Predicting the Micro-Timing of User Input for an Incremental Spoken Dialogue System that Completes a User's Ongoing Turn

Timo Baumann
Department of Linguistics
Potsdam University
Germany
timo@ling.uni-potsdam.de

David Schlangen
Faculty of Linguistics and Literature
Bielefeld University
Germany
david.schlangen@uni-bielefeld.de

Abstract

We present the novel task of predicting temporal features of continuations of user input, while that input is still ongoing. We show that the remaining duration of an ongoing word, as well as the duration of the next can be predicted reasonably well, and we put this information to use in a system that synchronously completes a user's speech. While we focus on collaborative completions, the techniques presented here may also be useful for the alignment of back-channels and immediate turn-taking in an incremental SDS, or to synchronously monitor the user's speech fluency for other reasons.

1 Introduction

Turn completion, that is, finishing a user's ongoing utterance, can be considered an ideal test-case of *incremental* spoken language processing, as it requires that all levels of language understanding and production are carried out in real time, without any noticeable lags and with proper timing and even with the ability to predict what will come. Spoken dialogue systems, especially incremental ones, have come a long way towards reducing lags at turn changes (e. g. (Raux and Eskenazi, 2009; Skantze and Schlangen, 2009)), or even predicting upcoming turn changes (Schlangen, 2006; Baumann, 2008; Ward et al., 2010). Compared to regular turn changes, where short pauses or overlaps occur frequently (Weilhammer and Rabold, 2003), turn completions in natural dialogues are typically precisely aligned and prosodically highly integrated with the turn that is being completed (Local, 2007). With ever more incremental (and hence quicker) spoken dialogue systems, the phenomenon of completion comes into reach for SDSs, and hence questions of micro-timing become important.

While completing someone else's turn – especially for a computer – may be considered impolite or even annoying, *being able* to do so can be a useful capability. Some tasks where it might be helpful are

- negotiation training to induce stress in a human trainee as presented by DeVault et al. (2009), or
- pronunciation aids for language learners, in which hard to pronounce words could be spoken simultaneously by the system.

A system should certainly not try to complete all or even many user turns, but having the capability to do so means that the system has a very efficient interactional device at its disposal.

Furthermore, monitoring the user's timing, as is required for the temporal prediction of turn continuations, can also be used for other conversational tasks such as producing back-channels that are precisely aligned to the user's back-channel inviting cues, to enable micro-alignment of turn-onsets, or to quickly react to deviations in the user's fluency.

In this paper, we concentrate on the temporal aspects of turn completion, that is, the prediction of the precise temporal alignment of a turn continuation and the technical realization of this timing. We assume the task of predicting the completion itself to be handled by some other system component. Such components are indeed under development (see Section 2). However, previous work has left out the question of how the precise timing of turn completions can be accomplished, which is what we try to answer here.

The remainder of this paper is structured as follows: In Section 2 we review literature on turn completion and related work in spoken dialogue systems,

Proceedings of the SIGDIAL 2011: the 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 120–129,
Portland, Oregon, June 17-18, 2011. ©2011 Association for Computational Linguistics

before we explain what exactly our task is in Section 3. In Section 4 we present our system's overall architecture and the duration modelling technique that we use, before describing the corpus that we use in Section 5. In Section 6 we first analyse whether enough time to output a completion is available sufficiently often, before turning to the question for the actual sub-tasks of *when* and *how* to complete. We wrap up with concluding remarks and ideas for future work.

2 Related Work

The general phenomenon of turn completion can be broken down into cases where the completion is spoken simultaneously with the original speaker (*turn sharing*, (Lerner, 2002)) and where the floor changes in mid-utterance (*collaborative turn sequences* (Lerner, 2004) or *split utterances* (Purver et al., 2009)). In this paper, a differentiation between the two cases is not important, as we only deal with the question of when to start speaking (for the previously non-speaking system) and not the question of whether the current turn owner will stop speaking. Moreover, whether the other speaker will stop is beyond the system's control. Lerner (2004) distinguishes turn *co-optation*, in which a listener joins in to come first and win the floor, and turn *co-completion*, in which the completion is produced in chorus. Both of these phenomena relate to the current speaker's speech: either to match it, or to beat it. While we focus on matching in this work, the methods described similarly apply to co-optation.

As Lerner (2002) notes, attributing this view to Sacks et al. (1974), simultaneous speech in conversation is often treated exclusively as a turn taking problem in need of repair. This is exactly the point of view taken by current spoken dialogue systems, which avoid overlap and interpret all simultaneous speech as *barge-in*, regardless of content. However, Lerner (2002) also notes that simultaneous speech systematically occurs without being perceived as a problem, e. g. in greetings, or when saying good bye, which are relevant sub-tasks in deployed SDSs.

Two corpus studies are available which investigate split utterances and their frequency: Skuplik (1999) looked at *sentence cooperations* in a corpus of task-oriented German (Poesio and Rieser, 2010) and found 3.4 % of such utterances. Purver et al. (2009) find 2.8 % of utterance boundaries in the BNC (as annotated by Fernández and Ginzburg (2002)) to meet their definition of utterances split between speakers. Thus, while the absolute frequency may seem low, the phenomenon does seem to occur consistently across different languages and corpora.

Local (2007) describes phonetic characteristics at utterance splits (he calls the phenomenon *turn co-construction*) which distinguish them from regular turn handovers, namely temporal alignment and close prosodic integration with the previous speaker's utterance. In this paper, we focus on the temporal aspects (both alignment and speech rate) when realizing turn completions, leaving pitch integration to future work.

Cummins (2009) analyses speech read aloud by two subjects at the same time (which he calls *synchronous speech*): Synchrony is slightly better in a live setting than with a subject synchronizing to a recording of speech which was itself spoken in synchrony and this is easier than to a recording of unconstrained speech. Cummins (2009) also experiments with reduced stimuli: eliminating f_0-contour had no significant impact on synchrony, while a carrier without segmental information (but including f_0-contour) fared significantly better than speaking to an uninformative hiss. (The first sentence of each recording was always left unmodified, allowing subjects to estimate speech rate even in the HISS condition.) Thus, pitch information does not seem necessary for the task but may help in the absence of segmental information.

On a more technical level and as mentioned above, much work has been put into speeding up end-of-turn detection and reducing processing lags at turn changes (Raux and Eskenazi, 2009) and more recently into end-of-turn prediction: Ward et al. (2010) present a model of turn-taking which estimates the remaining duration of a currently ongoing turn. We extend the task to predicting the remaining duration of any currently ongoing *word* in the turn. Of course, for this to be possible, words must be recognized while they are still being uttered. We have previously shown (Baumann et al., 2009) that this can be achieved with incremental ASR for the vast majority of words and with an average of 102 ms between when a word is first recognized and the word's end.

As mentioned above, our work relies on other incremental components to form a meaningful, turn completing application and such components are being developed: Incremental understanding is well underway (Sagae et al., 2009; Heintze et al., 2010), as is decision making on whether full understanding of an utterance has been reached (DeVault et al., 2009), and Purver et al. (2011) present an incremental semantics component aimed explicitly at split utterances. In fact, DeVault et al. (2009) provide exactly the counterpart to our work, describing a method that, given the words of an ongoing utterance, decides when the point of maximum understanding has been reached and with what words this utterance is likely to end. However, in their system demonstration, Sagae et al. (2010) use short silence time-outs to trigger system responses. Our work eliminates the need for such time-outs.

Hirasawa et al. (1999) present a study where immediate, overlapping back-channel feedback from the system was found to be inferior to acknowledging information only after the user's turn. However, they disregarded the back-channels' micro-temporal alignment as explored in this study (presumably producing back-channels as early as possible), so their negative results cannot be taken as demonstrating a general shortcoming of the interactional strategy.

3 The Task

The general task that our timing component tackles is illustrated in Figure 1. The component is triggered into action when an understanding module signals that (and with what words) a turn should be completed. At this *decision point*, our component must estimate (a) *when* the current word ends and (b) *how* the user will speak the predicted continuation. Ideally, the system will start speaking the continuation precisely when the next word starts and match the user's speech as best as possible. Thus, our component must estimate the time between decision point and ideal onset (which we call *holding time*) and the user's *speech rate* during the following words.

In order for the system to be able to produce a continuation ("*five six seven*" in Figure 1) in time, of course the decision point must come sufficiently early (i. e. during "*four*") to allow for a completion to be output in due time. This important precondition must be met by-and-large by the employed ASR. However, it is not a strict requirement: If ASR results are lagging behind, the timing component's estimated holding time should turn negative. Depending on the estimated lag, a completion can be suppressed or, if it is small, fairly good completions can still be realized by shortening the first (few) phonemes of the completion to be synthesized.

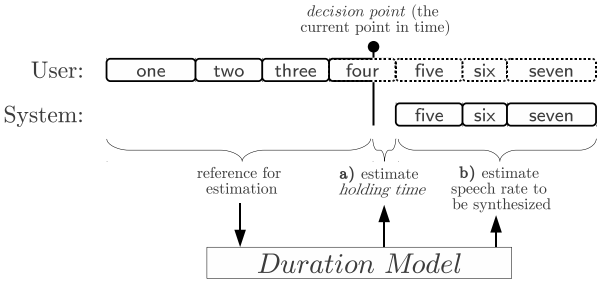


Figure 1: The task: When notified that the ongoing utterance should be completed with "*five six seven*" after the word "*four*", the first three words are used to (a) estimate the remaining duration of "*four*" and to (b) estimate the speech rate for the completion.

We will now present our overall system before describing two strategies we developed for solving the task just described, and further on present the experiments we conducted with the system and their results in Sections 5 and 6.

4 System Description

Our system is based on the InproTK toolkit for incremental spoken dialogue systems (Schlangen et al., 2010) which uses Sphinx-4 (Walker et al., 2004) and MaryTTS (Schröder and Trouvain, 2003) as underlying ASR and TTS engines, respectively. The core of our system is a component that incrementally receives rich speech recognition input (words, their durations and a pitch track) from an incremental ASR and computes the timing of completions.

When receiving a new word from ASR, our component queries an understanding component whether a completion can be predicted, and if so, whether such a completion should be performed. In order to not duplicate the work of DeVault et al. (2009), we use a mock implementation of an understanding module, which actually knows what words are going to be spoken (from a transcript file) and aims to complete after *every* word spoken.

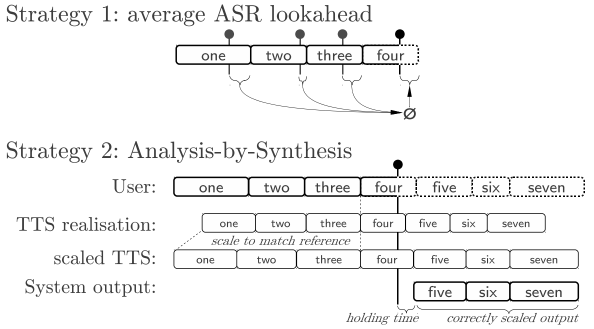
We have implemented two strategies for the timing module, which we will describe in turn, after first discussing a simple baseline approach.

Baseline: Speak Immediately A first, very simple approach for our timing component is to never wait between the decision point and outputting a completion right away. We believe that this was the strategy taken by Hirasawa et al. (1999) and we will show in our evaluation in Section 6.2 that it is not very good.

Strategy 1: Estimating ASR Lookahead In our ASR-based strategy (illustrated in Figure 2, top) the system estimates what we call its *lookahead* rate, i. e. the average time between when a word is first recognized by ASR and the word's end in the signal. This lookahead is known for the words that have been recognized so far and the average lookahead can then be used as an estimate of the remaining duration of the word that is currently being detected (i. e. its *holding time*). Once the currently spoken word is expected to end, the system should start to speak.

The strategy just described, as well as the baseline strategy, only solve half of the task, namely, when the continuation should be started, but not the question of *how* to speak, which we will turn to now. Both sub-tasks can be solved simultaneously by estimating the speech rate of the current speaker, based on what she already said so far, and considering this speech rate when synthesizing a completion. Speech rate estimation using some kind of duration model thus forms the second strategy's main component. For the purpose of this work, we focus on duration models in the context of TTS, where they are used to assign durations to the phones to be uttered. Rule-based approaches (Klatt, 1979) as well as methods using machine learning have been used (primarily CART (Breiman et al., 1984)); for HMM-based speech synthesis, durations can be generated from Gaussian probability density functions (PDFs) (Yoshimura et al., 1998). We are not aware of any work that uses duration models to predict the remaining time of an ongoing word or utterance.

In our task, we need the duration model to make estimations based on limited input (instead of providing plausibility ratings as in most ASR-related applications). As it turns out, a TTS system in itself is an excellent duration model because it potentially ponders all kinds of syntactic, lexical, post-lexical, phonological and prosodical context when assigning durations to words and their phones. Also, our task already involves a TTS system to synthesize the turn completion – in our case MaryTTS (Schröder and Trouvain, 2003). The durations can be accessed in symbolic form in MaryTTS, and the system allows to manipulate this information prior to acoustic synthesis. Depending on which voice is used, MaryTTS uses machine-learned duration models (CART or PDFs) or an optimized version of Klatt's (1979) rules which have been shown to perform only marginally worse than the CART-based approach (Brinckmann and Trouvain, 2003).


Strategy 1: average ASR lookahead

Strategy 2: Analysis-by-Synthesis

Figure 2: Our strategies to estimate holding time (*when* to speak), and speech rate (*how* to speak; only Strategy 2).

Strategy 2: Analysis-by-Synthesis As just described, we hence employ the TTS' duration model in an analysis-by-synthesis approach in this second strategy, as illustrated in Figure 2 (bottom): When triggered to complete an ongoing utterance, we query the TTS for the durations it would assign to a production of the predicted full utterance, i. e. the prefix that was heard plus the predicted continuation of the turn. In that way, the TTS can take the full utterance into account when assigning prosodic patterns which may influence durations. We then compute the factor that is needed to scale the TTS's duration of the words already finished by the user (in the example: "*one two three*") to the duration of the actual utterance and apply this scaling factor to the remaining words in the synthesized completion. We can then read off the expected duration of the currently spoken word from the scaled TTS output and, by subtracting the time that this word is already going on, find out the *holding time*. Similarly, the completion of the turn which is now scaled to match the user's speech rate can be fed back to the synthesis system in order to generate the acoustic waveform which is to be output to the speakers once the system should start to speak.

5 Corpus and Experiment Setup

In order to evaluate the accuracy of the individual components involved in the specific subtasks, we conducted a controlled offline experiment. We have not yet evaluated how actual users of our system would judge its performance at outputting collaborative completions.

As evaluation corpus we use recordings of the German version of the story *The North Wind and the Sun* (IPA, 1999) from the Kiel Corpus of Read Speech (IPDS, 1994). The story (including title) consists of 111 words and is read by 16 speakers, giving a total of 1776 words in 255 inter-pausal-units (IPUs), altogether resulting in about 12 minutes of speech. (In the following, we will equate "turns" with IPUs, as our corpus of read speech does not contain true turns.) Words and phones in our corpus have a mean/median/std dev duration of 319/290/171 ms and 78/69/40 ms, respectively.

We assume that every word can be a possible completion point in a real system, hence we evaluate the performance of our timing component for all words in the corpus. (This generalization may have an influence on our results: real collaborative completions are sometimes invited by the speaker, probably by giving cues that might simplify co-completion; if that is true, the version tackled here is actually harder than the real task.)

Good turn completions (and good timings) can probably only be expected in the light of high ASR performance. We trained a domain-specific language model (based on the test corpus) and used an acoustic model trained for conversational speech which was not specifically tuned for the task. The resulting WER is 4.2 %. While our results could hence be considered too optimistic, Baumann et al. (2009) showed that incremental metrics remained stable in the light of varying ASR performance. We expect that lower ASR performance would not radically change prediction quality itself; rather, it would have an impact on how often continuations could be predicted, since that is based on correct understanding of the prefix of the utterance, limiting the amount of data points for our statistics.

Even though we simulated the understanding and prediction module, we built in some constraints that are meant to be representative of real implementations of such a module: it can only find the right completion if the previous two words are recognized correctly and the overall WER is lower than 10 %. (Coming back to Figure 1, if the system had falsely recognized "*on two three*", no completion would take place: Even though the last two words "*two three*" were recognized correctly, the WER between "*on two three*" and "*one two three*" is too high.) Under this constraint, the timing component generated data for 1100 IPU-internal and 223 IPU-final words in our corpus.

The main focus of this paper is turn completion and completions can only take place if there is something left to complete (i. e. after turn-internal words). It is still useful to be able to predict the duration of turn-final words, though, as this is a prerequisite for the related task of timing speaker changes. For this reason, we include both turn-internal and turn-final words in the analyses in Section 6.2.

In the evaluation, we use the ASR's word alignments from recognition as gold standard (instead of e. g. hand-labelled timings), which are essentially equal to output from forced alignment. However, when evaluating how well our timing component predicts the following word's duration, we need that word to also be correctly recognized by ASR. This holds for 1045 words in our corpus, for which we report results in Section 6.3.

6 Results

We evaluate the timing of our system with regards to whether completions are possible in general, when a completion should be produced, and what the speech rate of the completion should be in the subsections below.

6.1 Availability of Time to Make a Decision

While it is strictly speaking not part of the timing component, a precondition to being able to speak *just-in-time* is to ponder this decision sufficiently early as outlined above.

Figure 3 shows a statistic of when our ASR first hypothesizes a correct word relative to the word's end (which can be determined post-hoc from the final recognition result) on the corpus. Most words are hypothesized before their actual endings, with a mean of 134 ms (median: 110 ms) ahead. This leaves

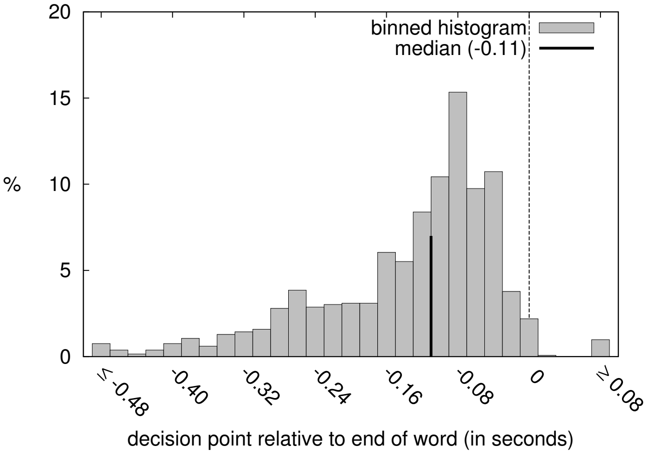


Figure 3: Statistics of when decisions can be first taken relative to the word's end (determined post-hoc).

model	error distribution metrics (in ms)			
	mean	median	std dev	MAE
baseline: all	-134	-110	107	110
baseline $-\mu$	0	23	107	63
ASR-based : all	-2	19	105	60
IPU-internal	26	33	82	51
IPU-final	-148	-143	87	142
TTS-based : all	-3	4	85	45
IPU-internal	12	11	77	41
IPU-final	-78	-76	83	79

Table 1: Descriptive statistics of the error distributions over estimated onset times for different duration models.

enough lookahead to synthesize a completion and for some delays that must be taken into account for input and output buffering in the sound card, which together take around 50 ms in our system.

Interestingly, lookahead differs widely for the speakers in our corpus with means between 97 and 237 ms. As can be seen in Figure 3, some words are only hypothesized *after the fact*, or at least too late to account for the inevitable lags, which renders impossible successful turn-completions following these words. However, the timing component should know when it is too late – the holding time should be negative – and could either not output the completion at this point or e. g. back off to setting in one or more phones or syllables later (actually, back off until the holding time turns positive).

6.2 When to Start Speaking

We evaluate the strategies from Section 4 by comparing the predicted holding times with the ideal holding time, i. e. the time necessary to match the ASR's lookahead.

Figure 3 can also be taken as depicting the error distribution of our baseline strategy to find out when to start a completion: on average, the completion will be early by 134 ms if it is uttered immediately and the distribution is somewhat skewed. An unbiased baseline strategy is obtained by subtracting the global mean from the holding times. This however requires the mean to be known in advance and is hence inflexible: the global mean may very well be different for other data sets as it already differs between speakers in our corpus. The two other strategies' error distributions are less skewed, so we just report the distributions' mean, median, and standard deviation,[1] as well as the median absolute error (MAE) for the ASR-based, the TTS-based and the baseline strategies in Table 1.

As can be seen in Table 1, both strategies are similarly effective in predicting the average remaining time of a currently uttered word, reducing the mean error close to zero, a significant improvement over starting a completion or next turn immediately. (ANOVA with post-hoc Tukey's honest significance differences test.) While our two approaches perform similarly when comparing the performance for all words, there actually are differences when looking separately at IPU-internal and IPU-final words. In both cases the TTS-based approach has a significantly lower bias (paired Student's t-tests, $p < 0.01$).

The bias of both strategies differs depending on whether the current word is IPU-internal or -final. We believe this to be due to final lengthening: phones are about 40 % longer in IPU-final words. This is not captured by the ASR-based strategy and the lengthening may be stronger than what is predicted by the pronunciation model of the TTS we use.

A low standard deviation of the error distribution is probably even more important than a low mean error, as it is variability, or *jitter*, that makes a system unpredictable to the user. While there is no significant improvement of the ASR-based approach over the baseline, the TTS-based approach significantly outperforms the other approaches with a 20 % re-

[1] We prefer to report mean and std dev for bias and jitter separately; notice that RMSE= $\sqrt{\mu^2 + \sigma^2}$.

task	error distribution metric (in ms)			
	mean	median	std dev	MAE
TTS-based : duration	-5	4	75	45
+ **ASR-based** : onset	26	33	82	51
= end of word	25	30	100	81
+ **TTS-based** : onset	12	11	77	41
= end of word	7	10	94	74

Table 2: Descriptive statistics of the error distributions for the first spoken word of a completion.

duction of jitter down to about the average phone's length (Browne-Forsythe's modified Levene's test, $p < 0.001$).

Regarding human performance in synchronous speech, Cummins (2002) reports an MAE of 30 ms for the synchronous condition. However, MAE increased to 56 ms when synchronizing to an (unsynchronously read) recording, a value which is in the range of our results (and with our system relying on similar input).

6.3 How to Speak

As explained in the task description, knowing when to speak is only one side of the medal, as a turn completion itself must be integrated with the previous speech in terms of duration, prosodic shape and loudness.

Only our TTS-based strategy is capable of outputting predictions for a future word; our ASR-based approach does not provide this information. However, both duration and onset estimation (the next onset is identical to the end of the current word as estimated in Section 6.2) together determine the error at the word's end. Hence, we report the error at the next word's end for the TTS strategy's duration estimate combined with both strategies' onset estimates in Table 2.

Duration prediction for the next word with the TTS-based strategy works similarly well as for ongoing words (as in Section 6.2), with an MAE of 45 ms (which is again in the range of human performance). However, for the next word's end to occur when the speaker's word ends, correct onset estimation is just as important. When we combine onset estimation with duration prediction, errors add up and hence the error for the next word's end is somewhat higher than for either of the tasks alone, with a standard deviation of 94 ms and an MAE of 74 ms for the TTS-based model, which again outperforms the ASR-based model.

So far, we have not evaluated the matching of prosodic characteristics such as loudness and intonation (nor implemented their prediction). We believe that simple matching (as we implemented for onset and speech rate) is not as good a starting point for these as they are more complex. Instead, we believe these phenomena to mostly depend on communicative function, e. g. a co-optation having a wide pitch-range and relatively high loudness regardless of the current speaker's speech. Additionally, pitch-range would have to be incrementally speaker-normalized which results in some implementation difficulties.[2]

7 Demo Application: Shadowing

To get a feeling for the complete system and to demonstrate that our timing component works on live input, we implemented a shadowing application which completes – or rather shadows – a user utterance word-by-word. Given the prediction for the next word's onset time and duration it prepares the output of that next word while the user is still speaking the preceding word. As the application expects to know what the user is going to speak, the user is currently limited to telling the story of *North Wind and the Sun*.

Two examples of shadowings are shown in Appendix A.[3] As can be seen in the screenshots, the decision points for all words are sufficiently early before the next word, allowing for the next word's output generation to take place. Overall, shadowing quality is good, with the exception of the second "*die*" in the second example. However, there is an ASR error directly following ("*aus*" instead of "*luft*") and the ASR's alignment quality for "*sonne die*" is already sub-optimal. Also, notice that the two words following the ASR error are not shadowed as per our error recovery strategy outlined in Section 5.

[2] Edlund and Heldner (2007) report that for a reliable pitch-range estimation 10 to 20 seconds of *voiced* speech and hence – in our view – twice the amount of audio is necessary. This would have reduced our corpus size by too much.

[3] Audio files of the examples are available at http://www.ling.uni-potsdam.de/~timo/pub/shadowing/.

8 Discussion and Future Work

We described the task of micro-timing, or micro-aligning a system response (in our case a turn completion and shadowing a speaker) to the user's speech based on incremental ASR output and with both ASR and symbolic TTS output as duration models to predict when and how a completion should be uttered.

We have shown first of all, that a completion is possible after most words, as an incremental ASR in a small-enough domain can have a sufficient lookahead. Additionally, we have shown that the TTS-based duration model is better than both the baseline and the ASR-based model. Both the next word's onset and duration can be predicted relatively well ($\sigma = 77$ ms and $\sigma = 75$ ms, respectively), and within the margin of human performance in synchronously reading speech. It is interesting to note here that synchronous speech is simplified in prosodic characteristics (Cummins, 2002), which presumably facilitates the task. Errors in speech rate estimation add up, so that the deviation at the next word's end is somewhat higher ($\sigma = 94$ ms). Deviation will likely increase for longer completions, underlining the need for an incremental speech synthesis system which should allow to instantly adapt output to changes in speech rate, content, and possibly sentiment of the other speaker.

Clearly, our duration modelling is rather simplistic and could likely be improved by combining ASR and TTS knowledge, more advanced (than a purely linear) mapping when calculating relative speech rate, integration of phonetic and prosodic features from the ASR, and possibly more. As currently implemented, improvements to the underlying TTS system (e. g. more "conversational" synthesis) should automatically improve our model. The TTS-based approach integrates additional, non-ASR knowledge, and hence it should be possible to single out those decision points after which a completion would be especially error-prone, trading coverage against quality of results. Initial experiments support this idea and we would like to extend it to a full error estimation capability.

We have focused the analysis of incrementally comparing expected to actual speech rate to the task of micro-aligning a turn-completion and shadowing a speaker. However, we believe that this capability can be used in a broad range of tasks, e. g. in combination with word-based end-of-turn detection (Atterer et al., 2008) to allow for swift turn taking.[4] In fact, precise micro-alignment of turn handovers could be used for controlled testing of linguistic/prosodic theory such as the oscillator model of the timing of turn-taking (Wilson and Wilson, 2005).

Finally, duration modelling can be used to quickly detect deviations in speech rate (which may indicate hesitations or planning problems of the user) *as they happen* (rather than post-hoc), allowing to take the speaker's fluency into account in understanding and turn-taking coordination as outlined by Clark (2002).

Acknowledgments

This work was funded by a DFG grant in the Emmy Noether programme. We wish to thank the anonymous reviewers for their very helpful comments.

References

Michaela Atterer, Timo Baumann, and David Schlangen. 2008. Towards incremental end-of-utterance detection in dialogue systems. In *Proceedings of Coling*, Manchester, UK.

Timo Baumann, Michaela Atterer, and David Schlangen. 2009. Assessing and improving the performance of speech recognition for incremental systems. In *Proceedings of NAACL*, Boulder, USA.

Timo Baumann. 2008. Simulating spoken dialogue with a focus on realistic turn-taking. In *Proceedings of the 13th ESSLLI Student Session*, Hamburg, Germany.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and regression trees*. Wadsworth, Monterey.

Caren Brinckmann and Jürgen Trouvain. 2003. The role of duration models and symbolic representation for timing in synthetic speech. *International Journal of Speech Technology*, 6(1):21–31.

Herbert H. Clark. 2002. Speaking in time. *Speech Communication*, 36(1):5–13.

Fred Cummins. 2002. On synchronous speech. *Acoustic Research Letters Online*, 3(1):7–11.

Fred Cummins. 2009. Rhythm as entrainment: The case of synchronous speech. *Journal of Phonetics*, 37(1):16–28.

[4] Additionally, both our models consistently under-estimate the duration of IPU-final words. It should be possible to turn this into a feature by monitoring whether a word actually has ended when it was predicted to end. If it is still ongoing, this may be an additional indicator that the word is turn-final.

David DeVault, Kenji Sagae, and David Traum. 2009. Can I finish? Learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of SIGDIAL*, London, UK.

Jens Edlund and Mattias Heldner. 2007. Underpinning /nailon/: Automatic Estimation of Pitch Range and Speaker Relative Pitch. In *Speaker Classification II*, volume 4441 of *LNCS*, pages 229–242. Springer.

Raquel Fernández and Jonathan Ginzburg. 2002. Non-sentential utterances: A corpus-based study. *Traitement automatique des languages*, 43(2):13–42.

Silvan Heintze, Timo Baumann, and David Schlangen. 2010. Comparing local and sequential models for statistical incremental natural language understanding. In *Proceedings of SIGDIAL*, Tokyo, Japan.

Jun-ichi Hirasawa, Mikio Nakano, Takeshi Kawabata, and Kiyoaki Aikawa. 1999. Effects of system barge-in responses on user impressions. In *Proceedings of Eurospeech*, Budapest, Hungary.

International Phonetic Association, IPA. 1999. *Handbook of the International Phonetic Association*. Cambridge University Press.

Institut für Phonetik und digitale Sprachverarbeitung, IPDS. 1994. The Kiel corpus of read speech. CD-ROM.

Dennis H. Klatt. 1979. Synthesis by rule of segmental durations in English sentences. *Frontiers of Speech Communication Research*, pages 287–299.

Gene H. Lerner. 2002. Turn sharing: The choral co-production of talk in interaction. In C. Ford, B. Fox, and S. Thompson, editors, *The Language of Turn and Sequence*, chapter 9. Oxford University Press.

Gene H. Lerner. 2004. Collaborative turn sequences. In Gene H. Lerner, editor, *Conversation Analysis: Studies from the First Generation*, Pragmatics & Beyond, pages 225–256. John Benjamins, Amsterdam.

John Local. 2007. Phonetic detail and the organisation of talk-in-interaction. In *Proceedings of the 16th ICPhS*, Saarbrücken, Germany.

Massimo Poesio and Hannes Rieser. 2010. Completions, coordination, and alignment in dialogue. *Dialogue and Discourse*, 1(1):1–89.

Matthew Purver, Christine Howes, Patrick G. T. Healey, and Eleni Gregoromichelaki. 2009. Split utterances in dialogue: a corpus study. In *Proceedings of SIGDIAL*, London, UK.

Matthew Purver, Arash Eshghi, and Julian Hough. 2011. Incremental semantic construction in a dialogue system. In *Proceedings of the 9th IWCS*, Oxford, UK.

Antoine Raux and Maxine Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *Proceedings of NAACL*, Boulder, USA.

Harvey Sacks, Emanuel A. Schegloff, and Gail A. Jefferson. 1974. A simplest systematic for the organization of turn-taking in conversation. *Language*, 50:735–996.

Kenji Sagae, Gwen Christian, David DeVault, and David Traum. 2009. Towards natural language understanding of partial speech recognition results in dialogue systems. In *Proceedings of NAACL*, Boulder, USA.

Kenji Sagae, David DeVault, and David Traum. 2010. Interpretation of partial utterances in virtual human dialogue systems. In *Proceedings of NAACL*.

David Schlangen, Timo Baumann, Hendrik Buschmeier, Okko Buß, Stefan Kopp, Gabriel Skantze, and Ramin Yaghoubzadeh. 2010. Middleware for incremental processing in conversational agents. In *Proceedings of SIGDIAL*, Tokyo, Japan.

David Schlangen. 2006. From reaction to prediction: Experiments with computational models of turn-taking. In *Proceedings of Interspeech*, Pittsburgh, USA.

Marc Schröder and Jürgen Trouvain. 2003. The German text-to-speech synthesis system MARY: A tool for research, development and teaching. *International Journal of Speech Technology*, 6(3):365–377.

Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of EACL*, Athens, Greece.

Kristina Skuplik. 1999. Satzkooperationen. Definition und empirische Untersuchung. Technical Report 1999/03, SFB 360, Universität Bielefeld.

Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. 2004. Sphinx-4: A flexible open source framework for speech recognition. Technical Report SMLI TR2004-0811, Sun Microsystems Inc.

Nigel Ward, Olac Fuentes, and Alejandro Vega. 2010. Dialog prediction for a general model of turn-taking. In *Proceedings of Interspeech*, Tokyo, Japan.

Karl Weilhammer and Susen Rabold. 2003. Durational aspects in turn taking. In *Proceedings of the 15th ICPhS*, Barcelona, Spain.

Margaret Wilson and Thomas P. Wilson. 2005. An oscillator model of the timing of turn-taking. *Psychonomic Bulletin & Review*, 12(6):957–968.

Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. 1998. Duration modeling for HMM-based speech synthesis. In *Proceedings of the 5th ICSLP*, Sydney, Australia.

Appendix A Examples of Shadowing

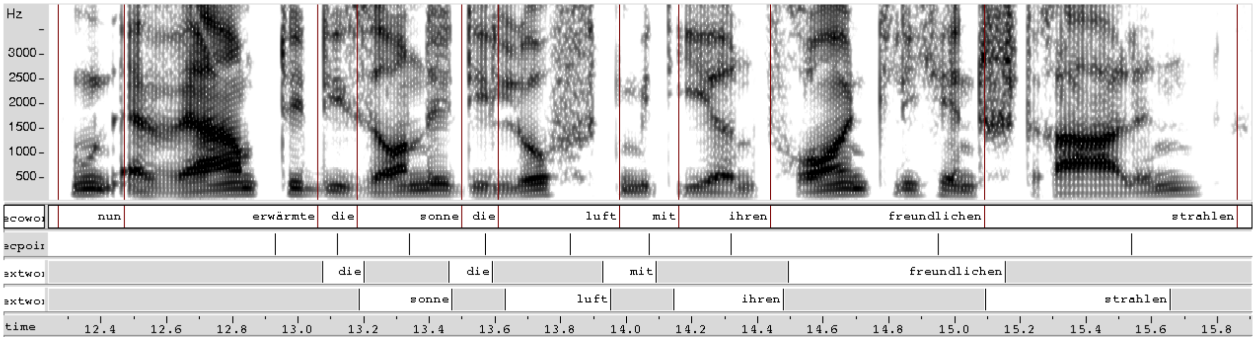


Figure 4: Example of shadowing for a file in our corpus (k73nord2). The first line of labels shows the final ASR output, the second line shows the decision points for each word and the third and fourth lines show the system's output (planned output may overlap, hence two lines; in the system, an overlapped portion of a word is replaced by the following word's audio).

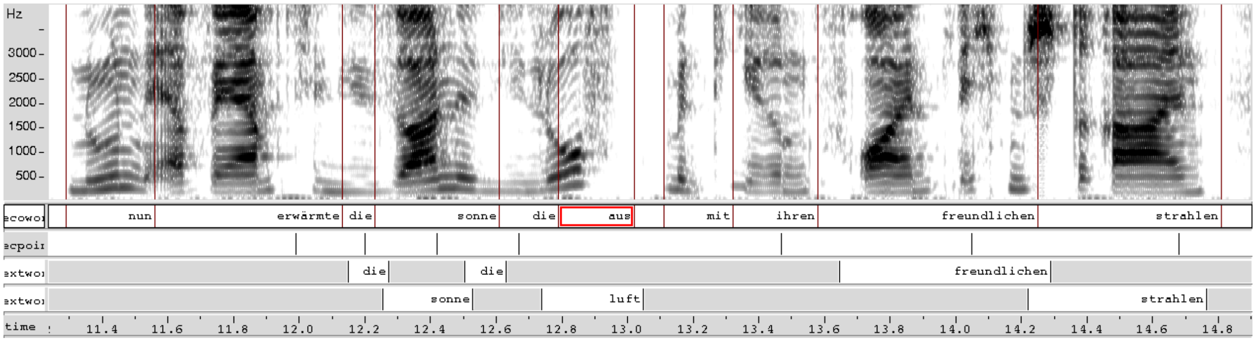


Figure 5: Example of shadowing with live input (verena2nord2). Notice that "*Luft*" is predicted and synthesized although it is (later) misunderstood by ASR as "*aus*", resulting in a missing shadowing of "*mit*" and "*ihren*". In order to not disturb the speaker, the system's audio output was muted.