IMPLEMENTING GLOSSING IN MOBILE-ASSISTED LANGUAGE LEARNING ENVIRONMENTS: DIRECTIONS AND OUTLOOK

Hansol Lee, Korea Military Academy
Jang Ho Lee, Korea Military Academy

While mobile technology, such as the touch-based smart-phone, has become part of our daily lives, research into and classroom practices surrounding mobile-assisted language learning (MALL) have generally not kept up with the pace of technological development. This situation may be caused in part by the fact that a considerable proportion of researchers and teachers have not had personal experience with the newly developing technology. In order to improve this situation, the present paper provides comprehensible and useful directions for building web-based reading materials with integrated electronic glossing, with which readers can create their own web-based materials. The authors provide methodological descriptions and discussions of how to implement two useful formats of electronic glossing: (a) hyperlinks to target iframe and (b) tooltips by touching in web pages. The directions are carefully written so that these two formats of glossing work properly in both PC and mobile device environments. Lastly, the authors urge researchers and teachers to be more actively involved in the field of MALL and look forward to future work bridging the gap between computer-assisted language learning and mobile-assisted language learning.

Key Words: Computer-Assisted Language Learning, Glossing, Mobile-Assisted Language Learning, Vocabulary Acquisition

APA Citation: Lee, H., & Lee, J. H. (2013). Implementing glossing in mobile-assisted language learning environments: Directions and outlook. *Language Learning & Technology*, *17*(3), 6–22. Retrieved from http://llt.msu.edu/issues/october2013/emerging.pdf

Received: March 30, 2013; **Accepted:** May 17, 2013; **Published:** October 1, 2013

INTRODUCTION

Only seven years ago, in 2006, did Chinnery predict that an area of future language learning research would include mobile-assisted language learning (MALL). Initially developed from computer-assisted language learning (CALL), MALL immediately seized the attention of language education researchers and has been recognized as different from CALL in many respects, due to its “anytime, anywhere” principle (Kukulska-Hulme & Shield, 2008). Kukulska-Hulme (2009) notes that “mobile technology can assist learners at the point of need and in ways that fit in with their mobile lifestyles” (p. 162). Godwin-Jones (2011a) further suggests that “mobile devices [that deliver MALL] integrate real-life interests and academic roles” on the part of learners (p. 8). He additionally noted that this characteristic of MALL would eventually promote “learner autonomy” (Benson, 2007), meaning that learners can take more responsibility for their own learning. Stockwell (2007) holds that studying MALL is a natural path of computer-based learning research in light of the current situation in which “technologies themselves are becoming more pervasive, and the number of students who own mobile technologies … is increasing at an impressive rate” (p. 366). In addition, recent studies (e.g., Lu, 2008; Thornton & Houser, 2005) have shown that learners generally have positive attitudes towards using mobile devices.

Since 2006, a reasonable amount of research in a wide array of contexts has been conducted on the effects of MALL, in the areas of reading (e.g., Kondo et al., 2012; Lan, Sung, & Chang, 2007), speaking and listening (e.g., Demouy & Kukulska-Hulme, 2010), and vocabulary (e.g., Chen & Chung, 2008; Wong & Looi, 2010). Although more research with larger sample sizes and more sophisticated designs is necessary, the evidence accumulated so far suggests that mobile devices have much to offer to language learners.

While mobile technology is making inroads into classroom environments and our daily lives at a fast pace (Kukulska-Hulme, 2009), research into MALL has not kept up with the speed of technological development (Ballance, 2012). What is worse, in our view, is that most teachers and researchers interested in MALL themes are largely ignorant of how to build and tailor mobile-based learning materials for their students and participants. As a result, when adapting technology, teachers and researchers are highly reliant on computer experts, who may not prioritize concerns regarding language teaching and learning. Although some may argue that equipping oneself with knowledge of computer languages and programs might be burdensome, we suggest otherwise. It is only after those directly concerned with language teaching and learning begin to grasp how mobile technology operates that MALL will become fully integrated into the process of language teaching and learning. This will require that language teachers micro-tailor their materials to the requirements of the curriculum and needs of learners, and that researchers take a much greater role in adjusting materials to their research focus. For example, in the case of glossing, teachers may be able to put the definition of a target word in learners' mother tongue (see our examples below) or any other information they find useful for their learners (e.g., using concordance data for target words: http://hansol6461.dothome.co.kr/call/test(corpus).htm). Thus, we believe that it would be a useful undertaking for these two groups to start getting their feet on the expanding ground of MALL.

To this end, we attempt to present here succinct and useful directions for building web-based reading materials integrated with electronic glossing for target language teachers and researchers without much knowledge of computer language. The present paper provides directions for creating materials that can interface with a wide range of smart-phones, especially as glossing has been shown to be effective for language learning (Abraham, 2008).

The present paper begins by providing a brief review of the effects of glossing on language learning documented in previous research, with reference to paper-, computer-, and mobile-based studies. It goes on to deal with theoretical and methodological descriptions, and discussions of how to integrate electronic glossing into web-based reading materials. Lastly, we suggest that researchers and teachers be more involved in the field of MALL, and indicate the ways in which the gaps between CALL and MALL may be bridged.

BACKGROUND FOR GLOSSING IN TARGET LANGUAGE READING AND MALL

Given the value of reading in a target language (Coady, 1997; Krashen, 1989), researchers have explored several pedagogical options that enable language learners to benefit from reading linguistically demanding texts, for example texts with low-frequency vocabulary. One line of research has focused on how reading can become more manageable, comprehensible, and convenient for language learners. To this question, glossing—written information relating to unfamiliar vocabulary contained in texts, usually being a definition—has been one of the most suggested pedagogical options. Nation (2009) summarizes the benefits of using glossing as follows:

> Under appropriate conditions glosses can help learners read a text more fluently, can raise comprehension of the text, and can increase vocabulary learning. Glossing can be used with both hardcopy and computerized texts. Moreover, readers tend to have positive attitudes towards glosses. (p. 102)

The positive effect of glossing on vocabulary acquisition has been found in several reading studies (e.g., Hulstijn, Hollander, & Greidanus, 1996; Jacobs, Dufon, & Fong, 1994; Watanabe, 1997), in which increases in target vocabulary were attributed to glossing. A general claim resulting from these studies is that glossing makes target vocabulary more salient, consequently increasing the likelihood of these items being acquired by the language learners. It is thought that ambitious learners may revisit glossed words as well as their definitions several times to record them in their memory, and this may lead to long-term vocabulary acquisition.

In the context of language learning through technology, Chapelle (2003) reiterated the value of glossing as input enhancement, which makes "the input understandable to the learner through any means that gets at the meaning" (p. 40). In fact, as part of the growing interest in CALL materials including the use of mobile devices, glossing has quickly become part of computerized support options, especially to enhance reading comprehension and vocabulary acquisition. Research has already documented the pedagogical effects of glossing with these devices: Abraham (2008) carried out a meta-analysis of 11 studies that investigated computer-mediated glossing and found that glossing has a medium effect on language learners' reading comprehension and a large effect on language learners' incidental vocabulary acquisition (though the effect may be mediated to some extent by some factors such as the learners' proficiency levels, text types, and assessment measures). With the limited amount of evidence accumulated thus far regarding the effects of computer-mediated glossing, it is expected that the next phase of computer-mediated glossing research will begin to examine the effects of glossing in mobile environments, presumably with a focus on its relevance to the "anytime, anywhere" principle.

However, technical difficulties concerned with creating and using computer and mobile learning materials have placed MALL research somehow out of reach for a considerable number of researchers and teachers in the field of language teaching and learning. In addition, a significant portion of research seems to have taken place in a rather restricted number of regions to date. For example, the few studies that have examined the effects of mobile devices on vocabulary acquisition (e.g., Lu, 2008; Song & Fox, 2008; Stockwell, 2010) were carried out in Asian countries like Hong Kong, Japan, and Taiwan. It is of utmost importance that, for the effects of MALL to be validated, research should be conducted within a wide range of educational contexts and with learners of various backgrounds. To meet this end, we believe that researchers and teachers in regions where MALL research remains not as widely developed should take a sizable step in the direction of examining the effects of MALL with their learners. In what follows, we provide comprehensive directions so that these populations can try building web-based reading materials that are enhanced by glossing. We have chosen this topic as it only requires CSS (Cascading Style Sheets) and JavaScript in Hypertext Mark-up Language (HTML) which, according to Godwin-Jones (2011b), can be built by just "scripting [in human languages] rather than [coding] programming languages, hence do not involve compilation into byte-code" (pp. 5–6).

DIRECTIONS ON HOW TO IMPLEMENT GLOSSING

In this section, we would like to introduce two distinct formats of electronic glossing that can be built into web pages. As will be made clear below, the suggested methods of glossing are meant to operate properly in mobile device environments (not only in PC settings). It is noteworthy that we are dealing here with web-based materials—rather than software applications specifically designed to run on mobile devices—for the following reasons: since there are numerous types of devices with different programming environments (e.g., Objective-C and Apple's Xcode for iPhone, and Java for Android based smart-phones), producing a type of application that suits one device would not be interoperable with others. Furthermore, web-based materials are more conducive to resolving any technical difficulties that can be caused by diverse types, versions and configurations of mobile web browsers (e.g., Chrome, FireFox, Internet Explorer, Opera, and Safari) and by various screen sizes (e.g., Galaxy, iPhone, and Nexus).

To understand these mobile environments, we begin by describing their interfaces, and provide preset HTML source code for one to be able to integrate the suggested glossing into one's own web pages. This is further followed by our own example—one that we built for our university-level students in Korean

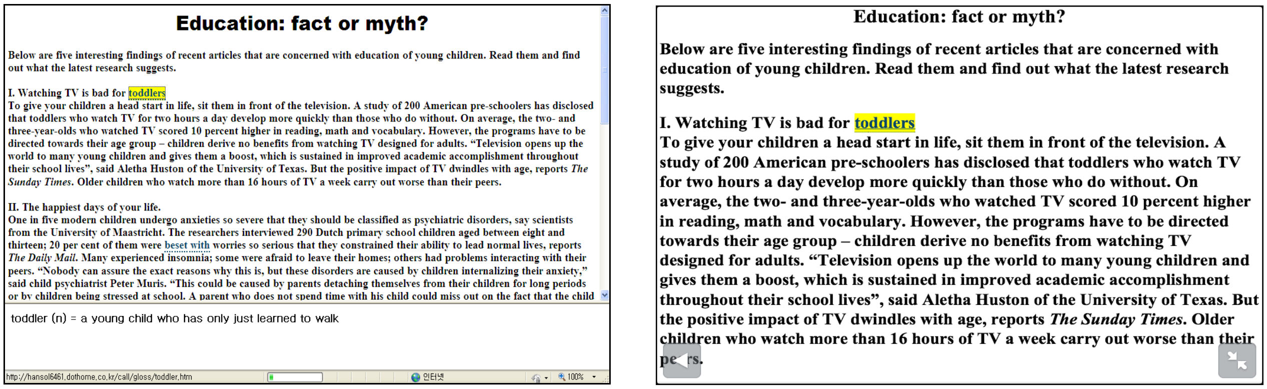
EFL reading classes in 2012 (for more information on our examples, visit the following web site: http://hansol6461.dothome.co.kr/call(2012).htm). In order for readers to build their own version using these methods, they may simply place our HTML source code (see Appendices below) into their HTML file, and doing so will help readers implement the suggested glossing formats.[1]

Before delving into the directions on how to implement electronic glossing in HTML, it would be useful for readers to familiarize themselves with the concept and develop some knowledge of HTML. First, to put it simply, we would say that an HTML file consists of two elements: (a) HTML tags enclosed in angle brackets (e.g., <head>) that define the structures of the document; and (b) the HTML file's contents (e.g., a reading passage). Second, these HTML tags form a mark-up language, which is mostly composed of pairs of start (e.g., <head>) and end (e.g., </head>) tags. These start and end tags take a major role in delivering information when constructing pages for web browsers. These tags translate information into the graphical user interface (GUI), which is what the users see and interact with on their screens. It should be noted in passing that the same HTML web page could be loaded differently, albeit to a small degree, according to different types of web browsers.

Hyperlinks to Target Iframe

One of the first and most important steps when developing a coding framework is to consider how the screen the user sees will be divided. To divide a web page into multiple sections so that several items can be loaded to different parts of the screen on a single window simultaneously, a normal HTML frame can be used. For example, one ubiquitous HTML frame is the vertically divided web page with menus located on the left hand side and main contents on the right (e.g., http://hansol6461.dothome.co.kr/cbt(2011)/index.htm). However, in the case of designing a computer-based reading text, you may want to split the screen horizontally, using the upper part of the window to load the reading text and the lower part to provide extra information about target words (e.g., http://hansol6461.dothome.co.kr/call/test(frame).htm).

In terms of framing, unfortunately, the aforementioned properties of HTML as a mark-up language prevent us from using the same frameset when using touch-based devices. As shown in Figure 1 where the reading passage "Education: fact or myth?" from the textbook *Cutting Edge Advanced* (Cunningham, Moor, & Carr, 2003) is displayed, web pages with framed divisions on the PC will not be displayed in the same way on most mobile web browsers (e.g., the mobile version of Safari for Apple's iPhone and ICS Stock Android Browser for Samsung's Galaxy).



Education: fact or myth?

Below are five interesting findings of recent articles that are concerned with education of young children. Read them and find out what the latest research suggests.

I. Watching TV is bad for toddlers
To give your children a head start in life, sit them in front of the television. A study of 200 American pre-schoolers has disclosed that toddlers who watch TV for two hours a day develop more quickly than those who do without. On average, the two- and three-year-olds who watched TV scored 10 percent higher in reading, math and vocabulary. However, the programs have to be directed towards their age group – children derive no benefits from watching TV designed for adults. "Television opens up the world to many young children and gives them a boost, which is sustained in improved academic accomplishment throughout their school lives", said Aletha Huston of the University of Texas. But the positive impact of TV dwindles with age, reports *The Sunday Times*. Older children who watch more than 16 hours of TV a week carry out worse than their peers.

II. The happiest days of your life.
One in five modern children undergo anxieties so severe that they should be classified as psychiatric disorders, say scientists from the University of Maastricht. The researchers interviewed 290 Dutch primary school children aged between eight and thirteen; 20 per cent of them were beset with worries so serious that they constrained their ability to lead normal lives, reports *The Daily Mail*. Many experienced insomnia; some were afraid to leave their homes; others had problems interacting with their peers. "Nobody can assure the exact reasons why this is, but these disorders are caused by children internalizing their anxiety," said child psychiatrist Peter Muris. "This could be caused by parents detaching themselves from their children for long periods or by children being stressed at school. A parent who does not spend time with his child could miss out on the fact that the child

toddler (n) = a young child who has only just learned to walk

Computer

Education: fact or myth?

Below are five interesting findings of recent articles that are concerned with education of young children. Read them and find out what the latest research suggests.

I. Watching TV is bad for toddlers
To give your children a head start in life, sit them in front of the television. A study of 200 American pre-schoolers has disclosed that toddlers who watch TV for two hours a day develop more quickly than those who do without. On average, the two- and three-year-olds who watched TV scored 10 percent higher in reading, math and vocabulary. However, the programs have to be directed towards their age group – children derive no benefits from watching TV designed for adults. "Television opens up the world to many young children and gives them a boost, which is sustained in improved academic accomplishment throughout their school lives", said Aletha Huston of the University of Texas. But the positive impact of TV dwindles with age, reports *The Sunday Times*. Older children who watch more than 16 hours of TV a week carry out worse than their pe rs.

iPhone

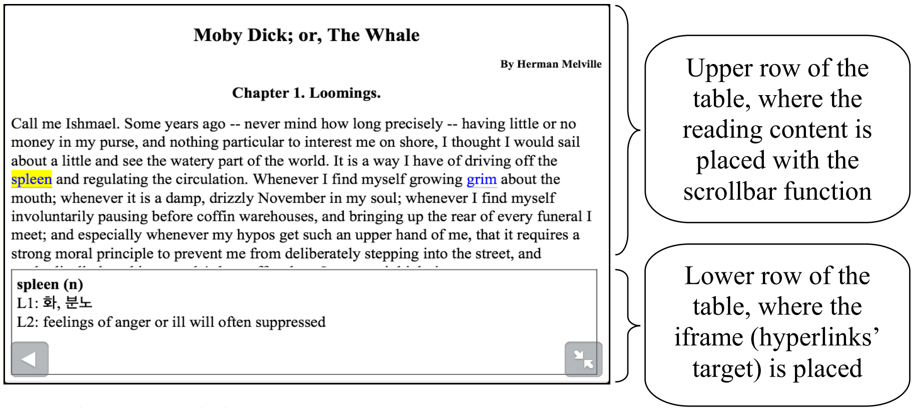
Figure 1. A framed HTML page on Computer and iPhone screens. (http://hansol6461.dothome.co.kr/call/test(frame).htm)

In particular, the differences may occur when the contents overflow the frame, (i.e., when the user needs to scroll up or down because the text is larger than the screen size). In web browsers for PC settings, one can designate the attribute of automatic scrolling when setting the frames, such that each frame carries a scroll function. The frame size is then set to the screen, and the scrollbar appears on the right side, so the user can scroll up and down to see the whole text within the frame. However, in the mobile web browsers of smart phones, the framed window does not have this scrollbar function. Thus the framed part is expanded to the size of its content, extending beyond the visible screen. This means that the bottom frame containing glossary information of the target words will literally appear at the very bottom of the reading text, and thus readers need to scroll laboriously to the end of the web page using their fingers to see the glossary information they are looking for. This is to say, on mobile devices, one cannot simultaneously take in the text, a target word, and its glossary information in a single glance. In other words, using the same HTML file divided by frames for PC settings will not work for touch-based devices. This makes it difficult—when using touch-based devices—to follow Al-Seghayer's (2003) suggestion on glossing:

> An ideal place for the annotated words display would be in an area separate from the text. When readers click on a word, the definition window immediately appears in a separate glossary window that is adjacent to or alongside the text being read. This location allows learners to see the word in its context while they are reading the definition. (p. 3)

To create a conceptualization of your web page in a way that allows you to overcome these drawbacks, we suggest using a table and HTML inline frame tags called *iframe*, which "is displayed as a separate box or window within a Web page" (Carey & Canovatchel, 2006, p. REV 45). The direction we suggest is as follows: First, using the *table* tag enables us to divide any type of mobile screen as if we were using the typical HTML frame for PC settings. Then, by inserting the *division* tags inside this table, you can provide the scrollbar function indirectly in the window. In doing so, a table with two rows (a top and a bottom row) can effectively function as a HTML frameset that divides the screen into upper and lower frames to be used for the placement of upper and lower content.

Second, placing iframe tags in the lower row of the table enables this lower row to be used as the target frame for hyperlinks. Iframe tags ensure that target words in the text function as hyperlinks to glossary information in the lower row. This lower row will then project the glossary information of target words, making this table act just like the normal HTML frame in PC settings. In Figure 2, an excerpt from Melville's *Moby Dick* (1851) is displayed on the iPhone screen, using the hyperlinks that link the target word *spleen* to its glossary information in the iframe.[2]



Moby Dick; or, The Whale

By Herman Melville

Chapter 1. Loomings.

Call me Ishmael. Some years ago -- never mind how long precisely -- having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and

spleen (n)
L1: 화, 분노
L2: feelings of anger or ill will often suppressed

Upper row of the table, where the reading content is placed with the scrollbar function
Lower row of the table, where the iframe (hyperlinks' target) is placed

Figure 2. "Hyperlinks to Target Iframe" on iPhone screen.
(http://hansol6461.dothome.co.kr/llt/iframe.htm)

For the technical construction of this type of web site using the hyperlinks to target iframe structure, you will need at least two HTML pages: the main page with iframe tags (which concerns the text) and the source page that presents glossary information (see Appendix A and B for more detailed information).

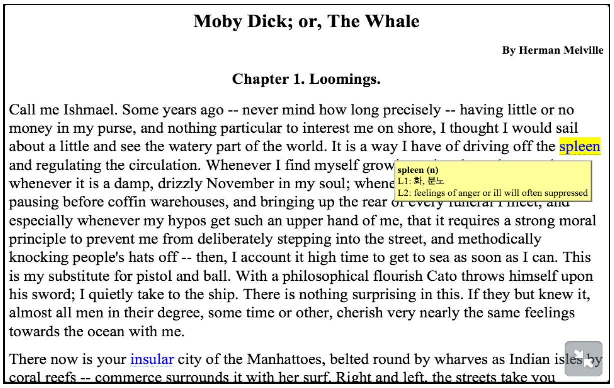
As a first step, you need to build a table that spans two rows in the main page, which defines the layout of a set of the text and glossing rows in a single window view (see Step 3, 4 in Appendix A). In this example, the table tag (<table> and </table>), whose height and width attributes are 100%, indicates that you are going to draw a full sized table on the screen compatible with any type of display. For the upper row of the table, division tags (<div> and </div>) with the scrolling function (in this example, *overflow: scroll* is inserted as the style of the division) have to be designated. In this way, this division will provide users of mobile devices with an invisible scroll bar that allows them to scroll up and down to see the whole texts within the fixed size of the upper row. In the lower row, another set of division tags (<div> and </div>) should be included to set the inline frame function or iframe tags (<iframe> and </iframe>) inside this lower row itself. Since the iframe is to be a blank page used to display the text from the hyperlinks, there is no need to type in any kind of contents. However, we need to designate attributes such as width, height, and name (in this example, the name for this iframe is *iframe*).

As a second step, to create a hyperlink for a specific target word, specific HTML markup must be added around the target word (see Step 6, 7 in Appendix A). For the word *spleen*, the head tag <a href="spleen.htm" target="iframe"> and end tag </a> must be placed to the left and right of the target word. In addition, to make the clicked words visually distinguishable, the CSS Class function is included in the hyperlinks' attributes. In this example, we named the CSS Class ".emphasis" in the style section of the web page. As one of the principles of HTML tags, the period (.) before the randomly selected name ("emphasis" in this case) serves as attributing a certain CSS Class function (as defined in Step 2 in Appendix A) to this name. This period is no longer required when this randomly selected name is used in hyperlinks. The attributes of the CSS Class in our example make the target words visually salient and distinguishable when one clicks or touches the vocabulary, loading the function of CSS Class added in the head section of the main HTML page.

As a third or final step, one needs to build additional HTML pages for each word's glossary information (see the steps in Appendix B). In particular, for example, the word *spleen* in the sentence "It is a way I have of driving off *spleen* and regulating the circulation" and the word *grim* in the sentence "Whenever I find myself growing *grim* about the mouth" (our emphasis) were set to be hyperlinks in this example. If you click these words, the web page "spleen.htm" or "grim.htm" will be loaded on the iframe in the lower table. Note that we provide the meaning of this word in both the learners' first language (Korean) and second language (English).

Tooltips by Touching

It is thought that glossing should provide relevant information in a useful location close to the target words without interrupting the flow of one's reading (e.g., by blocking the portion of the text one is reading or by searching for glosses in a different location) (Al-Seghayer, 2003). The *tooltip* function, which you can experience in many computer programs, could be a suitable option in this case. In some commercial computer programs (e.g., Microsoft Word and Internet Explorer), you can activate this tooltip function when you move your mouse cursor over a range of menus or buttons in the program interface. This function provides descriptive information about the menus or buttons that your mouse cursor is pointing at before you actually click and initiate certain commands; literally, the tooltip function provides readers tips for understanding the tool. The next HTML format of electronic glossing we introduce is the one that creates the tooltip. This format will provide glossary information for target vocabulary as a form of tooltip when the mouse cursor is placed on top of a word (see Figure 3).



Moby Dick; or, The Whale

By Herman Melville

Chapter 1. Loomings.

Call me Ishmael. Some years ago -- never mind how long precisely -- having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself grow whenever it is a damp, drizzly November in my soul; whene pausing before coffin warehouses, and bringing up the rear of every funeral I meet, and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and methodically knocking people's hats off -- then, I account it high time to get to sea as soon as I can. This is my substitute for pistol and ball. With a philosophical flourish Cato throws himself upon his sword; I quietly take to the ship. There is nothing surprising in this. If they but knew it, almost all men in their degree, some time or other, cherish very nearly the same feelings towards the ocean with me.

spleen (n)
L1: 화, 분노
L2: feelings of anger or ill will often suppressed

There now is your insular city of the Manhattoes, belted round by wharves as Indian isles by coral reefs -- commerce surrounds it with her surf. Right and left, the streets take you

Figure 3. “Tooltips by touching” on iPhone screen. (http://hansol6461.dothome.co.kr/llt/tooltip.htm)

In PC environments, there are two simple ways to implement the tooltip function using HTML. The first way is using *abbreviation* tags, which are originally used to represent the complete form of the abbreviation or acronym when you move the mouse cursor over the abbreviation element (e.g., http://hansol6461.dothome.co.kr/call/test(abbr).htm). While this may be the simplest way to implement tooltips, this tag enables the glossary information to appear only briefly once activated, and thus does not satisfy those learners who may want the glossary information to be present much longer. The second way is adding a few lines of preset CSS tags (which is rather similar to what we introduce in the following paragraphs) to the simple HTML code, which can supply customized tooltips to users for several actions of the mouse cursor (e.g., http://hansol6461.dothome.co.kr/call/test(css).htm).

However, in touch-based mobile devices, these two methods will not work properly unless thoroughly designed considering the unique properties of these devices. The reason for this is that the finger-touch input system of these devices does not have a mouse cursor. Consequently, it cannot recognize the mouse cursor’s movements on the screens, except for clicks on clickable objects or hovers on the objects that are customized with certain codes. Especially on iPhones, the action of touching the screen is almost always regarded as the intention of scrolling the page up and down or left and right, not as the action of clicking. As a result, abbreviation tags cannot be used in mobile settings. Furthermore, mobile devices do not recognize the action of the cursor moving off a target word, meaning that if you tried to use the simple CSS tags as mentioned above, the glossary information would remain visible even if you did not want it to be. In addition, this customized method cannot prevent tooltips from being cut off by the border lines of the screen, especially when it comes to the small and fixed screen size of smart-phones.

To solve this conceptual problem, abbreviation tags with CSS tags and JavaScript code including library plug-ins, widely used tags to add abundant styling effects to simple HTML pages, can be added to reproduce the PC setting tooltip function with complete fidelity in the touch-based mobile devices environments. As mentioned above, when designing this “tooltips by touching” method, there are several caveats we need to keep in mind: this method has (a) to enable touch devices to recognize the mouse cursor’s hovering action on the target word; (b) to make the tooltips disappear when the reader wants to; (c) and to provide alternative ways to prevent tooltips from being cut off by screen edges. To this end, we need to employ CSS tags along with JavaScript code with a jQuery library.[3] Thanks to this JavaScript library, one page with several CSS and JavaScript tags is all we need to build the “tooltips by touching,” as shown in Appendix C.

As a first technical step, we designate CSS ID selector (e.g., #tooltip) as a part of the CSS code along with CSS Class (e.g., .emphasis) in the style section (see Step 4, 5 in Appendix C). While we used a period (.) as a mark for the CSS Class, the hash sign (#) is a required sign for the CSS ID. So using this hash sign along with the customized name, in this example, we designate CSS ID selector as *#tooltip*. The reason for adopting this selector is to set the customized formative features of tooltips so that we could link this selector to the target words' abbreviation tags by declaring the relationship (e.g., <abbr rel="tooltip">) to customize the style of abbreviation functions. In adding its properties, you can modify the color or style of the tooltip boxes' inner space and outer lines. The position, however, has to be *absolute*, meaning that you place the selector exactly where you want it to be without being affected by other elements. In this way you can manipulate the specific location of each tooltip to be near the target words in the next area of the direction of reading. As we did in the previous section "Hyperlinks to Target Iframe," the CSS Class, with the name "emphasis," is also required to be written in this section in order to make the target words visually distinguishable when clicked or touched. To summarize, in this example, CSS ID selector carries attributes of the formative features of the tooltips, while CSS Class designates styling attributes of the target words.

As a second step, the JavaScript code located in the script section (<script> and </script>) follows the style section (see Step 6 through 14 in Appendix C). The script section is to add further specifications to CSS ID selector, or in this example, the tooltips that will contain and present glossary information for the target words. This section consists mainly of five parts: (a) initiating the tooltip function (Step 7), (b) designating a function to start the tooltip event (Step 8), (c) setting the tooltips' initial positions (Step 9), (d) suggesting an alternative when tooltips are cut off by the right or bottom borderlines (Step 10, 11), and (e) removing tooltips when the mouse leaves in the PC setting or when a finger is removed from a word on mobile devices (Step 13).

To follow this complex-looking step, there are several points you need to understand. First, a *mouseover* code in the second part enables tooltips to appear when a mouse cursor hovers over the target word in PC settings, or your finger touches it in mobile devices. Second, several mathematical functions in the second, third and fourth parts are added as well to designate the size and position of tooltips. According to our example, these functions determine the positions for the initial state and the contingencies of being cut off by right and bottom borderlines. Third, the last part ensures that the tooltip will disappear after consulting the target word, making this "tooltips by touching" method work properly in both PC and mobile device environments. As part of this effort, the *mouseleave* code enables tooltips on the screen to disappear when the mouse cursor leaves the target words in PC settings, while a *click* code would make the tooltip vanish when you click or touch it in mobile devices.

As a last step, you need to provide several attributes to the target words to make them perform the above-mentioned functions as a tooltip (see Step 16, 17 in Appendix C). First, the target word should be designated with abbreviation tags (<abbr> and </abbr>). Second, one should apply attributes to each abbreviation tag that are necessary for it to function as a customized tooltip, involving relation (e.g., rel="tooltip") and title (e.g., title="*glossary information*"). Third, to make target words visually distinguishable when clicked or touched, CSS Class attributes tags should be attached before the abbreviation tags (e.g., <a class=emphasis> and </a>).

CONCLUSION AND OUTLOOK

In this paper, we have provided directions for implementing two useful formats of electronic glossing on the mobile devices: (a) hyperlinks to target iframe and (b) tooltips by touching. We hope that our illustration here has not scared researchers and teachers away from using technology-based materials, but that we have shown that they are manageable and even enjoyable. We also hope the present work will entice readers who suffer from technophobia to undertake their own projects. We would like to reiterate that researchers and teachers who are about to introduce and share technology with their learners should not remain in ignorance of how it works. If they have some basic knowledge of building web sites and writing HTML code, they may be able to micro-tailor mobile-based learning materials so that they work most effectively for their own learners.

In light of the rise of smart-phones and tablet PCs in the market which entails the need to develop learning materials that can be applied to more than one particular device, we hope our efforts here will be seen as an attempt to bridge the gap between CALL and MALL. That is, as Godwin-Jones (2011b) notes, developing applications for mobile devices has somewhat limited effects because each OS system has its own platform, and thus the way of developing and utilizing each application should be subject to that particular programming environment. On the other hand, his alternative suggestion of building compatible web applications, while useful and cost-effective for program developers, would be rather too ideal for most language teachers and researchers at the starting line of developing MALL. Nonetheless, this suggestion has made us explore a third path, building web pages that are compatible with both PC and mobile device environments. The directions given above are entirely built in this effort, and we expect that more diverse types of web learning materials will be developed following this path. In this way, researchers and language teachers will be able to keep up with growing technology with no insurmountable obstacles, and infuse more pedagogically-oriented ideas based on education theory, practical teaching experience, and insiders' knowledge of learners and contexts into the development of MALL.

APPENDIX A. Sample HTML Code for the Main Page of the "Hyperlinks to Target Iframe"

Step 1. Starting HTML codes

```
<html>
<head>
<title>Hyperlinks to iframe</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Step 2. Defining CSS Class

```
<style type="text/css">
.emphasis{position:absoulte;z-index:24;color:blue;border-bottom:thin dotted #046;text-decoration:none}
.emphasis:hover{z-index:25; background-color:#FF0}
</style>
```

```
</head>
```

Step 3. Defining table tag with a full screen size

```
<table width="100%" height="100%">
```

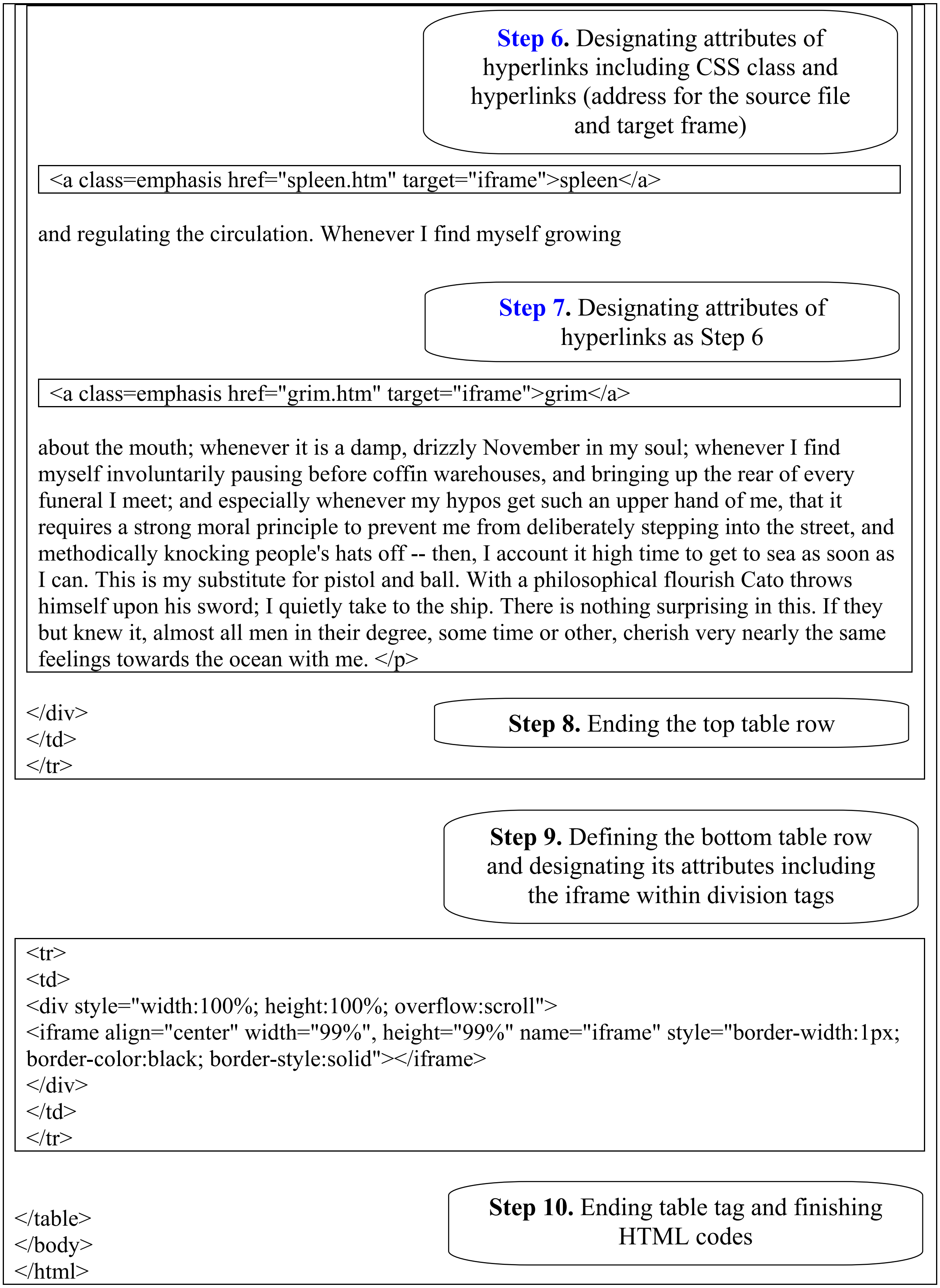
Step 4. Defining the top table row and designating its attributes

```
<tr>
<td height=70%>
<div style="width:100%; height:100%; overflow:scroll">
```

Step 5. Typing the main text

```
<p><h1 align=center><b>Moby Dick; or, The Whale</h1></b></p>
<p><h3 align=right>By Herman Melville</h3></p>
<p><b><h4 align=center>Chapter 1. Loomings.</h4></b></p>

<p>Call me Ishmael. Some years ago -- never mind how long precisely – having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the
```



Step 6. Designating attributes of hyperlinks including CSS class and hyperlinks (address for the source file and target frame)

```
<a class=emphasis href="spleen.htm" target="iframe">spleen</a>
```

and regulating the circulation. Whenever I find myself growing

Step 7. Designating attributes of hyperlinks as Step 6

```
<a class=emphasis href="grim.htm" target="iframe">grim</a>
```

about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and methodically knocking people's hats off -- then, I account it high time to get to sea as soon as I can. This is my substitute for pistol and ball. With a philosophical flourish Cato throws himself upon his sword; I quietly take to the ship. There is nothing surprising in this. If they but knew it, almost all men in their degree, some time or other, cherish very nearly the same feelings towards the ocean with me. </p>

```
</div>
</td>
</tr>
```

Step 8. Ending the top table row

Step 9. Defining the bottom table row and designating its attributes including the iframe within division tags

```
<tr>
<td>
<div style="width:100%; height:100%; overflow:scroll">
<iframe align="center" width="99%", height="99%" name="iframe" style="border-width:1px;
border-color:black; border-style:solid"></iframe>
</div>
</td>
</tr>
```

```
</table>
</body>
</html>
```

Step 10. Ending table tag and finishing HTML codes

APPENDIX B. Sample HTML Code for the Source Page (spleen.htm) of the "Hyperlinks to Target Iframe"

Step 1. Starting HTML codes

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
</head>
```

Step 2. Adding the lexical information for glossing

```
<b>spleen (n)</b><br>
L1: 화, 분노<br>
L2: feelings of anger or ill will often suppressed
```

Step 3. Finishing HTML codes

```
</body>
</html>
```

APPENDIX C. Sample HTML Code for the Main Page of the "Tooltips by Touching"

Step 1. Starting HTML codes

```
<html>
<head>
<title>Tooltips by touching</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<meta content=IE=edge, chrome=1 http-equiv=X-UA-Compatible>
<meta name=robots content=all>
<meta name=viewport content=width=device-width, initial-scale=1>
```

Step 2. Loading jQuery library from Google CDN in script section

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
```

Step 3. Adding style section

```
<style>
```

Step 4. Designating attributes for CSS ID including its selector's position value (absolute)

```
#tooltip
{position: absolute;
border-style: solid; border-color: black; padding:.3em; border:2px outset #BBB; color:#000;
background:#FF9;
text-align: center;}
```

Step 5. Defining CSS Class

```
.emphasis{position:absoulte;z-index:24;color:blue;border-bottom:thin dotted #046;text-
decoration:none}
.emphasis:hover{z-index:25; background-color:#FF0}

</style>
```

Step 6. Adding script section

Step 7. Initiating the tooltip function: designating a function to execute the tooltip event when the DOM (jQuery) is loaded and ready

```
<script>
$( document ).ready( function()
{
        var targets = $( '[rel~=tooltip]' ),
                target      = false,
                tooltip     = false,
                title  = false;
```

Step 8. Designating a function to start the tooltip event when a mouse cursor locates on or a finger touches the target words

```
targets.bind( 'mouseover', function()
        {
                target      = $( this );
                tip    = target.attr( 'title' );
                tooltip     = $( '<div id="tooltip"></div>' );

                if( !tip || tip == '' )
                        return false;

                target.removeAttr( 'title' );
                tooltip.css( 'opacity', 0 )
                           .html( tip )
                           .appendTo( 'body' );
```

Step 9. Setting tooltip’s initial positions

```
var init_tooltip = function()
                {
                        if( $( window ).width() < tooltip.outerWidth() * 1.5 )
                                tooltip.css( 'max-width', $( window ).width() / 2 );
                        else
                                tooltip.css( 'max-width', 340 );

                        var pos_left = target.offset().left + 20,
                                pos_top     = target.offset().top + 20;
```

Step 10. Suggesting an alternative when tooltips are cut off by the right borderline

```
if( pos_left + tooltip.outerWidth() > $( window ).width() )
                        {
                                pos_left = target.offset().left - tooltip.outerWidth() + target.outerWidth() / 2
+ 20;
                                tooltip.addClass( 'right' );
                        }
                        else
                                tooltip.removeClass( 'right' );
```

Step 11. Suggesting an alternative when tooltips are cut off by the bottom borderline

```
if( pos_top + tooltip.outerHeight() > $( window ).height() )
                        {
                                pos_top     = target.offset().top - target.outerHeight() + target.outerHeight() /
2 + 20;
                                tooltip.addClass( 'bottom' );
                        }
                        else
```

```
                    tooltip.removeClass( 'bottom' );
```

Step 12. Setting tooltip’s revised positions and attributes (e.g., opacity)

```
      tooltip.css( { left: pos_left, top: pos_top } )
                    .animate( { top: '+=10', opacity: 1 });
          };

          init_tooltip();
          $( window ).resize( init_tooltip );
```

Step 13. Setting a function to remove tooltips when mouse leaves in PC or when finger touches it on mobile devices

```
var remove_tooltip = function()
          {
               tooltip.animate( { top: '-=10', opacity: 0 }, 50, function()
               {
                    $( this ).remove();
               });

               target.attr( 'title', tip );
          };

target.bind( 'mouseleave', remove_tooltip );

tooltip.bind( 'click', remove_tooltip );

     });
});</script>
```

Step 14. Finishing the script section

```
</head>
<body>
```

Step 15. Typing the text

```
<p><h1 align=center><b>Moby Dick; or, The Whale</h1></b></p>
<p><h3 align=right>By Herman Melville</h3></p>
<p><b><h4 align=center>Chapter 1. Loomings.</h4></b></p>

<p>Call me Ishmael. Some years ago -- never mind how long precisely – having little or no
money in my purse, and nothing particular to interest me on shore, I thought I would sail about
a little and see the watery part of the world. It is a way I have of driving off the
```

Step 16-1. Designating attributes of CSS class (e.g., emphasis)

Step 16-2. Designating attributes of ABBR tags including CSS ID selector (e.g., tooltip) and the information for glossing

```
<a class=emphasis><abbr rel="tooltip" title="<div align=left><b>spleen (n)</b> <br> L1:
화, 분노<br> L2: feelings of anger or ill will often suppressed</div>">spleen</abbr></a>
```

```
and regulating the circulation. Whenever I find myself growing
```

Step 17. Designating attributes of CSS class and ABBR tags as Step 16

```
<a class=emphasis><abbr rel="tooltip" title="<div align=left><b>grim (adj)</b> <br> L1:
(보기나 듣기에 아주) 엄숙한[단호한]<br>L2: fierce in disposition or
action</div>">grim</abbr></a>
```

```
about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself
involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I
meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong
moral principle to prevent me from deliberately stepping into the street, and methodically
knocking people's hats off -- then, I account it high time to get to sea as soon as I can. This is
my substitute for pistol and ball. With a philosophical flourish Cato throws himself upon his
sword; I quietly take to the ship. There is nothing surprising in this. If they but knew it, almost
all men in their degree, some time or other, cherish very nearly the same feelings towards the
ocean with me. </p>
```

```
</body>
</html>
```

Step 18. Finishing HTML codes

NOTES

1. For those who lack experiences with HTML, we recommend reading *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics* (Robbins, 2012), or visiting reliable websites such as *Wikipedia* or *HTML tutorial* by W3School for detailed information on HTML.

2. If you right-click with your mouse on the example page in PC browsers, you will see "viewsource" on the menu. Then you will be able to see the HTML source code of the content.

3. This library is a single JavaScript file that contains DOM (Document Object Model) manipulations that allow us to give additional effects on CSS ID selectors, among numerous other useful functions. See Godwin-Jones, 2004, 2011b for more discussion on this matter. To load the jQuery library, which is free and open source, our example recommends that you use Google's CDN (Content Delivery Network) page or Microsoft's CDN page unless you are going to use this teaching material in off-line environments (in this case you have to download this library file from the Internet). As one of the most popular JavaScript libraries in use, the huge number of connections in this file in the jQuery server (http://jquery.com) would slow the loading time of your web page; therefore, Google and Microsoft CDN page provide online code archives along with jQuery's website.

ABOUT THE AUTHORS

Hansol Lee is an assistant professor in the Department of English at Korea Military Academy. His research interests include computer- and mobile-assisted language learning, corpus linguistics, and language assessment.

Jang Ho Lee received his DPhil in education from the University of Oxford. He is presently an assistant professor in the Department of English at Korea Military Academy. His areas of interest are teachers' code-switching in English classrooms, learners' attitudes towards teachers' language uses, and using mobile technology for language learners.

All correspondence should be addressed to Jang Ho Lee at jangho330@gmail.com.

REFERENCES

Abraham, L. B. (2008). Computer-mediated glosses in second language reading comprehension and vocabulary learning: A meta-analysis. *Computer Assisted Language Learning*, *21*(3), 199–226.

Al-Seghayer, K. (2003). Technological and pedagogical considerations for a more effective electronic glossary. *The Reading Matrix*, *3*(1). Retrieved from http://www.readingmatrix.com/articles/al-seghayer/article.pdf

Ballance, O. J. (2012). Mobile language learning: More than just "the platform." *Language Learning & Technology*, *16*(3), 21–23. Retrieved from http://llt.msu.edu/issues/october2012/ballance.pdf

Benson, P. (2007). State-of-the-art article: Autonomy in language teaching and learning. *Language Teaching*, *40*(1), 21–40.

Carey, P., & Canovatchel, F. (2006). *New perspectives on JavaScript—comprehensive*. Boston, MA: Thomson/Course Technology.

Chapelle, C. A. (2003). *English language learning and technology*. Amsterdam, Netherlands: John Benjamins.

Chen, C. M., & Chung, C. J. (2008). Personalized mobile English vocabulary learning system based on item response theory and learning memory cycle. *Computers & Education*, *51*(2), 624–645.

Chinnery, G. M. (2006). Going to the MALL: Mobile assisted language learning. *Language Learning & Technology*, *10*(1), 9–16. Retrieved from http://llt.msu.edu/vol10num1/pdf/emerging.pdf

Coady, J. (1997). L2 vocabulary acquisition through extensive reading. In J. Coady & T. Huckin (Eds.), *Second language vocabulary acquisition: A rationale for pedagogy* (pp. 225–237). Cambridge: Cambridge University Press.

Cunningham, S., Moor, P., & Carr, J. C. (2003). *Cutting edge advanced with phrase builder: Students' book*. Harlow, UK: Pearson Education.

Demouy, V., & Kukulska-Hulme, A. (2010). On the spot: Using mobile devices for listening and speaking practice on a French language programme. *Open Learning: The Journal of Open, Distance and e-Learning*, *25*(3), 217–232.

Godwin-Jones, R. (2004). Making the web dynamic: DOM and DAV. *Language Learning & Technology*, *8*(1), 8–12. Retrieved from http://llt.msu.edu/vol8num1/pdf/emerging.pdf

Godwin-Jones, R. (2011a). Autonomous language learning. *Language Learning & Technology*, *15*(3), 4–11. Retrieved from http://llt.msu.edu/issues/october2011/emerging.pdf

Godwin-Jones, R. (2011b). Mobile apps for language learning. *Language Learning & Technology*, *15*(2), 2–11. Retrieved from http://llt.msu.edu/issues/june2011/emerging.pdf

Hulstijn, J. H., Hollander, M., & Greidanus, T. (1996). Incidental vocabulary learning by advanced foreign language students: The influence of marginal glosses, dictionary use, and reoccurrence of unknown words. *The Modern Language Journal, 80*(3), 327–339.

Jacobs, G. M., Dufon, P., & Fong, C. H. (1994). L1 and L2 vocabulary glosses in L2 reading passages:

Their effectiveness for increasing comprehension and vocabulary knowledge. *Journal of Research in Reading*, *17*(1), 19–28.

Kondo, M., Ishikawa, Y., Smith, C., Sakamoto, K., Shimomura, H., & Wada, N. (2012). Mobile assisted language learning in university EFL courses in Japan: Developing attitudes and skills for self-regulated learning. *ReCALL*, *24*(2), 169–187.

Krashen, S. (1989). We acquire vocabulary and spelling by reading: Additional evidence for the input hypothesis. *The Modern Language Journal*, *73*(4), 440–464.

Kukulska-Hulme, A. (2009). Will mobile learning change language learning? *ReCALL*, *21*(2), 157–165.

Kukulska-Hulme, A., & Shield, L. (2008). An overview of mobile assisted language learning: From content delivery to supported collaboration and interaction. *ReCALL*, *20*(3), 271–289.

Lan, Y.-J., Sung, Y.-T., & Chang, K.-E. (2007). A mobile-device-supported peer-assisted learning system for collaborative early EFL reading. *Language Learning & Technology*, *11*(3), 130–151. Retrieved from http://llt.msu.edu/vol11num3/pdf/lansungchang.pdf

Lu, M. (2008). Effectiveness of vocabulary learning via mobile phone. *Journal of Computer Assisted Learning*, *24*(6), 515–525.

Melville, H. (1851). *Moby dick; or the whale*. Retrieved from http://www.gutenberg.org/files/2701/2701-h/2701-h.htm

Nation, P. (2009). New roles for L2 vocabulary? In L. Wei & V. Cook (Eds.), *Contemporary applied linguistics volume 1: Language teaching and learning* (pp. 99–116). London, UK: Continuum.

Robbins, J. N. (2012). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics*. Cambridge, UK: O'Reilly Media.

Song, Y., & Fox, R. (2008). Using PDA for undergraduate student incidental vocabulary testing. *ReCALL*, *20*(3), 290–314.

Stockwell, G. (2007). Vocabulary on the move: Investigating an intelligent mobile phone-based vocabulary tutor. *Computer Assisted Language Learning*, *20*(4), 365–383.

Stockwell, G. (2010). Using mobile phones for vocabulary activities: Examining the effect of the platform. *Language Learning & Technology*, *14*(2), 95–110. Retrieved from http://llt.msu.edu/vol14num2/stockwell.pdf

Thornton, P., & Houser, C. (2005). Using mobile phones in English education in Japan. *Journal of Computer Assisted Learning*, *21*(3), 217–228.

Watanabe, Y. (1997). Input, intake, and retention: Effects of increased processing on incidental learning of foreign language vocabulary. *Studies in Second Language Acquisition*, *19*(3), 287–307.

Wong, L.-H., & Looi, C.-K. (2010). Vocabulary learning by mobile-assisted authentic content creation and social meaning-making: Two case studies. *Journal of Computer Assisted Learning*, *26*(5), 421–433.