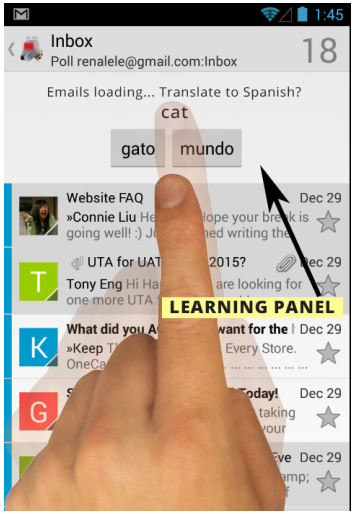

Figure 1: Pull-To-Learn prototype presenting a Multiple-Choice format exercise. Exercise appears in the dead-space revealed when a user pulls to refresh email.

Pull-To-Refresh and Learn: Leveraging Mobile Email Load Time for Education

Anji Ren
MIT CSAIL
362 Memorial Dr.
Cambridge, MA 02139 USA
aren@mit.edu

CHI'15, April 18 - 23, 2015, Seoul, Republic of Korea
ACM 978-1-4503-3146-3/15/04.
http://dx.doi.org/10.1145/2702613.2726968

Abstract
Users of connected devices regularly encounter technical inefficiencies that result in waiting for small durations of time. In this paper, we examine how we can leverage these brief moments of waiting when users pull-to-refresh email on their mobile phones to engage people in learning. We extended an existing mobile email client by presenting a micro-quiz when the user pulls to refresh email. The exercise is displayed within the dead-space resulting from the pull to minimize intrusiveness, and allows the user to complete the exercise without leaving the email context. In multiple iterations of preliminary user testing with 10 users and a pilot deployment with 3 users, we found that participants were able to non-intrusively integrate learning into their existing mobile activities.

Author Keywords
Design; Education

ACM Classification Keywords
K.3.1 [Computer Uses in Education]: .

Introduction
In an increasingly connected world, activities on our mobile devices occupy a large share of our time. In these activities, users often encounter *technical inefficiencies*, performance issues in technology that are out of the user's

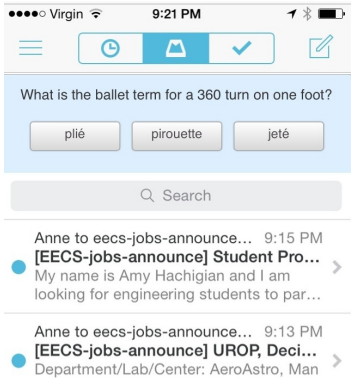

Figure 2: Learning panel sized to regular pull-to-refresh loading panel size.

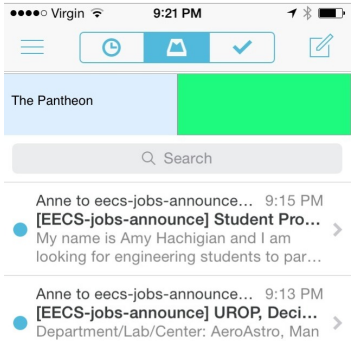

Figure 3: After user swipes to answer, a color affordance displays whether the answer was correct.

control. For example, users can frequently encounter downtime due to an unsteady Internet connection problem, resulting in a slow Web page load. The amount of time a user is willing to wait is defined by expectation: a faster Web connection means a user is willing to wait less, and modern advancements in technology are leading to decreasing tolerance from users [6]. Those who desire to learn informally often find it difficult to find time to learn. Downtime resulting from technical inefficiencies can be used as opportunities for learning. Previous work in *micro-learning* has shown various ways to distribute learning into small moments throughout a person's day-to-day life [5]. The ALOE system teaches second language by augmenting Web pages with translations of the primary language into a secondary language for learning [7]. MicroMandarin teaches Mandarin Chinese in small doses by leveraging physical context provided by Foursquare [4]. It has been shown that spaced exposure [3] and repetition [8] aid in vocabulary retention.

More recent research in *wait-learning* tackles usage of time people ordinarily spend waiting for distributed micro-learning. For example, WaitChatter helps instant messenging (IM) users in learning a second language through embedding language exercises into a chat client during moments when the user is awaiting an IM response [2]. Notably, it was found through WaitChatter studies that an opportune time to present an exercise is right after the user sends a message, as it catches him or her during a moment of conversational dead-time (while the user waits for a response) and gives the user an opportunity to spend the dead-time doing something of interest.

In this paper, we extend wait-learning by embedding learning during waiting moments induced by technical inefficiency. This category of waiting results from issues in technology that delay users from activities on their devices. We present the design and early evaluation of Pull-To-Learn (PTL) for second language acquisition, a mobile email client prototype that enables users to learn while they wait for remote fetching of email (Figure 1). The moment post-pull is an opportune time for injection because it occurs when the user is waiting in anticipation of new content loading, making him or her potentially receptive to doing something else in the meantime. The proposed benefit is that learning occurs during dead-time, which is otherwise lost to a technical process, and within context of the email client, allowing users to make use of this time without having to intentionally switch to another application.

Design Space

In designing the PTL prototype, we faced a number of human-centric design decisions. These decisions directly address the core tensions in introducing an embedded secondary activity of learning into the main activity of managing email. This section discusses these alternatives and the decisions we made based on feedback from 10 users on early design iterations. Designs were tested as animated HTML on-rails prototypes built in Tumult Hype, a rapid-prototyping software.

Where and how should the exercise appear?

User studies of WaitChatter suggest that the timing and presentation of the exercise relative to the main IM activity are critical to users perceiving learning as intrusive or time-consuming [2]. To minimize intrusiveness, we design the exercise to appear within the dead-space that is normally uncovered when the user pulls for new email. However, since users tend to navigate mobile devices with one hand using their thumbs, it is unclear whether an affordance at the top of the screen would be sufficiently

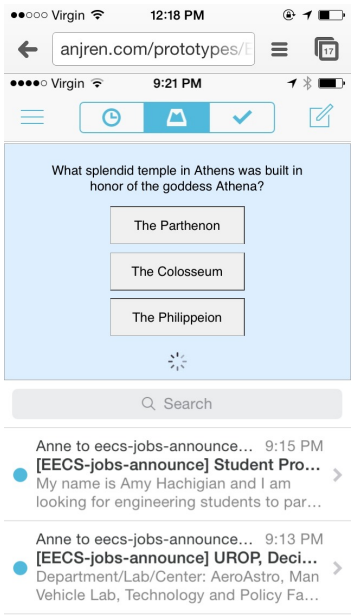

Figure 4: Larger than usual size for learning panel.

easy to reach. Thus, in addition to a prototype which fills the standard amount of space typically revealed during pull-to-refresh (Figure 2), we also tested an alternative in which users pull-to-refresh from the side, and another in which the user answers the exercise by swiping horizontally rather than pushing a button that could potentially be positioned too far away (Figure 3). During initial user testing, users found it less intuitive to pull from the side because the pull-to-refresh mechanism typically requires the user pull from the top. Users also found swiping horizontally dangerous because it resembled swipe gestures they would use within their email applications. Accidentally swiping on top of an email can result in an unwanted deletion.

Based on this feedback, the exercise appears at the top where normally a loading icon would provide affordance that data is currently being downloaded. Users pull from the top of the screen to the bottom to trigger the pull-to-refresh mechanism and reveal the loading affordance, which in our PTL prototype is replaced by an exercise. Users interact with the exercise by button-pressing, as opposed to typing or writing text, to minimize motor effort and to accommodate one-handed usage. Another consideration was the size of the exercise and its components. Mobile devices have limited screen space for content. We prioritize ease of access as users have a limited amount of time, restricted by email loading time, to interact with the exercise. At the same time, the exercise should be minimally sized in order to give the user sufficient view of his or her current emails. In one design alternative, we sized the learning panel much larger than usual (Figure 4). Many users expressed concern that the panel blocked view of their emails. One user commented, "I can only see two of my emails, whereas I usually can see five."

When and how should the exercise be dismissed?
It is important that the learning activity provided to the user does not demand excessive attention from the user if he or she is not interested in interacting with the exercise. We created several different designs by varying the number of seconds before the exercise automatically dismisses, both in response to user interaction and the absence of user interaction. We also varied mechanisms for users to actively dismiss the exercise.

For timing, if users did not wish to interact with the exercise at all, they found a wait of ten seconds from first seeing the exercise too long and too disruptive. Three seconds was not enough time for users to decide to interact with the exercise, but users found around five to six seconds a good amount of time for the exercise to wait. We also considered the length of delay after a user interacts with an exercise before hiding the learning panel. Users found five seconds, however, too long of a wait after interacting with the exercise, but two to three seconds was optimal for absorbing whether their provided answer was correct or incorrect without annoying the user.

Through testing, users also found swiping left, right, or down as unintuitive ways to manually dismiss the exercise. They expected swiping up to dismiss the exercise because it is opposite to the motion that triggered it in the first place and it also indicates that they are more interested in their emails.

Therefore, in our prototype, the user can dismiss the exercise at any time by swiping up, an opposite motion to the one that initially triggered the exercise. This will collapse the exercise but not stop the email load and keep the loading icon affordance visible. The system also automatically retracts the exercise five seconds after email loading is complete so that the user can more quickly

resume browsing his or her email.

How should users fetch follow-up exercises?
Prior research on wait-learning suggests that, when users do engage with the exercise, these moments should be capitalized by optionally allowing the user to continue learning [2]. We can encourage users to learn more by providing them with additional opportunities to continue fetching multiple exercises in sequence. We user tested two alternatives: one implementation keeps the learning panel open after the user completes one exercise, and allows the user to fetch additional exercises by pressing a button. The other prototype requires the user to actively pull again to fetch a second exercise.

Half of our test users preferred being given the option to fetch another exercise after completing the initial exercise, and the other half preferred for the panel to disappear automatically after the initial exercise. Because feedback was mixed, we decided to build the prototype where users are required to pull again to fetch another exercise, for our deployment study.

User Interface

We built the PTL prototype (Figure 1) as an extension of K-9, an open-source email client for Android. The default behavior of K-9 when pull-to-refresh is triggered by a user is to display a progress icon, indicating that emails are being currently loaded into the local device from remote servers. The PTL prototype augments the progress window that appears on a user pull to enable presentation of interactive educational content.

Interactive Exercise Panel
Incorporating the exercise into the normal user flow of a person using their email client, we have the panel appear at the top of the client window after the user has pulled and released, starting a manual refresh. The panel is positioned at the top where normally only loading text and a progress icon appears. This panel displays the exercise in addition to the visual indicators which let the user see if email downloading has completed.

Although the learning panel could display any micro-learning content such as flashcards or facts, for the current prototype, we designed learning around second language acquisition. The content and formatting was constructed based on existing work that has demonstrated success in teaching vocabulary through micro-learning.

Exercise Formats
The Flashcard format is an interactive flashcard with self-grading, similar to the flashcard format used in MicroMandarin [4]. The prompt is a word either in the primary language of the user or the second language for learning. The translation of the prompt to the other language is hidden by a button that the user can click to reveal. After revealing the translation, the user clicks 'yes' or 'no' to indicate whether they had correctly translated the prompt as a means of self-grading.

In the Multiple-Choice (MC) format (Figure 1), the prompt is again a word in either of the two languages and the two answer choices are buttons comprising of a *target* (correct answer) and *distractor* (wrong answer) response.

In both formats, the entire learning panel is dismissed automatically five seconds after email load completion if the user has not interacted with the panel, or two seconds it the user answered the exercise. The five second delay before automatic dismissal allows decision and interaction. The two second delay post-interaction allows users to review feedback.

Preliminary Deployment Study

We deployed two versions of a working prototype to participants in a larger active study to measure the following:

1) Frequency of learning opportunities: how often people pull-to-refresh determines how many learning opportunities a user of PTL has.

2) Degree of engagement: what percentage of exercises users interact with and how quickly. In addition, how often users pull multiple times in succession.

Participants

Three users between the ages of 17 and 22 (1M, 2F) took part in our deployment study. The participants were undergraduate students in a single university. Users were recruited based on the requirement that they regularly check their email on their mobile devices, use pull-to-refresh as a mechanism for refreshing their emails, and use Android as their primary mobile platform. The participants also all had some previous knowledge of Spanish but were not native speakers or advanced learners.

Method

For this deployment, we loaded PTL with Spanish vocabulary content, targeted for teaching Spanish to native English speakers.

Procedure

Each participant was asked to use the prototype as their normal email client for two weeks and were told what we were logging. At the end of two weeks, the users were interviewed remotely post-study for qualitative feedback. The data collected over two weeks was quantified to learn about the frequency of use.

Results

Our results suggest that pull-to-refresh enhanced with learning presents perceivable benefits to users who are interested in learning. From logged data, the three users were exposed to 3.2 exercises per day on average (individually: 2.1, 3.5, 4). Some exercises were completed in clusters during a short time period where users were interested in doing follow-up exercises after the first pull. Discounting follow-up pulls, users pulled on average 2.2 times per day (1.8, 2.4, 2.4) with a 26% rate of engagement (25%, 18%, 35%), with successful engagement defined as a user answering an exercise. In processing the data, we counted only the first pull in clusters of multiple successive pulls to order to accurately the rate of engagement. The three users also averaged a response time of 4.1 seconds (4.0, 4.5, 3.8) in interacting with the exercise. Response time was measured from pulling to clicking on reveal, in Flashcard format exercises, and to clicking on an answer choice, in MC format exercises.

The data suggests that while our users did not pull-to-refresh their email very often per day, there are opportunities created for them to learn. The low rate of engagement combined with an average response time relatively close to the 5 second timeout points us towards further investigation on whether 5 seconds is really enough for users to interact with the exercises. Qualitative feedback from interviews with users indicated that they found PTL interesting and useful but also raises areas for improvement for future deployment testing. One user said, "The [Flashcard format] exercises require a lot of thought, so I usually ran out of time before I could click the 'reveal' button."

On the whole, our participants found PTL enjoyable. One

participant gave as feedback: "I would use it if it was installed on other applications that I use." A different participant remarked that they were "pleasantly surprised with something to do" and "had fun because [they] had always wanted to review Spanish", after having studied it before, but "took initiative to get around to it."

Conclusion & Future Work

An important contribution of this research is demonstrating how applications for wait-learning can be expanded to moments of technical inefficiency, and the design decisions central to these interactions. Results from our preliminary user testing and pilot deployment suggest that the current design enables users to access learning while pulling to refresh their email. The decisions made leading to the prototype allows users to make use of email loading time educationally through micro-quizzes and bite-sized acquisition of vocabulary. We also learned that the PTL feature is also enjoyable for users.

However, the size of our current study and deployment restricts how much we know about the overall impact of our concept. Future work includes examining how effective learning is through this mechanism, which can be extended to a number of other data-dependent mobile applications since pull-to-refresh is found in other domains, such as social media activity feeds. We have observed that users can maintain a daily frequency of exposure to micro-exercises through our prototype but need to continue deployment testing and also measure the extent to which learning is retained.

Acknowledgements

This research was conducted with the User Interface Design group in MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). This work would not have been possible without the help of my direct advisors Carrie Cai and Rob Miller and the entire UID group.

References

[1] Berleant, D., Lovelady, S., Viswanathan, K. (2004). A foreign vocabulary learning aid for the networked world of tomorrow. ACM SIGICE Bulleton, 19(3), 22-29.

[2] Cai, C. J., et al. (2015). Wait-Learning: Leveraging wait time for second language education. CHI 2015.

[3] Dempster, F. N. (1987). Effects of variable encoding and spaced presentations on vocabulary learning. Journal of Educational Psychology, Vol. 79(2), June 1987, 162-170.

[4] Edge, D., et al. (2011). MicroMandarin: mobile language learning in context. CHI 2011.

[5] Gassler, G., Hug, T. & Glahn, C. (2004). Integrated micro learning - an outline of the basic method and first results. Interactive Computer Aided Learning 4.

[6] Nah, F. (2004). A Study on Tolerable Waiting Time: How Long Are Web Users Willing to Wait?, Behaviour and Information Technology, Vol. 23, No. 3.

[7] Trusty, A. & Truong, K. (2011). Augmenting the web for second language vocabulary learning. CHI 2011.

[8] Webb, S. (2007). The effects of repetition on vocabulary knowledge. Applied Linguistics 28(1), 46-65.