A Study of the Use of Current Speech Recognition in an Information-Intensive Task

Shiry Ginosar
University of California, Berkeley
Berkeley, CA 94704 USA
shiry@cs.berkeley.edu

Marti Hearst
University of California, Berkeley
Berkeley, CA 94704 USA
hearst@berkeley.edu

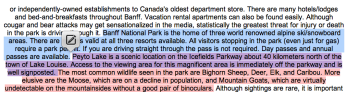
CHI'14, April 26–May 1, 2014, Toronto, Canada.

DOI string from ACM form confirmation

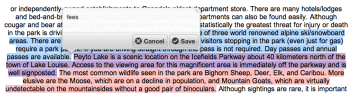
Abstract
Speech input is growing in importance, especially in mobile applications, but less research has been done on speech input for information intensive tasks like document editing and coding. This paper presents results of a study on the use of a modern publicly available speech recognition system on document coding. We record the performance and preferences of 7 expert coders on two types of documents. Participants voiced concern about the well-known drawbacks of speech recognition: the response time was felt to be slower than desired and the recognition was thought to be not accurate enough. Other concerns include the need to work with others in quiet or loud spaces. However, some of the experienced coders preferred the speech interface because they saw the advantages of a multimodal design for this task, commenting on the reduced manual manipulation needed for typing, and a less repetitive feeling.

Author Keywords
Speech input; multimodal; document coding

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.



(a) Edit button



(b) Input window

Figure 1: A coder selects the desired text (in blue) and clicks an edit button (a) in order to pop up the input window (b). Here she can assign the appropriate code using typing in one interface and speech in the other. In this view two codes were already applied, one color-coded in purple and one in pink.

Introduction

Speech recognition interfaces have seen tremendous advances in recent years primarily in mobile technologies to which they are well suited [6]. However, a few previous studies examined the use of speech interfaces in the context of information-intensive tasks such as annotating collaboratively written papers and commenting computer code [4, 5, 7]. Unfortunately, at the time these studies were run, speech recognition technology was not yet advanced enough to enable the researchers to use a working system. Instead, most studies relied on simulations. We set out to re-examine user preference and performance when interacting multi-modally in information intensive tasks. In contrast to previous studies, we take advantage of a current state-of-the-art publicly-available speech recognition system [8].

The task we focus on is *document coding*, a practice used in various disciplines in order to prepare textual data for in-depth analysis [3]. This process involves reading text documents and marking each word, sentence or paragraph as belonging to a descriptive category, or code, according to a given schema. The process of coding a document requires that the annotator perform two simultaneous (time-sharing) tasks: (1) reading the document and (2) annotating it according to the given schema, which requires first selecting text and then annotating it with a code. We chose this task because we hypothesize that an interface that allows the coder to use the complementary modalities of reading (visual) and speaking (audio) would lead to better performance than an interface that allows the coder to use only visual modalities like reading (visual) and typing (visual). Furthermore, because the text selection component requires manual manipulation, the audio input should provide manual relief over the need to type to enter the code.

Interface Designs

We designed and implemented two web-based document coding interfaces that vary only in the modality used for assigning a code to a chunk of selected text. In one interface, the code is input via keyboard entry (**typing**) and the other uses audio (**speech**) input.

Both interfaces are extensions of the Annotator tool, an open source document annotator [2]. In both interfaces the coder selects the desired text using a mouse and clicks an edit button (Figure 1a) to pop up an input window (Figure 1b). Once a code is assigned, the relevant text is highlighted with the color associated with the code. Multiply-coded text is highlighted with a mixture of all relevant code colors. Both interfaces display all codes, uniquely color coded, in a column at the right of the screen so that the coder need not memorize the schema.

In the typing interface the coder types the appropriate code into the input window and may use an auto-complete functionality by hitting either the Tab or the Enter key. A second press on Enter or a click on the Save button records the code and closes the entry box.

In the speech interface the coder speaks the appropriate code and its transcription is automatically entered into the input window. Speech recognition is performed in real-time using Google's implementation of the HTML5 Web Speech API [8]. However, a keyboard is not necessary for using this interface as the text can be selected and buttons clicked using only the mouse.

Experiment

We designed an experiment to test two hypotheses for the document coding task: that participants would code more efficiently using the speech interface and that they would prefer it over typing. These hypotheses were only partially

supported by the results of the study.

Participants
Participants in the study were 7 researchers (2 male), who have been coding documents about the Occupy Movement for between 3 and 8 months as part of the DOSSIER project [1]. Each participant coded 851 documents on average before the start of this study (min 10, max 1626). All were students in the sociology department, 6 undergraduate and one graduate (ave age 22.1) and all were native English speakers.

Tasks
Two document sets were used: excerpts of news articles about the Occupy Movement (avg 14.5 sentences per document, 10 codes in the schema), and excerpts of pages from Wikivoyage, a free collaborative travel guide [9] (avg 24.25 sentences per document, 15 codes in the schema). Occupy documents were taken from the DOSSIER project articles. Although these were shorter in length, they often required elaborate double and triple coding of sentences according to the DOSSIER schema. Travel guide documents were chosen as their structure and content is readily familiar to most educated people. The schema we used for this task consisted of 15 top-level section headers from the Wikivoyage documents such as "see", "eat" and "get around".

Procedure
Each session lasted 1 hour and was conducted in a lab setting. Coding was done on a 15" MacBook pro. Participants' speech was captured using a Logitech headset with an attached microphone.

The order of display of documents was fixed, but order of presentation of interface modality was chosen at random. Participants did all coding in one modality first, then filled out an interface-specific questionnaire, then coded the remaining documents using the other modality, filled out a second questionnaire, and then completed a final post-study questionnaire that compared the two interfaces. (Participants coded 2 Occupy, 2 Travel in one modality and then, 2 Occupy, 2 Travel in the other). We captured **coding action time:** from when the user clicked on the edit button to when the assigned code was saved.

Results

Qualitative
While the Likert scale post-interface questions produced inconclusive results, a post-study direct comparison questionnaire revealed that participants prefer the speech to the typing interface (4 participants out of 7) and view it as more efficient (4 out of 7) and easier to use (5 out of 7). Still, they find the typing interface more suitable for coding documents due to the slow response times and low accuracy of using current speech recognition technology.

Participants listed as disadvantages of the speech interface the variable accuracy of speech recognition, the difficulty of working in a shared environment and the difficulty of correcting recognition mistakes. Interestingly, participants recognized that speech input has the potential to be more physically efficient than switching between a pointing device and a keyboard and allows for better time-sharing with a visual task. In essence, they brought up all the major theoretical points that support multimodal speech interfaces in the literature [6]. The willingness of expert users to pay the price of reduced accuracy may be because of their need to alleviate some of the repetitiveness of long-term manual annotation work.

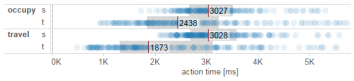


Figure 2: Participants took less time to type a code than to speak it. Graph shows action times, measured from edit button click to recognition, by task (occupy vs. travel) and condition (speech vs. typing). Outliers longer than 6s were removed from graph. Median, 2nd and 3rd quartiles highlighted.

Quantitative

Applying a code via speech was slower than typing for both types of documents (Figure 2). For occupy documents, participants took 3.15 seconds on average to apply a code using speech (standard deviation 0.71 seconds) and 2.65 seconds by typing (standard deviation 1.15 seconds). The difference is statistically significant with $p < 0.05$ (Student T-Test, $t = 5.123541$, $p < 0.00001$). For travel documents, participants took 3.14 seconds on average to apply a code using speech (standard deviation 0.66 seconds) and 2.16 seconds by typing (standard deviation 1.17 seconds). The difference is again statistically significant ($t = 11.049841$, $p < 0.00001$). The larger variance in the typing condition may be due to differences in participants' typing speeds.

The difference in the time it takes to apply a single code is partly due to performing speech recognition in real-time, especially using the Web Speech API as voice input needs to travel to a remote server and back in addition to the time spent in recognition itself [8]. However, when applying multiple codes in sequence the speech interface can allow for some parallelism as the next code can be applied while the last one is being recognized.

Conclusion

In this paper we revisited the inclusion of speech input in interfaces for information-intensive tasks. Specifically, we focused on document coding, a common data preparation task in the humanities and social sciences. In a small-scale study, we asked experienced coders to annotate two types of documents. We measured their performance and collected their preferences using two variations of input to a coding interface: speech and typing. Using speech was slower across the board due to time spent on recognition and a round trip to the speech recognition application's servers. However, we learned from the qualitative data that while the participants are aware of the limitations of currently available speech recognition systems, most of them see the benefits of using speech input for coding documents. Participants preferred the speech interface and welcomed an alternative input method.

Acknowledgements

This work is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1106400 and by grant HK-50011 from the National Endowment for the Humanities.

References

[1] Adams, N., et al. Dossier: Detailed observations of sequential strategies, interactions, events, and repertoires. Tech. Rep. NSF Grant SES-1303662, UC Berkeley, 2013.

[2] Annotator. http://okfnlabs.org/annotator/.

[3] Auerbach, C. F., and Silverstein, L. B. *Qualitative data: An introduction to coding and analysis*. NYU press, 2003.

[4] Kraut, R., Galegher, J., Fish, R., and Chalfonte, B. Task requirements and media choice in collaborative writing. *Hum.-Comput. Interact. 7*, 4 (Dec. 1992), 375–407.

[5] Neuwirth, C. M., Chandhok, R., Charney, D., Wojahn, P., and Kim, L. Distributed collaborative writing: a comparison of spoken and written modalities for reviewing and revising documents. In *Proc. CHI*, ACM (1994), 51–57.

[6] Oviatt, S. Multimodal interfaces. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, Third Edition*, A. Sears and J. Jacko, Eds. Taylor & Francis, 2002.

[7] Soudian, S., and Fels, D. I. Verbal source code descriptor. In *Proc. IEEE Workshop on Empirical Studies of Software Maintenance* (2002).

[8] Web Speech API. https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html.

[9] Wikivoyage. http://en.wikivoyage.org/.