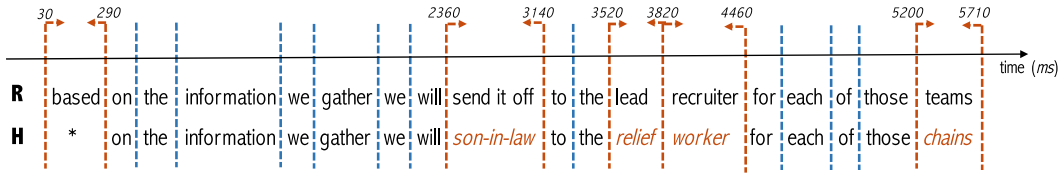

Figure 4: Time based alignment of reference (R) and hypothesized (H) text. The grouping with red dotted arrowhead lines indicates individualized errors aligned with corresponding reference text based on word level timestamps.

We extracted the verbatim script of what the human actor said during the videos, and we used the entire text of this video as a potential source of stimuli sentences for inclusion in this study (details in 5.1.1 and 5.1.2 below). Next, we processed the original audio recording from these videos using an ASR system that we expected to make a large number of errors (it is important for our stimuli selection process described in 5.1.2 for us to have many possible errors to choose from). For this processing, we used the CMU Sphinx 4 system with its off-the-shelf US English acoustic and language models which have been previously disseminated to the research community[3].

While a simplistic approach for creating stimuli for the study would have been to simply display the raw output of the ASR system to users, we were interested in obtaining judgments from participants on texts that had a variety of ACE metric scores. Furthermore, to investigate hypothesis H1, we were interested in presenting users with some pairs of ASR text output that displayed multiple hypotheses (i.e. two different guesses from the ASR system about what it heard), with one of the texts having a low WER-to-ACE score ratio (indicating that WER believed the text to be good, but ACE did not) and the other with a high WER-to-ACE ratio. Since ASR systems actually consider a wide variety of hypotheses when they analyze a speech audio file (with one hypothesis correct, and the remainder containing some variety of errors), we wanted to search the space of ASR output candidate hypotheses to select texts to display in our study with various WER-to-ACE ratios. Sections 5.1.1 and 5.1.2 describe our procedure for identifying ASR output hypotheses to display in our study with diverse WER-to-ACE ratios. Rather than inventing artificial errors to insert into the texts, our procedure obtains a large number of real ASR errors on a text and selects a subset of these errors to include in the texts displayed.

5.1.1 Time-based Alignment

After we prepared the meeting script and ran it against our low-accuracy ASR system, the next step was to align the reference text (the verbatim script of what the human actually said) and the hypothesis text (the output of the ASR system) to obtain a list of all the errors in the ASR output. While the ASR output hypothesis text already included timestamps of when the ASR believed each word had been spoken, we needed to identify timestamps for each word in the reference text. We manually compared the reference text to the original audio to obtain timestamp values for each word.

Next, we needed to time-align the hypothesis text to the reference text, to correctly identify all the errors in the hypothesis text. Standard alignment tools like [12] were ill-suited to this task because they are designed to compute the edit distance of the reference text from the hypothesis text. Our task required alignment of the text to capture the exact regions of errors – the goal of which is different slightly from the edit distance computation. For the purpose we wrote code to identify different error regions in the ASR output. Often, there is no one-to-one correspondence between an error word and a reference word. Multiple reference words can be misrecognized as a single word (substitution followed by deletions) and a single reference word can be misrecognized as multiple words (substitution followed by insertions) [22][32]. Our time-based error alignment software uses word-level timestamps to group the errors appropriately, as shown in Figure 4. The output of our processing is a list of confusion pairs for each sentence. For the example in Figure 4, the confusion pairs would be: (based, *); (send it off, son-in-law); (lead, relief); (recruiter, worker); (teams, chains).

5.1.2 Stimuli Selection

The alignment of the bad hypothesis output from the ASR system with the reference transcripts (in section 5.1.1 above) provided us with the list of confusion pairs, with each pair corresponding to an independent error (no overlap in the time frames) the ASR system made. We note that the reference text and the list of confusion pairs can be thought of as specifying an entire "space" of possible ASR outputs: Considering the reference text as a starting point, and considering each confusion pair as an "insert an error" operator, one can imagine an entire network of possible ASR text outputs that are possible. Each ASR output contains some subset of the errors from the list of confusion pairs.

Given this space of possible ASR outputs, our goal is to identify two output texts for each reference text, with these properties:

- The output texts should reflect reasonable performance of a commercial ASR system in noise typical of a workplace setting when the speaker is not wearing a special headset microphone; so, we wanted to identify text candidates with WER of approximately 0.25 (ranging between 20% and 30%).
- We wanted to identify one text candidate that has a low WER-to-ACE ratio and another candidate with a high WER-to-ACE ratio. We selected two candidates with identical WER: one with a high ACE score, and the other with a low ACE score.

Thus, the two text candidates identified represent two possible outputs from an ASR system. The errors that appear in the texts are realistic: They were actual errors made by an ASR system, and the overall WER error rate for the sentences is approximately 0.25. We can think of one of these text candidates as being "preferred by WER" (the one with the low WER-to-ACE ratio), and the other as being "preferred by ACE" (with the high WER-to-ACE ratio).

We wrote code to execute a search procedure through the space of possibilities to identify a pair of text candidates that fit the above criteria. We executed this code on 45 text sentences that had been extracted from the verbatim script of what the human spoke in our business meeting videos, and we thereby obtained 45 pairs of ASR text output candidates (two per sentence). Example stimuli from are available here: http://latlab.ist.rit.edu/assets2017ace

[3]https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/