discussions, and of those pages, it only highlights items in particular parts of the content, for example, it does not highlight items in learners' posts. The visual feel of F-Lingo has also been developed around FutureLearn. The majority of F-Lingo's styling is automatically inherited from FutureLearn's CSS style sheets. However, some elements were specifically chosen to work with FutureLearn, for example, the highlighting colour (teal) was chosen because it gives enough of a contrast with FutureLearn's colours to stand out, but does not clash in a way that would disturb the original feel of the site. The final future endeavour would be to develop a version of F-Lingo for Coursera and a version for EdX. Most of F-Lingo's processes are automated, so only those that are tailored specifically to the look and feel of FutureLearn would have to be altered. There is also the possibility of developing one version of F-Lingo that works on top of all three course platforms, or even any website. However, one of the main challenges that was addressed when developing F-Lingo was that it not disturb the original content on a page; tailoring F-Lingo to FutureLearn, even if only minutely, has allowed us to achieve this, and taking the time to tailor it to other course platforms in the future would ensure that it continues to do so.

Massive Open Online Courses have the potential to revolutionize education, opening university-level study up to new classes of learners, allowing

them to study from anywhere in the world, free of charge, and at their own pace. However, one major disadvantage is their reliance on language knowledge. F-Lingo has the potential to alleviate this disadvantage, providing learners with language support: identifying domain-specific vocabulary, administering vocabulary tests tailored to the course, and offering language resources for words, phrases, and concepts. Working seamlessly on top of existing platforms, F-Lingo has the potential to greatly expand the pool of online learners. F-Lingo takes the world a step closer to fulfilling Kofi Annan's inspirational message that "education is the premise of progress, in every society, in every family."

References

Adamson, H. D. (1993). *Academic Competence: Theory and Classroom Practice : Preparing Esl Students for Content Courses*. New York: Longmans.

Alanen, R. (1995). Input enhancement and rule presentation in second language acquisition. *Attention and awareness in foreign language learning*, 259-302.

Anthony, L. (2004). *AntConc: A learner and classroom friendly, multi-platform corpus analysis toolkit*. Paper presented at the Interactive Workshop on Language e-Learning.

Baayen, R. H., Piepenbrock, R., & van Rijn, H. (1993). The CELEX lexical data base on CD-ROM.

Baayen, R. H., & Schreuder, R. (2011). *Morphological structure in language processing* (Vol. 151): Walter de Gruyter.

Balota, D., Yap, M., Cortese, M., Hutchison, K., Kessler, B., Loftis, B., . . . Treiman, R. (2007). The English lexicon project. *Behavior Research Methods, 39*(3), 445-459.

Baroni, M., & Ueyama, M. (2006). *Building general-and special-purpose corpora by web crawling*. Paper presented at the 13th NIJL International Symposium. Language Corpora: Their Compilation and Application.

Bauer, L., & Nation, I. S. P. (1993). Word families. *International Journal of Lexicography, 6*(4), 253-279.

REFERENCES

Beeckmans, R., Eyckmans, J., Janssens, V., Dufranne, M., & Van de Velde, H. (2001). Examining the Yes/No vocabulary test: Some methodological issues in theory and practice. *Language Testing, 18*(3), 235-274.

Bell, T. C., Cleary, J. G., & Witten, I. H. (1990). *Text compression* (Vol. 348): Prentice Hall Englewood Cliffs, NJ.

Berko, J. (1958). The child's learning of English morphology. *Word, 14*(2-3), 150-177. doi:10.1080/00437956.1958.11659661

Biber, D. (1993). Representativeness in corpus design. *Literary and linguistic computing, 8*(4), 243-257.

Biber, D., & Barbieri, F. (2007). Lexical bundles in university spoken and written registers. *English for specific purposes, 26*(3), 263-286.

Bijma, F., Jonker, M., & van der Vaart, A. (2017). *An introduction to mathematical statistics*: Amsterdam University Press.

Bird, S., & Tan, L. (2018). The Natural Language Toolkit: NLTK Project. Retrieved from https://www.nltk.org/index.html

Bird, S., Tan, L., & Nothman, J. (2018). PunktSentenceTokenizer: NLTK Project. Retrieved from https://www.nltk.org/_modules/nltk/tokenize/punkt.html

Brinton, D., & Holten, C. (1989). What novice teachers focus on: The practicum in TESL. *TESOL Quarterly, 23*(2), 343-350.

British National Corpus. (2007). (version 3 XML ed.): Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium.

Brown, T. S., & Perry Jr, F. L. (1991). A comparison of three learning strategies for ESL vocabulary acquisition. *TESOL Quarterly, 25*(4), 655-670.

Buchmeier, M. (2008a). Bilingual Dictionaries for Offline Use - Italian frequency list. from Wiktionary https://en.wiktionary.org/wiki/User:Matthias_Buchmeier#Italian_frequency_list

Buchmeier, M. (2008b). Bilingual Dictionaries for Offline Use - Spanish frequency list. from Wiktionary https://en.wiktionary.org/wiki/User:Matthias_Buchmeier# Spanish_frequency_list

Buchmeier, M. (2009). Bilingual Dictionaries for Offline Use - German frequency list. from Wiktionary https://en.wiktionary.org/wiki/User:Matthias_Buchmeier# German_frequency_list

Burnard, L. (2007). *Reference Guide for the British National Corpus* (L. Burnard Ed. XML ed.). Oxford University Computing Services: The Research Technologies Service.

Cardenas, J. M. (2009). *Phonics instruction using pseudowords for success in phonetic decoding.* Florida International University, FIU Electronic Theses and Dissertations. Retrieved from http://digitalcommons.fiu.edu/etd/139

CARLA. (2019). CoBaLTT instructional modules. Retrieved from http://carla.umn.edu/cobaltt/modules/index.html

Carroll, J. B., Davies, P., & Richman, B. (1971). *The American Heritage word frequency book*: Houghton Mifflin Boston.

Carroll, S. E. (1999). Putting 'input'in its proper place. *Second Language Research, 15*(4), 337-388.

Coady, J., & Huckin, T. (1997). *Second language vocabulary acquisition: A rationale for pedagogy*: Cambridge University Press.

Coursera. (2019). About Coursera. Retrieved from https://blog.coursera.org/about/

Coxhead, A. (1998). *An Academic Word List* (Vol. 18): School of Linguistics and Applied Language Studies.

Coxhead, A. (2000). A new academic word list. *TESOL Quarterly, 34*(2), 213-238.

Coyle, D. (1999). Supporting students in content and language integrated contexts: Planning for effective classrooms. In J. Masih (Ed.), *Learning Through a Foreign*

Language – Models, Methods and Outcomes (pp. 46-62). London: Centre for Information on Language Teaching and Research (CILT).

Crandall, J. J. (1992). Content-centered learning in the US. *Annual Review of Applied Linguistics, 13*, 110-126.

Darn, S. (2006). Content and language integrated learning (CLIL): A European Overview. *ERIC Institute of Education Sciences*.

Davies, M. (2002). The Corpus of Historical American English (COHA). from Brigham Young University https://corpus.byu.edu/coha/

Davies, M. (2008). The Corpus of Contemporary American English (COCA). from Brigham Young University https://corpus.byu.edu/coca/

Davies, M. (2013a). Global Web-Based English (GloWbE). from Brigham Young University https://corpus.byu.edu/glowbe/

Davies, M. (2013b). News on the Web (NOW). from Brigham Young University https://corpus.byu.edu/now/

Davies, M. (2015). The Wikipedia Corpus. from Brigham Young University https://corpus.byu.edu/wiki/

Dónaill, C. Ó., & Gimeno-Sanz, A. (2013). Tools for CLIL Teachers. *The EuroCALL Review, 21*(2), 56-63.

Dourda, K., Bratitsis, T., Griva, E., & Papadopoulou, P. (2014). Content and language integrated learning through an online game in primary school: a case study. *Electronic Journal of e-Learning, 12*(3), 243-258.

Duyck, W., Desmet, T., Verbeke, L. P., & Brysbaert, M. (2004). WordGen: A tool for word selection and nonword generation in Dutch, English, German, and French. *Behavior Research Methods, Instruments and Computers, 36*(3), 488-499. doi:10.3758/BF03195595

edX. (2019). About edX. Retrieved from https://www.edx.org/about-us

Firth, J. (1951). *Papers in Linguistics*. UK: Oxford University Press.

FutureLearn. (2019a). About FutureLearn. Retrieved from https://www.futurelearn.com/about-futurelearn

FutureLearn. (2019b). Data Mining with Weka. Retrieved from https://www.futurelearn.com/courses/data-mining-with-weka

Gass, S. M. (2017). *Input, Interaction, and the Second Language Learner*: Routledge.

Gass, S. M., Svetics, I., & Lemelin, S. (2003). Differential effects of attention. *Language learning, 53*(3), 497-546.

Gimeno, A., Seiz, R., De Siqueira, J. M., & Martinez, A. (2010). Content and language integrated learning in higher technical education using the inGenio online multimedia authoring tool. *Procedia-Social and Behavioral Sciences, 2*(2), 3170-3174.

Graham, S., Harris, K. R., & Loynachan, C. (1993). The Basic Spelling Vocabulary List. *The Journal of Educational Research, 86*(6), 363-368.

Grier, J. B. (1971). Nonparametric indexes for sensitivity and bias: computing formulas. *Psychological bulletin, 75*(6), 424.

Groff, P. (2003). The Usefulness of Pseudowords. Retrieved from http://www.nrrf.org/old/essay_pseudowords.html

Halliday, M. A. (1966). Lexis as a linguistic level. In C. E. Bazell & J. R. Firth (Eds.), *In Memory of J.R. Firth* (pp. 150-161). London: Longmans.

Hindmarsh, R. (1986). *Cambridge English lexicon: a graded word list for materials writers and course designers*: Cambridge University Press.

Hu, M., & Nation, P. (2000). Unknown vocabulary density and reading comprehension. *Reading in a foreign language, 13*(1), 403-430.

Huibregtse, I., Admiraal, W., & Meara, P. (2002). Scores on a yes-no vocabulary test: Correction for guessing and response style. *Language Testing, 19*(3), 227-245.

Hundt, M., Nesselhauf, N., & Biewer, C. (2007). *Corpus linguistics and the web*: BRILL.

Izumi, S. (2002). Output, input enhancement, and the noticing hypothesis: An experimental study on ESL relativization. *Studies in second language acquisition, 24*(4), 541-577.

Jäppinen, A.-K. (2005). Thinking and content learning of mathematics and science as cognitional development in content and language integrated learning (CLIL): Teaching through a foreign language in Finland. *Language and Education, 19*(2), 147-168.

Jourdenais, R., Ota, M., Stauffer, S., Boyson, B., & Doughty, C. (1995). Does textual enhancement promote noticing? A think-aloud protocol analysis. *Attention and awareness in foreign language learning*, 183-216.

Jurasek, R. (1993). Foreign languages across the curriculum: A case history from Earlham College and a generic rationale. *Language and content: Discipline-and content-based approaches to language study*, 85-102.

Kaplan, A. M., & Haenlein, M. (2016). Higher education and the digital revolution: About MOOCs, SPOCs, social media, and the Cookie Monster. *Business Horizons, 59*(4), 441-450.

Kazemi, M., Katiraei, S., & Rasekh, A. E. (2014). The impact of teaching lexical bundles on improving Iranian EFL students' writing skill. *Procedia-Social and Behavioral Sciences, 98*, 864-869.

Keuleers, E., & Brysbaert, M. (2010). Wuggy: A multilingual pseudoword generator. *Behavior Research Methods, 42*(3), 627-633.

Knight, S. (1994). Dictionary use while reading: The effects on comprehension and vocabulary acquisition for students of different verbal abilities. *The Modern Language Journal, 78*(3), 285-299.

Krueger, M., & Ryan, F. (1993). *Language and Content: Discipline and Content-based Approaches to Language Study* (Vol. 3). Lexington, MA: DC Heath & Co.

Kučera, H., & Francis, W. N. (1967). *Computational analysis of present-day American English* (1 ed.): Brown University Press.

Laufer, B. (1989). What percentage of text-lexis is essential for comprehension. *Special language: From humans thinking to thinking machines, 316323*.

Laufer, B. (1992). How much lexis is necessary for reading comprehension? In *Vocabulary and applied linguistics* (pp. 126-132): Springer.

Leow, R. P. (2001). Do learners notice enhanced forms while interacting with the L2?: An online and offline study of the role of written input enhancement in L2 reading. *Hispania*, 496-509.

Li, L. (2016). *Sentence initial bundles in L2 thesis writing.* (Doctor of Philosophy), The University of Waikato,

Loper, E., & Bird, S. (2002). *NLTK: The natural language toolkit*. Paper presented at the The Second ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL.

Mackey, W. F. (1967). *Language Teaching Analysis*: Indiana University Press.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. New York: Cambridge University Press.

Marsh, D. (2002). *CLIL/EMILE-The European Dimension: Actions, Trends and Foresight Potential*. University of Jyväskylä, Finland: UniCOM, Continuing Education Centre.

McCarthy, M. (1990). *Vocabulary*: Oxford University Press.

McEnery, T., & Hardie, A. (2011). *Corpus linguistics: Method, theory and practice*: Cambridge University Press.

Meara, P. (1992). *EFL Vocabulary Tests*. Washington, D.C: Distributed by ERIC Clearinghouse.

Meara, P. (2010). *EFL Vocabulary Tests* (2nd ed.): Swansea: Lognostics.

Meara, P., & Buxton, B. (1987). An alternative to multiple choice vocabulary tests. *Language Testing, 4*(2), 142-154.

Meara, P., & Milton, J. (2003). *X-lex: the Swansea levels test*: Express Publishing.

Meara, P., & Miralpeix, I. (2015). V_YesNo v1. 0. In: Citeseer.

Meara, P., & Miralpeix, I. (2016). *Tools for researching vocabulary*: Multilingual Matters.

Meyer, C. F. (2002). *English corpus linguistics: An introduction*: Cambridge University Press.

Milne, D. (2010). *Applying wikipedia to interactive information retrieval.* (Doctor of Philosophy in Computer Science), The University of Waikato,

Milne, D., & Witten, I. H. (2013). An open-source toolkit for mining Wikipedia. *Artificial Intelligence, 194*, 222-239.

Milton, J. (2009). *Measuring Second Language Vocabulary Acquisition* (Vol. 45): Multilingual Matters.

Mochida, k., & Harrington, M. (2006). The Yes/No test as a measure of receptive vocabulary knowledge. *Language Testing, 23*(1), 73-98.

Musumeci, D. (1993). Content language learning: Symbiosis in the academe. *Language, Communication. and Social Meaning*, 147-157.

Nagy, W. E. (1995). *On the role of context in first-and second-language vocabulary learning*. Retrieved from

Nation, I. S. P. (1983). Testing and teaching vocabulary. *Guidelines, 5*(1), 12-25.

Nation, I. S. P. (1986). *Word Lists: Words, Affixes and Stems* (revised ed.). Wellington: Victoria University English Language Centre.

Nation, I. S. P. (1990). *Teaching and Learning Vocabulary*. New York: Newbury House.

Nation, I. S. P. (2001). *Learning Vocabulary in Another Language* England: Cambridge University Press.

Nation, I. S. P. (2013). *Learning Vocabulary in Another Language* (2nd ed.). England: Cambridge University Press.

Nation, I. S. P., & Beglar, D. (2007). A vocabulary size test. *The Language Teacher, 31*(7), 9-13.

Nation, I. S. P., Heatley, A., & Coxhead, A. (2002). Range: A program for the analysis of vocabulary in texts. Retrieved from https://www.victoria.ac.nz/lals/about/staff/paul-nation

Nattinger, J. R., & DeCarrico, J. S. (1992). *Lexical phrases and language teaching*: Oxford University Press.

Nesselhauf, N. (2004). What are collocations. *Phraseological units: Basic concepts and their application*, 1-21.

New, B., Pallier, C., Brysbaert, M., & Ferrand, L. (2004). Lexique 2: A new French lexical database. *Behavior Research Methods, Instruments and Computers, 36*(3), 516-524.

Nielsen, J. (1994). *Enhancing the explanatory power of usability heuristics*. Paper presented at the The SIGCHI Conference on Human Factors in Computing Systems, Boston, MA, USA.

Nordquist, R. (2018). Definition and Examples of Pseudowords. Retrieved from https://www.thoughtco.com/pseudoword-definition-1691549

Norvig, P. (2016). How to write a spelling corrector. Retrieved from http://norvig.com/spell-correct.html

Nunnally, J. C., & Bernstein, I. (1994). *Psychometric Theory* (Vol. 3). New York: McGraw-Hill.

O'Keeffe, A., & McCarthy, M. (2010). *The Routledge handbook of corpus linguistics*: Routledge.

REFERENCES

Oxford English Dictionary. (2018). Oxford University Press.

Palmer, H. E. (1917). *The Scientific Study and Teaching of Languages*. London: George G. Harrap and Company.

Pancheva, T., & Antov, P. (2017). *Application of content and language integrated learning (CLIL) in engineering education*. Paper presented at the XIXth International Scientific Conference, Yundola, Bulgaria.

Pellicer-Sánchez, A. (2017). Learning L2 collocations incidentally from reading. *Language Teaching Research, 21*(3), 381-402.

Pignot-Shahov, V. (2012). Measuring L2 receptive and productive vocabulary knowledge. *Language Studies Working Papers, 4*(1), 37-45.

Proença, J., Lopes, C., Tjalve, M., Stolcke, A., Candeias, S., & Perdigão, F. (2017). *Automatic evaluation of children reading aloud on sentences and pseudowords*. Paper presented at the Interspeech, Stockholm, Sweden.

Python Software Foundation. (2019). 7.4 difflib - Helpers for computing deltas. Retrieved from https://docs.python.org/2/library/difflib.html

Rastle, K., Harrington, J., & Coltheart, M. (2002). The ARC nonword database. *The Quarterly Journal of Experimental Psychology, 55*(4), 1339-1362.

Read, J. (1988). Measuring the vocabulary knowledge of second langauge learners. *RELC journal, 19*(2), 12-25.

Read, J. (1993). The development of a new measure of L2 vocabulary knowledge. *Language Testing, 10*(3), 355-371.

Read, J. (2000). *Assessing Vocabulary*. Cambridge: Cambridge University Press.

Read, J. (2007). Second language vocabulary assessment: Current practices and new directions. *International Journal of English Studies, 7*(2), 105-126. doi:10.1017/S0261444812000377

Richards, B., & Malvern, D. (2007). Validity and threats to the validity of vocabulary measurement. *Modelling and assessing vocabulary knowledge*, 79-92.

Rueckl, J. G., & Olds, E. M. (1993). When pseudowords acquire meaning: Effect of semantic associations on pseudoword repetition priming. *Journal of Experimental Psychology: Learning, Memory, Cognition*, *19*(3), 515.

Rumsey, D. J. (2019). How to compare two population proportions. Retrieved from https://www.dummies.com/education/math/statistics/how-to-compare-two-population-proportions/

Santorini, B. (1990). Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision). *Technical Reports (CIS)*, 570.

Schmidt, R. W. (1990). The role of consciousness in second language learning. *Applied Linguistics*, *11*(2), 129-158.

Schmidt, R. W. (2001). Attention. In P. Robinson (Ed.), *Cognition and Second Language Instruction* (pp. 3–32). New York: Cambridge University Press.

Schmitt, N., Jiang, X., & Grabe, W. (2011). The percentage of words known in a text and reading comprehension. *The Modern Language Journal*, *95*(1), 26-43.

Schwartz, S. (2013). *Measuring reading competence: A theoretical-prescriptive approach*: Springer Science & Business Media.

Short, D. J. (1991). *How to Integrate Language and Content Instruction: A Training Manual* (2nd ed.). Washington, DC: Center for Applied Linguistics.

Short, D. J. (1993). Assessing integrated language and content instruction. *TESOL Quarterly*, *27*(4), 627-656.

Short, D. J. (1994). Expanding middle school horizons: Integrating language, culture, and social studies. *TESOL Quarterly*, *28*(3), 581-608.

Sinclair, J. (1991). *Corpus, concordance, collocation*: Oxford University Press.

Snow, M. A. (1993). Discipline-based foreign language teaching: Implications from ESL/EFL. *Language and content: Discipline-and content-based approaches to language study*, 37-56.

Snow, M. A., Met, M., & Genesee, F. (1989). A conceptual framework for the integration of language and content in second/foreign language instruction. *TESOL Quarterly, 23*(2), 201-217.

Stoller, F. L., & Grabe, W. (1997). A six-T's approach to content-based instruction. *The content-based classroom: Perspectives on integrating language and content*, 78-94.

Straight, H. S. (1994). *Languages Across the Curriculum: Translation Perspectives VII*. Binghamton: State University of New York, Center for Research in Translation.

Sudermann, D. P., & Cisar, M. A. (1992). Foreign language across the curriculum: A critical appraisal. *The Modern Language Journal, 76*(3), 295-308.

Tedick, D. J., & Cammarata, L. (2010). Implementing content-based instruction: The CoBaLTT framework and resource center. In J. F. Davis (Ed.), *World Language Teacher Education: Transitions and Challenges in the 21st Century*. Greenwich, CT: Information Age Publishing.

The Review Team. (2018). The best MOOC platform. Retrieved from https://www.reviews.com/mooc-platforms/

Truscott, J. (1998). Noticing in second language acquisition: A critical review. *Second Language Research, 14*(2), 103-135.

Uggen, M. S. (2012). Reinvestigating the noticing function of output. *Language learning, 62*(2), 506-540.

van Zeeland, H., & Schmitt, N. (2012). Lexical coverage in L1 and L2 listening comprehension: The same or different from reading comprehension? *Applied Linguistics, 34*(4), 457-479.

Walton, M. (2016). Welcoming 3 million people to FutureLearn. Retrieved from https://about.futurelearn.com/blog/welcoming-3-million-people-to-futurelearn

Wesche, M. B. (1993). Discipline-based approaches to language study: Research issues and outcomes. *Language and content: Discipline-and content-based approaches to language study*, 57-82.

West, M. P. (1953). *A General Service List of English Words: With Semantic Frequencies and a Supplementary Word-list for the Writing of Popular Science and Technology*. London: Longmans.

Wikipedia. (2019). Wikipedia: Manual of style/lead section. Retrieved from https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Lead_section

Wiktionary. (2017). List of languages. Retrieved from https://en.wiktionary.org/wiki/Wiktionary:List_of_languages

Wiktionary. (2019). Wiktionary, the free dictionary. Retrieved from https://www.wiktionary.org/

Wu, S. (2010). *Supporting collocation learning*. (Doctor of Philosophy in Computer Science), The University of Waikato,

Wu, S., Li, L., Witten, I. H., & Yu, A. (2016). Constructing a Collocation Learning System from the Wikipedia Corpus. *International Journal of Computer-Assisted Language Learning and Teaching (IJCALLT), 6*(3), 18-35.

Wu, S., & Witten, I. H. (2007). *Content-based language learning in a digital library*. Paper presented at the International Conference on Asian Digital Libraries, Hanoi, Vietnam.

Xiao, R. (2010). Corpus creation. In N. Indurkhya & F. J. Damerau (Eds.), *Handbook of Natural Language Processing* (2nd revised ed., pp. 147-165).

Appendix A. The DMwW wordlist

This appendix contains the list of 571 words present in the *DMwW wordlist*, a domain-specific wordlist generated from the *DMwW corpus*. The creation of the wordlist is described in Section 3.7.

absolute	assess	category	computer	datasource	discriminate
access	assign	challenge	concept	decision	display
according	associate	characteristic	conclusion	default	distinct
accurate	assume	chartfactory	configure	define	distribute
achieve	assumption	chemistry	confirm	definite	diverse
acid	attribute	choice	consist	definitive	document
active	auc	circumstance	console	delete	domain
adapt	automatic	classic	constant	demography	download
addcluster	available	classification	construct	demonstrate	downloaded
adjust	axis	classifier	consult	dense	drift
adwin	baseline	classify	context	deploy	economy
affect	basic	cleave	contribute	derive	edit
aggregate	basis	click	convenient	descent	effective
airline	batch	closed	converge	design	element
algorithm	bayes	cluster	convert	desktop	eliminate
alternative	bayesnet	clusterers	convey	detect	embed
amaze	benchmark	cobweb	core	deteriorate	enable
amino	bestfirst	code	correlate	deviate	encounter
analyse	bias	coefficient	correspond	device	engineer
anneal	bin	column	couple	diabetes	ensemble
anonymized	binary	comma	cpu	diagram	ensure
apache	bioinformatics	comment	create	dialogue	entropy
approach	boost	communicate	criteria	different	environment
appropriate	bracket	community	csv	dimension	equipment
approximate	breast	competition	customer	directed	equivalent
apriori	brief	complex	customise	discrepancy	error
arbitrary	bubble	complicate	cycle	discretization	especially
area	calibrate	component	data	discretize	essential
arff	capable	comprehensive	dataset	discretized	estimate
array	carbon	comprise	datasets	discretizing	ethical

ethics
evaluate
evidence
evolve
exclude
execute
expensive
expert
explicit
extensive
extract
facility
factor
fair
fastICA
feature
feedback
file
filename
filter
final
five
focus
folder
forecast
format
formula
fortunate
framework
frequency
frustrate
function
gamma
generate
generator
ggplot
global
goal
goodbye
google
gradient
graph
graphic
gridsearch
groovy
GUI
hadoop
header
headlamp
hello
histogram
hoeffding
hoeffdingtree
holdout
horizontal
however
huge
humid
hyperplane
hypothesis
hypothyroidism
ibk
identical
identify
ignore
illustrate
image
imagefilters
implement
import
impress
incorporate
increment
index
indicate
individual
infinite
informative
infrared
initial
input
install
instance
instruct
integer
interact
interface
internal
internet
interpolate
interpret
investigate
invoke
involve
ionosphere
iris
issue
item
iterate
iteratively
java
javadoc
job
JRIP
JVM
jython
kappa
kernel
knowledge
label
lag
layer
legal
liblinear
libSVM
licence
linear
link
linux
locate
logistics
loop
loss
lower
mac
magic
majority
makebinary
manual
margin
massive
mathematics
matrix
maximum
maybe
mechanism
menu
meta
method
metre
milligram
millilitre
million
miner
minimise
minimum
minor
misclassificatio
ns
misclassified
MOA
mode
modify
module
MRI
multifilter
multilayer
multinomial
multiple
naive
naivebayes
negative
network
neural
neuroimaging
neuron
newdate
news
nitrogen
node
nominal
normal
null
num
numeric
obtain
obvious
occur
ok
oneR
online
operate
optimal
optimise
option
organic
original
outcome
outlook
outlying
outperform
output
outputting
overall
overcast
overfit
overfitted
overfitting
overlap
overlay
overwhelm
panel
parallel
parameter
partition
partner
passenger
pdf
peak
peptide
percent
perceptron
perceptrons
period
pitfall
plas
plot
plugin
plus
polygon
positive
potential
practise
precise
predict
preprocess
preprocessing
prequential
previous
principal
principle
prior
prism
privacy
probably
procedure
process
proportion
protein
prune
python
query
quiz
quote
R
random
range
ranker
rdata
realise
really
recall
recognition
rectangle
recursive
redundant
region
regress
reject
relate
relative
release
relevant
rely
remove
replicate
require
resample
research
resource
respective
restore
restrict
retain
retrieve
reveal
revolution
rferns
ripper
ROC
RPart
rplugin
scenario
sceptic
scheme
script
scroll
seconds
section
seek
segment
select
sensible
sensitive
sentiment
sepalwidth
sequence
series
setosa
setup
sex
shortcut
shuffle
significance
significant
similar
simplicity
site
skip
SMO
SMOreg
software
solution
somewhat
sophisticated
sorted
source
spark
sparse
specific
specify
spreadsheet
stable
stack
statistic
stochastic
strategy
stratify
structure
substantial
success
sufficient
sum
summary
supermarket
supervise
svms
switch
tab
target
task
team
technical
technique
technology
tedious
template
text
textviewer
theorem
theoretical
three
threshold
tigerjython
tiny
topic
tradeoff
transform
trivial
tweet

twitter
typical
ultimate
unpruned
update
updateable
url
validate
vary
vector
verify
versicolor
version
versus
vertical
video
virginica
virtual
visual
visualizer
volume
voxels
web
weight
weird
weka
whereas
workbench
xmeans
xml
xrff
zeroR

Appendix B. Pseudowords

This appendix includes three collections of pseudowords: (1) the general-purpose pseudowords generated using the CGCA algorithm, (2) the pseudowords generated using external sources, and (3) the domain-specific DMwW pseudowords generated using the CGCA algorithm and the DMwW wordlist.

B.1. CGCA pseudowords

The following eight lists contains 100 pseudowords each, generated using the Character-gram Chaining Algorithm with 2-grams to 8-grams and r-grams. Their creation is outlined in Section 4.5.

2-grams

acive, ahes, ame, appin, aqual, ath, aud, aunie, axi, barcull, bic, ble, bre, cens, cinhes, cle, clutro, cral, cre, culk, cyclab, des, duch, dwabs, dyinate, dyntin, effin, eigh, eiguit, eles, elin, elize, epid, epin, epist, eving, excate, faugh, fred, gatin, ghol, gymma, gympart, hal, hummo, ign, irrat, irre, islas, ite, jous, kide, knia, knon, lyist, mism, motin, mudgin, nes, nigit, occel, oddeful, opil, ove, parce, pors, pyrie, quicass, quiciplue, scon, sexage, skage, ske, sle, slutio, smen, snatio, snegule, snemin, snutin, spie, stro, sunt, tbscrap, uglon, ving, vois, volin, wal, wate, weat, wicate, wilock, wrat, wrer, yart, yied, yies, zer, zon

3-grams

addream, aested, alars, alphise, ambat, animarrograduatio, apted, aquad, astreal, badle, begian, bencircula, bordle, bount, boximidate, bubbording, captime, civison, codier, coherind, cologist, cree, cusher, custer, dange, dashapped, datio, dioxing, direciarian, dotten, droposses, drous, dward, elses,

escal, eter, exhausive, faerint, ferray, feveal, filment, frust, fusin, gend, goddest, grunk, hotes, husbanimagic, hydraft, hydraw, ingly, islatic, ivor, jockade, launding, leist, melop, metributten, natigating, neoccupy, nette, normist, nutrance, obed, orand, ough, ouncing, petitigatio, phily, psychor, pulsifiably, punit, rading, recollusted, rect, reuniour, runnius, secur, segrough, situte, spher, stres, stroding, teasin, tidinella, tisspoil, toeinvey, toppinnie, totate, tremarrass, truct, upgram, vaguidined, valist, vitat, waggest, wilden, witle, wondness, wroning

4-grams

acknowier, aller, ammunization, basical, bassettle, becombination, beform, bipollution, boweller, castic, cathon, certake, clergency, clinist, clustrian, congregars, cosing, cripping, dailing, dairing, degret, demogram, dension, downess, encircuit, enging, epidest, estation, evalue, faintain, farewed, fictim, flockage, gelligencing, gunnier, helicit, horsement, hostily, incling, industract, inflatting, insing, irration, jokerage, lacies, majestive, malendable, mamming, manifes, manufactor, mathes, merritor, militic, misher, misnamic, nannish, neoclaim, nobilize, outral, parise, pondence, prograph, puddiness, reorganic, requent, riskies, rollegal, scepted, screet, sevent, shatten, sinced, spation, sphers, subles, subrouts, summars, supplicise, sweaten, syndrous, tamin, targemending, tennia, tonnect, truise, unence, unfemining, unglamour, unhurricula, unidirect, unilluster, unnumbent, unpolisation, unscious, untoleration, vertion, viction, voiceleranch, waginable, wastic

5-grams

acquisiting, antitation, barrely, basicals, beadied, besidence, butched, cholesaling, conferent, daughtier, daughtieth, demograph, desertive, disably, disturise, easters, especifies, essentee, eventure, flocker, franting, geograph, gesturism, hanger, heritancy, housins, hurlines, impactness, impatibly, impraction, incorrelation, indelication, indiscriminal, injusting, inoffensible, insistible, invaried, irrecover, irreplace, jogginess, kindlines, labellious, laughting, lazines, lingery, manlined, masculing, metropologies, minist, misundering, nobodied, noncline, nonsent, nonspect, novelines, orches, overses, pestical, phantable, poststruct, recrease, registract, rehabilise, restinative, restle, shrubbers, simultant, subrouting, sufficial, surgeois, suscepting, testablish, unaccessful, unanimation, uncompromise, uncultural, undeliver, undeservoir, unelaboration, unenthusiast, unhappie, unherals, unhinders, unincorporate, unlabelling, unmonitor, unopposes, unorganize, unpatriot, unpollutant, unsation, unscend, unsecute, unsenting, unsignifies, unspect, untalentleman, waterial, wheated, wordinarily

6-grams

accountabilise, adaptabilise, ammunity, antipolluted, artifically, artilleried, bidirection, bindering, browness, centuring, clickering, conscientific, curatories, deferenda, descripted, dreamery, durabilitation, durabilitative, elicities, encompatibility, flourist, illiberation, inattendee, inconvention, incorrupting, indefined, indestruct, informance, inhumans, instabilise, insubordinariness, invulnerabilitation, irrespection, millingness, misapprove, misapproving, neoclassification, nonclining, orderline, pitchery, reasonal, rehability, repressant, resolutism, retainee, scription, sentimetre, slendered, sterness, subrouting, subsequencies, traininess,

unachievably, unannounce, unappreciate, unattribute, uncaptures, uncategorise, uncategorising, uncelebrating, uncirculate, uncirculation, uncollect, uncommittal, uncoordinate, underness, undiagnose, undiminish, unemotionless, unfashions, unhesitation, unimportation, uninfection, uninities, uninsures, unjustification, unmaintain, unmediate, unmentions, unobstruct, unoccupier, unorganize, unpreventation, unpunisher, unquestion, unreactivate, unreflection, unrefresher, unreservist, unsanction, unscript, unsentiment, unsilencer, unstruct, unsuccessor, untestant, unwrinkler, visitating, watchery, wearably

7-grams

allegians, auctionism, autobiographing, bidirection, chickener, dauntings, deferencing, disapplicable, disapplicating, herbalisation, herbalising, immunition, implicities, improbabilistic, incompetition, incomprehend, indestructural, indetermine, indisciplinary, indistinguish, inexpressed, insignification, insusceptibilities, interdiscipline, intergenerative, justificate, momentaries, nondemocratization, orderlines, practitioned, preregistrative, resignating, semanticising, sequenting, simplicitness, switchiness, tacticalities, tacticality, unaccompanist, unaccountant, unanticipating, unapproaching, uncelebrate, unceremonies, uncertificate, unchallenge, uncirculate, uncirculation, uncollective, uncommunicator, uncompetition, uncontroller, uncoordinator, uncultivate, uncultivating, uncultures, undemonstrator, undependence, undependent, undesirabilities, undeterminist, undiagnosis, undifference, undischarge, undocumentary, unelaborating, unenlighten, unentertain, unenthusiast, unexception, unidentifically, unidentificate, unidentifier, unillustrate, unillustrating, uninsulas, unmaintainability, unmitigates, unobjection, unoccupies, unorganic, unpopulates, unprecedence, unpresentation, unpresented, unpresenting, unprocesses, unprocessing, unprofession, unpronounce, unpublisher, unqualifier, unreconstruct, unrefresher, unsentiment, unstruction, unsupervise, unsupervisor, untranslate, youthfuls

8-grams

aerialistic, bidirection, conveniencing, counterproduction, implementary, improbabilistic, inaccessibilities, inappropriating, incomprehend, inconsiders, incorporatism, indestruction, indetermines, indisciplining, inferentiate, inferentiation, insignification, insusceptibilities, interdepartment, intergovernment, laborator, nondemocratisation, poststructure, poststructuring, reaffirmative, reintroductory, representably, resignating, semanticise, unaccompanies, unaccompaniment, unaccompanist, unanticipate, unappreciate, unattributes, unauthorises, uncategorisation, uncategorise, uncategorization, uncertificate, uncertification, uncharacterise, uncharacterising, uncirculates, unclassification, unclassifies, uncommittee, uncommunicate, uncompetition, uncomprehend, unconforming, unconsolidate, unconsolidation, unconstitute, unconstitution, uncontaminate, uncontaminating, uncontamination, unconvention, uncoordinate, undeterminant, undiagnoses, undifference, undiminishes, undischarger, undisciplinary, undiscoverer, undistinguish, unentertains, unenthusiast, unexceptions, unidentifier, unidentifies, unidirection, unillustrate, unillustration, unillustrator, unimagination, unincorporate, uninfluences, uninterruption, unorganizes, unprejudices, unpresentation, unpresentative, unprofession,

unpronounce, unpronouncing, unreconstruct, unrepresentation, unrepresents, unresponsibility, unresponsibly, unrestriction, unsophisticating, unstandardise, unstandardize, unstandardizing, unstructure, untranslates

r-grams

agonier, alternationalist, anter, apartmentalizing, arrogatories, aspectacularly, automobilising, becorate, braveller, cancellings, casualtiest, ceasiness, cresterning, decorationist, disbehaviour, dispensate, drinkage, droughness, dustering, eigh, eightist, evolutionaries, feminiscent, fibroutines, frigerational, gallenging, gotters, gottom, greeness, grimmers, hastel, histor, iconist, inauthenticate, incontestant, intesting, irrelevising, iss, jointingly, junctionality, keeness, kniversalising, laborator, lasingly, lettes, lumpetent, magistrative, mishearten, mispronourable, neithesizer, nervouring, ninthood, notablishment, oddditional, oldier, pavincidences, platefully, plaust, poolininity, povern, probabilitation, rathematically, reservationism, rhydrate, scarcelled, slammation, slappie, slightfulness, spillager, stal, starlessly, subscriptional, takeries, targements, teamily, therapeuticals, toleratives, tomators, treelessly, tutorializing, unblocketing, uncontaminates, undiminisher, unelaborateness, unexception, unfriendless, unillustrating, unkness, unobjectional, unpopulation, unreliabilitation, unseeder, unsparent, unstandardization, untest, unthreatens, virginitialise, weddie, whitecturalisation, zer

B.2. External pseudowords

The following five lists contains 100 pseudowords each from: the English Lexicon Project, the ARC Non-word Database, WordGen, Wuggy, and Meara. Their creation is outlined in Section 4.7.1.

The English Lexicon Project

ansorbed, ashigned, asjoins, asolished, aubiences, aubited, banaligy, banglos, blaymate, bonspire, capturong, chianri, chrives, clevelant, clotillas, creemason, curched, drimaced, drimstone, egocative, egualled, emolliest, fensive, filuted, flansman, freemar's, gactics, galvespon, gastrami, geaceable, gebanese, guinbess, gultitude, heartek, holitaire, holitudes, homrades, hotanical, inlards, irduction, italicibe, keepint, kittons, knuckred, lageantry, lambist, lambslin, larmony, leaspoon, lecurity, libersy, liburally, lurdled, madenna, moltages, mostrils, neafiest, nightkine, nonclude, oaklind, opioles, pabulate, palivary, parpoon, patehouse, peatwave, pepotism, phatches, phemists, pixiluted, priesh's, ranvassed, revipal, roldness, rollect, rublery, selectove, selodious, semocrat, semorial, simmied, simuloted, smaffing, soloust, sommence, sonehead, spagnate, struced, sustpan, teagued, thlorides, tources, triflong, trimson, trushes, turitan, videshow, virector, wastors, windfard

The ARC Non-word Database

blid, bloap, blogs, bramb, broon, chiln, clat, clulled, craint, craste, creats, crenched, dwat, dweighth, eamed, elled, fant, frim, fris, gerf, ghenced, gir, glalks, glerps, glozed, grear, gretts, grev, grings, gud, gwibed, gwurs, hoved, jarled, kaks, keezed, kerm, kisp, knoists, lepth, luilt, lulped, mirds, mun,

neagued, nurk, nus, olks, phralk, phroin, plurps, preered, prirque, prirr, rhalf, scrocked, scroft, shreef, shrort, shroths, shruise, shrune, shrusks, skacs, skenes, sledes, sloars, slucked, smarge, smossed, smurged, spails, spriz, sprymphs, spuilt, squiche, squite, stetch, strak, strangs, strebb, stulb, stules, swawls, swoursed, teague, theened, thrarse, thwaints, thwibbed, thwiped, trerd, trerts, tuss, twamp, tweggs, vaphed, whem, yause, zoc

WordGen

adoke, advero, applk, awig, baiman, beant, bery, biled, boni, bouice, brede, brft, brozer, brug, busty, carher, cariet, catar, charte, cheed, cluch, coefer, daney, debage, deiry, diff, dohey, doma, drll, drrdge, ecit, eman, embey, eurn, fangle, flort, floter, flum, fraze, freedy, gerk, gien, gimpet, goef, golto, grrpe, heager, hila, hirt, hoory, iraise, javy, joat, kocker, kolly, lars, leff, lineap, loeer, lorus, lous, mailen, maiman, mastie, meady, moay, mondly, mynger, olfish, olto, opin, ostlet, oves, penky, pewl, piet, pixes, ragio, rair, refide, roox, rumer, salls, saming, sappet, scurty, seaped, shnrk, shwrk, sodo, spel, tasil, teiple, thindy, tiexon, tinse, toint, vanid, vocate, yaen

Wuggy

abvonote, agimmed, ahs, awop, baff, beller, blive, bourd, buit, bur, burd, cloke, comfake, corraty, dag, deyityte, doop, dre, erv, eskite, euprer, extrail, farf, fasp, fave, fims, fordly, fovearint, gar, grale, guk, habs, hact, hammy, hamp, harf, heer, henshed, hoins, hu, hugh, hunets, hust, ider, imeyits, insows, istye, kenchen, lagle, lak, lale, lanish, lonten, lut, mab, marter, mistive, mither, moifs, mully, naw, onsce, oot, owd, pasp, pliedes, polyll, proment, purs, quother, rangle, recure, reesick, rews, ro, rooble, roor, roud, roule, saturcip, sheat, shie, siffs, snat, sork, soth, spigay, stell, stronk, stubant, sussest, swant, touth, uffive, uscicer, woins, woll, woubt, ymn, zesa

Meara

abrogative, acklon, adair, ager, ager, aistrope, almanical, baldock, balfour, bance, bastionate, batcock, benevolate, berrow, bodelate, buttle, cambule, cantileen, channing, charactal, charlett, combustulate, condimented, connery, contortal, contrivial, degate, descript, detailoring, dogmatile, dowrick, draconite, duffin, eckett, eldred, eluctant, fluctual, galpin, glandle, gumm, gummer, hapgood, haque, hemiaphrodite, homoglyph, horobin, horozone, hoult, hubbard, humberoid, jarvis, justal, kiley, lannery, lapidoscope, lauder, limidate, litholect, loveridge, menstruable, misabrogate, moffat, mundy, nickling, nonagrate, oestrogeny, oestrogeny, oligation, opie, overend, oxylate, pauling, pernicate, pocock, pring, quorant, ralling, recenticle, reservory, retrogradient, ridout, rudge, scudamore, scurrilise, snell, stace, stimulcrate, suddery, tooley, troake, trudgeon, twose, venn, vickery, webbert, whaley, whitrow, wilding, woolnough, wray

B.3. The DMwW pseudowords

The following four lists contains 100 pseudowords each, generated using the Character-gram Chaining Algorithm with 2-grams, 3-grams, 4-grams and r-grams. Their creation is outlined in Section 4.8.

APPENDIX B PSEUDOWORDS

2-grams

abstandout, abstipload, affdive, aggle, altegy, altic, apal, appror, attruct, bubsocept, caph, cirlablicant, defilimisifinite, distrare, ecorm, ecort, effdion, effilics, elenclate, erreque, ethort, ethyroximent, evion, exple, filtimat, fince, flosis, foracify, frangin, fratinal, frechassion, freconvoke, fruccedit, fruct, funal, funcy, goact, gratiminform, hadify, heava, holve, hypothyrophic, ign, illel, illevach, imal, imporm, irient, irivial, ite, kerfile, kerpote, knomproogure, knosplar, mect, modise, negal, nult, obtailassitive, obtane, occespothistive, onlicate, onlow, optial, optinate, optize, optrane, ori, ove, pargin, pary, pite, plospechnosed, proach, revaloy, sceselel, shon, solume, soundate, sper, strative, strunate, subble, sume, sumstive, swin, switfact, tassign, thely, tincify, tiniting, trix, typicatial, valedual, valibutic, valinal, vistassear, vold, vology, weigin

3-grams

absole, absolumn, accur, algority, approport, aprior, attrincipal, attrix, baset, bayer, bioinform, chally, circumeristic, clustrate, coefficance, coefficit, comple, comprest, dataseline, demove, deplicient, deriod, detes, diabel, distimise, dively, domate, econosphere, enably, enginear, especify, essentifier, estic, estimilar, evalent, feate, frequipmension, frequipment, frequire, frequivacy, functise, functive, hadoc, hypermal, idence, increprocedure, incretize, instantiment, instry, iterplate, jythod, kernate, lible, locatify, logistigate, logistinomial, majorithm, mathemate, mathemative, mathesign, matribute, matrinct, methon, miniminatic, miscrepance, miscretize, miscrimilayer, multilar, normat, operall, outperfit, passential, passentimise, passific, peption, percename, positial, potensive, redure, regregal, scept, schemistinomy, seconosphere, selevant, serift, shortual, spectance, spectistic, substange, suffle, summarket, switter, tedict, theorected, threst, trivacy, typicance, vectangle, vidence, vidential

4-grams

algority, alternal, alternel, alternet, apriorate, bioinformal, bioinformat, categy, challel, characteriod, characteriorate, characteriori, characteristinct, charactise, chemistribute, chemistrict, classificance, classificant, clustrate, clustratify, coeffding, colume, communical, constrate, decise, definitial, demonstruct, deteriori, deteristic, dimensible, dimensitive, diment, distic, distry, eliminal, especific, especify, espective, essenger, essentify, essentimensible, expensitive, extensitive, frameter, frustratify, generatify, hoefficient, idential, identimension, identimensitive, imprehension, imprehensitive, imprehensive, informat, initive, instrate, instratify, internative, iteract, kernet, logistinct, majorithm, mathemative, misclassic, misclassificance, misclassificant, misclassifier, normat, operatify, parallenge, paramework, passential, passentify, perceptide, periorate, periori, potentify, potentimensive, potentiment, practeriod, practeriorate, practeristic, prequence, prequency, replicit, respecially, respecific, restribute, sceptide, sceptron, sension, sential, sentify, simplicit, sophistinct, statistinct, strate, strative, technicate, volumn

r-grams

aminative, annear, approprior, apriate, aprid, assification, assified, assifier, bayer, binario, bioinformative, bries, challename, clustomatic, clustrategory, colume, competity, constrate,

datassify, delect, demonstruct, deristicated, dimensible, distristic, drify, ecorporate, edisclassifier, elimilar, encially, enginear, equivalement, errespond, essenger, execursive, expensitive, facipal, filenge, filternative, flogistics, frameworrespond, frustratively, globable, googy, helative, humeric, ign, impreshold, initive, irio, iteract, iterpolate, jave, legative, logisticated, logistramework, majorithm, miniminate, misclassifier, miscriminate, nain, nomino, normatic, obvioinformative, occurach, occurate, optimilar, origistigate, peptron, peries, periorate, plogisticated, practeristic, prequence, propriate, quivalenge, quivalyse, replicit, resphere, ripment, sential, sentify, serief, serive, shoidual, strate, substall, summunicate, switter, technologue, theorecision, tinformative, triorate, twittribute, typothesis, vected, vidency, videntify, volumn, voxelative, workbetition

Appendix C. Receptive vocabulary tests

This appendix includes three versions of the EFL Vocabulary Test. The first contains five versions of the original test, one for each level. The second contains five versions of the AEFL test, a recreation of the original test that was generated using the AEFL algorithm. The third is a domain-specific DMwW version of the AEFL test, generated using the AEFL algorithm and the DMwW wordlist.

C.1. The EFL vocabulary test

EFL Level 1:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	☐	bridge	2	☐	modern	3	☐	curtain
4	☐	prison	5	☐	classinate	6	☐	mascarate
7	☐	engine	8	☐	hurt	9	☐	ugly
10	☐	moon	11	☐	aspection	12	☐	expedalize
13	☐	patient	14	☐	explain	15	☐	also
16	☐	hallett	17	☐	postherent	18	☐	shake
19	☐	shell	20	☐	forest	21	☐	warm
22	☐	govern	23	☐	next	24	☐	berrow
25	☐	cymballic	26	☐	feature	27	☐	street
28	☐	person	29	☐	speak	30	☐	absalom
31	☐	lowry	32	☐	murtagh	33	☐	copy
34	☐	rickard	35	☐	tax	36	☐	portingale
37	☐	bite	38	☐	mad	39	☐	rice
40	☐	circle	41	☐	lie	42	☐	half
43	☐	bad	44	☐	hapgood	45	☐	suddery
46	☐	attard	47	☐	trousers	48	☐	row
49	☐	camera	50	☐	day	51	☐	harmonical
52	☐	plebocrat	53	☐	catling	54	☐	earn
55	☐	private	56	☐	test	57	☐	lazy
58	☐	kill	59	☐	size	60	☐	retrogradient

H: FA: Dm:

Answer PM112: 5 6 11 12 16 17 24 25 30 31 32 34 36 44 45 46 51 52 53 60

EFL Level 2:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	❑	cantileen	2	❑	claim	3	❑	bring about
4	❑	rotate	5	❑	batcock	6	❑	imaginary
7	❑	trudgeon	8	❑	astell	9	❑	influence
10	❑	vein	11	❑	sign	12	❑	effectory
13	❑	invention	14	❑	state	15	❑	planet
16	❑	claypole	17	❑	darkness	18	❑	ray
19	❑	annual	20	❑	negalogue	21	❑	background
22	❑	gift	23	❑	satisfy	24	❑	richings
25	❑	absorb	26	❑	gillen	27	❑	level
28	❑	devise	29	❑	pardoe	30	❑	blame
31	❑	emotion	32	❑	exhaust	33	❑	spin
34	❑	take	35	❑	particular	36	❑	fraction
37	❑	correspond	38	❑	already	39	❑	guide
40	❑	cease	41	❑	ashill	42	❑	frequid
43	❑	relationship	44	❑	pestulant	45	❑	oblige
46	❑	negative	47	❑	hobrow	48	❑	inertible
49	❑	confident	50	❑	universe	51	❑	military
52	❑	gallimore	53	❑	division	54	❑	topic
55	❑	product	56	❑	surman	57	❑	algoric
58	❑	chicorate	59	❑	figure	60	❑	reservory

H: FA: Dm:

Answer PM212: 1 5 7 8 12 16 20 24 26 29 41 42 44 47 48 52 56 57 58 60

EFL Level 3:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	☐	average	2	☐	ionopose	3	☐	whisper
4	☐	seward	5	☐	coal	6	☐	lucky
7	☐	occasion	8	☐	wages	9	☐	razor
10	☐	spare	11	☐	fountain	12	☐	lip
13	☐	personal	14	☐	nail	15	☐	propose
16	☐	suitable	17	☐	dormatize	18	☐	go off
19	☐	cage	20	☐	frequent	21	☐	whitrow
22	☐	disturb	23	☐	shame	24	☐	stace
25	☐	postpone	26	☐	crazy	27	☐	blanket
28	☐	term	29	☐	clear up	30	☐	craddock
31	☐	member	32	☐	boobier	33	☐	advise
34	☐	prison	35	☐	collexis	36	☐	independent
37	☐	harbour	38	☐	amey	39	☐	galeology
40	☐	briochery	41	☐	ladder	42	☐	fan
43	☐	stir	44	☐	dust	45	☐	pan
46	☐	aloud	47	☐	overend	48	☐	slide
49	☐	touch	50	☐	direction	51	☐	venn
52	☐	repreach	53	☐	request	54	☐	obliquate
55	☐	scout	56	☐	lavery	57	☐	kerkin
58	☐	wolliner	59	☐	impelirous	60	☐	temerify

H: FA: Dm:

Answer PM320: 2 4 17 21 24 30 32 35 38 39 40 47 51 52 54 56 57 58 59 60

EFL Level 4:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	❑	crisis	2	❑	concentrate	3	❑	devil
4	❑	identify	5	❑	operate	6	❑	lunarous
7	❑	devoidance	8	❑	fraternism	9	❑	keep back
10	❑	northern	11	❑	supernumerate	12	❑	issue
13	❑	resource	14	❑	literary	15	❑	punishment
16	❑	citizen	17	❑	frontier	18	❑	randle
19	❑	world-wide	20	❑	overcome	21	❑	absent
22	❑	twose	23	❑	feel up to	24	❑	waterproof
25	❑	bring in	26	❑	stock	27	❑	batstone
28	❑	youde	29	❑	tide	30	❑	splendid
31	❑	ankle	32	❑	reconciliant	33	❑	walter
34	❑	sightseeing	35	❑	decorite	36	❑	cassette
37	❑	triangle	38	❑	decrease	39	❑	ewing
40	❑	event	41	❑	tudball	42	❑	hold out
43	❑	raincoat	44	❑	gossip	45	❑	minor
46	❑	route	47	❑	pinkard	48	❑	porter
49	❑	mollific	50	❑	bore	51	❑	technical
52	❑	wookey	53	❑	defunctionary	54	❑	lowry
55	❑	outhold	56	❑	variety	57	❑	shopkeeper
58	❑	allard	59	❑	machinery	60	❑	dumb

H: FA: Dm:

Answer PM420: 6 7 8 11 18 22 27 28 32 33 35 39 41 47 49 52 53 54 55 58

EFL Level 5:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	☐	ventrice	2	☐	jam	3	☐	pendulate
4	☐	label	5	☐	straighten	6	☐	multiply
7	☐	merciless	8	☐	kitely	9	☐	mastiphitis
10	☐	wear off	11	☐	awkward	12	☐	astle
13	☐	skemp	14	☐	hume	15	☐	sincerity
16	☐	warrender	17	☐	fishlock	18	☐	temptation
19	☐	hearn	20	☐	pudding	21	☐	cabbage
22	☐	rust	23	☐	chicorate	24	☐	bench
25	☐	equality	26	☐	pragmadict	27	☐	crockery
28	☐	hook	29	☐	grease	30	☐	madness
31	☐	mollet	32	☐	flautism	33	☐	sportsman
34	☐	misery	35	☐	realise	36	☐	owing to
37	☐	deer	38	☐	flap	39	☐	fright
40	☐	universe	41	☐	title	42	☐	adequate
43	☐	postal	44	☐	aside	45	☐	fantastic
46	☐	overall	47	☐	catalogue	48	☐	syllogasm
49	☐	royle	50	☐	clockwork	51	☐	barrate
52	☐	separate	53	☐	liar	54	☐	depositionary
55	☐	angloprole	56	☐	peck	57	☐	trick
58	☐	downwards	59	☐	coloniate	60	☐	reservation

H: FA: Dm:

Answer PM508: 1 3 8 9 12 13 14 16 17 19 23 26 31 32 48 49 51 54 55 59

C.2. The AEFL vocabulary test

AEFL Level 1:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	❑	cover	2	❑	pale	3	❑	untrol
4	❑	pretend	5	❑	paint	6	❑	yellen
7	❑	terriend	8	❑	study	9	❑	kindly
10	❑	wrong	11	❑	best	12	❑	loved
13	❑	jumperime	14	❑	tap	15	❑	maineer
16	❑	cause	17	❑	thank	18	❑	rose
19	❑	rainy	20	❑	put	21	❑	met
22	❑	piece	23	❑	yestric	24	❑	coung
25	❑	fork	26	❑	busin	27	❑	light
28	❑	date	29	❑	break	30	❑	egg
31	❑	reach	32	❑	solid	33	❑	danch
34	❑	regun	35	❑	skin	36	❑	corrow
37	❑	fairly	38	❑	practised	39	❑	space
40	❑	trience	41	❑	back	42	❑	lie
43	❑	flew	44	❑	lenger	45	❑	seriously
46	❑	dreamt	47	❑	dance	48	❑	woman
49	❑	lavate	50	❑	brint	51	❑	unsuccessfully
52	❑	posite	53	❑	fiercely	54	❑	noisily
55	❑	sistmas	56	❑	stairs	57	❑	tirect
58	❑	practise	59	❑	furned	60	❑	villow

H: FA: Dm:

Answer PG101: 3 6 7 13 15 23 24 26 33 34 36 40 44 49 50 52 55 57 59 60

AEFL Level 2:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	☐	alreat	2	☐	worth	3	☐	official
4	☐	cooperate	5	☐	availy	6	☐	evaporal
7	☐	intend	8	☐	belop	9	☐	afterwards
10	☐	communicate	11	☐	adequent	12	☐	hearch
13	☐	sevel	14	☐	need	15	☐	oppose
16	☐	publight	17	☐	hurresport	18	☐	sight
19	☐	region	20	☐	rhythm	21	☐	supporal
22	☐	feather	23	☐	point	24	☐	energy
25	☐	set	26	☐	fundamental	27	☐	afford
28	☐	spoil	29	☐	neglect	30	☐	suppoint
31	☐	pilot	32	☐	owner	33	☐	overcome
34	☐	undergo	35	☐	search	36	☐	substand
37	☐	department	38	☐	frontier	39	☐	desire
40	☐	approach	41	☐	satio	42	☐	sensident
43	☐	profession	44	☐	especially	45	☐	royage
46	☐	lock	47	☐	abroad	48	☐	cemedy
49	☐	sort	50	☐	moistorizon	51	☐	anxious
52	☐	circumstances	53	☐	elect	54	☐	get
55	☐	elassition	56	☐	item	57	☐	volume
58	☐	busion	59	☐	orden	60	☐	land

H: FA: Dm:

Answer PG201: 1 5 6 8 11 12 13 16 17 21 30 36 41 42 45 48 50 55 58 59

AEFL Level 3:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO.

1	❑	subtract	2	❑	actually	3	❑	admit
4	❑	postep	5	❑	multy	6	❑	realth
7	❑	cemend	8	❑	tape	9	❑	contents
10	❑	spoil	11	❑	battence	12	❑	tail
13	❑	slight	14	❑	nail	15	❑	habit
16	❑	encourage	17	❑	grateful	18	❑	request
19	❑	rapitable	20	❑	founcturb	21	❑	gentle
22	❑	relation	23	❑	unnecent	24	❑	owner
25	❑	wing	26	❑	drain	27	❑	practical
28	❑	peacup	29	❑	lovely	30	❑	lump
31	❑	sale	32	❑	worrength	33	❑	uppeaker
34	❑	subject	35	❑	cuship	36	❑	nepher
37	❑	lookup	38	❑	rapite	39	❑	population
40	❑	examination	41	❑	attend	42	❑	passary
43	❑	pale	44	❑	electrical	45	❑	progress
46	❑	effort	47	❑	garage	48	❑	profession
49	❑	astomestic	50	❑	wondent	51	❑	situry
52	❑	wire	53	❑	wages	54	❑	feath
55	❑	persuade	56	❑	influence	57	❑	brict
58	❑	cartr	59	❑	width	60	❑	snake

H: FA: Dm:

Answer PG301: 4 5 6 7 11 19 20 23 28 32 33 35 36 38 42 49 50 51 54 57

AEFL Level 4:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	☐	credit	2	☐	revise	3	☐	deterpresearch
4	☐	outline	5	☐	operate	6	☐	econstal
7	☐	dine	8	☐	corress	9	☐	scenefinimum
10	☐	liner	11	☐	tire	12	☐	haircase
13	☐	trace	14	☐	appointment	15	☐	impath
16	☐	wherever	17	☐	effect	18	☐	anger
19	☐	tiger	20	☐	curse	21	☐	splendid
22	☐	construct	23	☐	stationery	24	☐	task
25	☐	microphone	26	☐	convenience	27	☐	pronunciation
28	☐	untincome	29	☐	humble	30	☐	tempt
31	☐	ideal	32	☐	faship	33	☐	accent
34	☐	negal	35	☐	modest	36	☐	oblight
37	☐	previous	38	☐	pation	39	☐	aparate
40	☐	barrel	41	☐	lodge	42	☐	editor
43	☐	dramatic	44	☐	flue	45	☐	reception
46	☐	laughouse	47	☐	soute	48	☐	witner
49	☐	bridegroom	50	☐	waitretch	51	☐	lean
52	☐	occasionally	53	☐	availy	54	☐	image
55	☐	honey	56	☐	impression	57	☐	superator
58	☐	flavour	59	☐	savinct	60	☐	whenery

H: FA: Dm:

Answer PG401: 3 6 8 9 12 15 28 32 34 36 38 39 46 47 48 50 53 57 59 60

AEFL Level 5:

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	❑	contandby	2	❑	administration	3	❑	cound
4	❑	embarrass	5	❑	evident	6	❑	slang
7	❑	simplicity	8	❑	sorrow	9	❑	goldsmiss
10	❑	bomber	11	❑	interfere	12	❑	dial
13	❑	downwards	14	❑	boldsmiss	15	❑	bang
16	❑	botten	17	❑	achievement	18	❑	jocken
19	❑	lessen	20	❑	harmful	21	❑	achiterpiece
22	❑	peck	23	❑	anxietness	24	❑	brand
25	❑	dismiss	26	❑	idle	27	❑	roast
28	❑	geomelet	29	❑	roller	30	❑	telesh
31	❑	welshment	32	❑	hastight	33	❑	breadth
34	❑	soften	35	❑	nerve	36	❑	award
37	❑	knickness	38	❑	senseless	39	❑	heel
40	❑	equartime	41	❑	smaste	42	❑	pea
43	❑	jockward	44	❑	rear	45	❑	separench
46	❑	pedest	47	❑	recreation	48	❑	firmness
49	❑	grease	50	❑	hook	51	❑	sneeze
52	❑	knickey	53	❑	rotten	54	❑	thermometer
55	❑	geometry	56	❑	lamb	57	❑	certicle
58	❑	invoice	59	❑	buttonhole	60	❑	lighten

H: FA: Dm:

Answer PG501: 1 3 9 14 16 18 21 23 28 30 31 32 37 40 41 43 45 46 52 57

C.3. The DMwW vocabulary test

Write your name here: _____

What you have to do:

Read through the list of words carefully. For each word:

if you know what it means, write Y (for YES) in the box.

if you don't know what it means, or if you aren't sure, write N (for NO).

1	☐	interface	2	☐	technical	3	☐	majorithm
4	☐	peptron	5	☐	optide	6	☐	identify
7	☐	cycle	8	☐	opervise	9	☐	procedure
10	☐	kernel	11	☐	reveal	12	☐	probably
13	☐	libline	14	☐	annear	15	☐	specialogy
16	☐	gradient	17	☐	infrared	18	☐	tedious
19	☐	preprocess	20	☐	circumstall	21	☐	mode
22	☐	correlate	23	☐	typical	24	☐	sensitive
25	☐	indicate	26	☐	apache	27	☐	predical
28	☐	deteriorate	29	☐	characteristic	30	☐	selete
31	☐	majority	32	☐	specific	33	☐	effective
34	☐	manual	35	☐	default	36	☐	outperform
37	☐	identiment	38	☐	algory	39	☐	stochanism
40	☐	structure	41	☐	visualizer	42	☐	multifilter
43	☐	explicit	44	☐	normat	45	☐	batch
46	☐	bioinform	47	☐	automatic	48	☐	passifier
49	☐	trivial	50	☐	sophisticated	51	☐	frustruct
52	☐	category	53	☐	finate	54	☐	assign
55	☐	distribute	56	☐	ripper	57	☐	accur
58	☐	bayer	59	☐	core	60	☐	template

H: FA: Dm:

Answer DMwW: 3 4 5 8 13 14 15 20 27 30 37 38 39 44 46 48 51 53 57 58

Appendix D. Meara's scoring matrix

This appendix contains Meara's scoring matrix for the EFL Vocabulary Test. Researchers and language teachers can use it to convert hit and false alarm counts to percentage vocabulary scores. Hits are listed in the first column, and false alarms in the first row. It is used to obtain percentage vocabulary scores in Section 5.8.1.

Meara's scoring matrix

where H refers to hit counts and FA refers to false alarm counts. Learners who score less than ten hits or more than ten false alarms should be handled separately as their results are likely to be unreliable (Meara, 2010, pp. 13-15).

FA / H	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	6	7	8	9	10
40	100	98	95	93	90	88	85	83	80	78	75	70	65	60	55	50
39	98	95	92	90	87	84	82	79	76	74	71	66	60	55	49	44
38	95	92	90	87	84	81	78	76	73	70	67	61	56	50	44	37
37	93	90	87	84	81	78	75	72	69	66	63	57	51	44	38	31
36	90	87	84	81	78	75	72	68	65	62	59	52	46	37	32	24
35	88	84	81	78	74	71	68	65	62	58	55	48	41	34	26	18
34	85	82	78	75	72	68	66	61	59	54	52	43	37	28	20	11
33	83	79	76	72	68	66	61	58	54	50	46	39	31	22	14	4
32	80	76	73	69	65	62	58	54	50	46	42	34	26	17	7	
31	78	74	70	66	62	58	54	50	46	42	38	29	20	11	1	
30	75	71	67	63	59	55	51	46	42	38	33	24	15	5		
29	73	68	64	60	56	51	47	43	38	34	29	19	9			
28	70	66	61	57	52	48	43	39	34	29	24	14	3			
27	68	63	59	54	49	44	40	35	30	25	20	9				
26	65	60	56	51	46	41	36	31	26	20	15	4				
25	63	56	53	48	42	37	32	27	21	16	10					
24	60	55	50	44	39	34	28	23	17	11	5					
23	58	52	47	41	35	30	24	18	12	6						
22	55	49	44	38	32	26	20	14	7	1						
21	53	47	41	34	28	22	16	9	3							
20	50	44	37	31	24	18	11	4								
19	48	41	34	28	21	14	6									
18	45	38	31	24	17	9	2									
17	43	35	28	20	13	5										
16	40	32	24	16	8	2										
15	38	29	21	12	4											
14	35	26	17	8												
13	33	23	13	4												
12	30	20	9													
11	28	17	6													
10	25	13	1													

Appendix E. Lexical bundles

This appendix contains a list of 160 academic sentence-initial lexical bundles, identified and recommending for academic writing by Li (2016). Their use in this thesis is discussed in Section 7.2.4.2.

As a matter of
As a result of
As a result, it
As a result, the
As can be seen
As discussed in Chapter
As far as the
As is shown in
As one of the
As shown in Table
As we all know,
At the beginning of
At the end of
At the same time
At the same time,
At the time of
Based on the above
By the end of
During the process of
First of all, the
For example, in the
For the purpose of
For the purposes of
For the sake of
From the above table,
From the perspective of
However, it is important
However, it is not
However, it should be
In a word, the
In addition to the
In addition to this,
In contrast to the
In order to find
In order to get
In order to make
In other words the
In other words, it
In other words, the
In other words, they
In spite of the
In terms of the
In the case of
In the context of
In the course of
In the current study
In the current study,
In the field of
In the light of
In the present study,
In the process of
In this case, the
In this chapter I
In this chapter, the
In this chapter, we
In this part, the
In this section I
In this section the
In this section, I
In this section, the
In this section, we
In this sense, the
In this study the
In this study, the
In this way, the
In view of the
It can be seen
It is also possible
It is argued that
It is believed that
It is clear that
It is difficult to
It is evident that
It is found that
It is hoped that
It is important that
It is important to
It is interesting that
It is interesting to
It is necessary to
It is not clear
It is obvious that
It is possible that
It is possible to
It is suggested that
It is true that
It may be that
It means that the
It must be noted
It seems that the

APPENDIX E LEXICAL BUNDLES

It should be noted
It should be pointed
It was important to
It would appear that
Last but not least,
Look at the following
On the basis of
On the one hand,
On the other hand,
One of the most
So it is necessary
That is to say
That is to say,
The aim of the
The aim of this
The analysis of the
The chapter concludes with
The fact that the
The findings of the
The findings of this
The first of these
The first one is
The following are some
The following is a
The following is an
The following is the
The following table shows
The limitations of the
The main purpose of
The majority of the
The next chapter will
The present study is
The purpose of the
The purpose of this
The result of the
The results from the
The results indicate that
The results of the
The results of this
The results show that
The results showed that
The thesis consists of
The use of the
There appears to be
There are a number
There is no doubt
There was a significant
There was no significant
There were no significant
Therefore, it is necessary
This chapter describes the
This chapter presents the
This is because the
This is followed by
This is not a
This is not to
This means that the
This suggests that the
This thesis consists of
To be more specific,
To put it another
To sum up, the
We can see from
We can see that
When it comes to
With regard to the
With respect to the
With the development of
With the help of

Appendix F. Ethics approval

This appendix contains the ethics approval letters for three studies: the EFL vocabulary test pilot study, described in Section 5.4, the EFL vocabulary test main study, described in Section 5.5, and the learner-data study, described in Section 8.2.

F.1. EFL pilot study

Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4021
www.fcms.waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

27 August 2018

Jemma Konig
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Jemma

Request for approval to conduct a user study with human participants

On the basis of the information you have provided on the FCMS Preliminary Ethics Application Form relating to your research "EFL Vocabulary Tests", the Committee has given you approval to proceed with your proposed study.

We wish you well with your research.

Mike Mayo
Human Research Ethics Committee
School of Computing and Mathematical Sciences

F.2. EFL main study

Faculty of Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4322
www.cms.waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

7 December 2018

Jemma Konig
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Jemma

Request for approval to conduct a user study with human participants

On the basis of the information you have provided on the FCMS Preliminary Ethics Application Form relating to your research "EFL Vocabulary Tests", the committee has given you approval to proceed with your proposed study.

We wish you well with your research.

Mike Mayo
Human Research Ethics Committee
Faculty of Computing and Mathematical Sciences

F.3. Learner-data study

Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4021
www.fcms.waikato.ac.nz

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

27 August 2018

Jemma Konig
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Jemma

Request for approval to conduct a user study with human participants

On the basis of the information you have provided on the FCMS Preliminary Ethics Application Form relating to your research "Online text enrichment study", the Committee has given you approval to proceed with your proposed study.

We wish you well with your research.

Mike Mayo
Human Research Ethics Committee
School of Computing and Mathematical Sciences