Tale following: Real-time speech recognition applied to live performance

Jean-Luc Rouas, Boris Mansencal, Joseph Larralde

Tale following: real-time speech recognition applied to live performance

Jean-Luc Rouas
CNRS - LaBRI
UMR 5800
F-33400 Talence, France
rouas@labri.fr

Boris Mansencal
Univ. Bordeaux - LaBRI
UMR 5800
F-33400 Talence, France
mansenca@labri.fr

Joseph Larralde
Univ. Bordeaux - LaBRI
UMR 5800
F-33400 Talence, France
larralde@labri.fr

ABSTRACT

This paper describes a system for tale following, that is to say speaker-independent but text-dependent speech recognition followed by automatic alignment. The aim of this system is to follow in real-time the progress of actors reading a text in order to automatically trigger audio events. The speech recognition engine used is the well known Sphinx from CMU. We used the real-time implementation pocketsphinx, based on sphinx II, with the French acoustic models developed at LIUM.

Extensive testing using 21 speakers from the PFC corpus (excerpts in “standard french”) shows that decent performances are obtained by the system – around 30% Word Error Rate (WER). However, testing using a recording during the rehearsals shows that in real conditions, the performance is a bit worse : the WER is 40%.

Thus, the strategy we devised for our final application includes the use of a constrained automatic alignment algorithm. The aligner is derived from a biological DNA sequences analysis algorithm.

Using the whole system, the experiments report that events are triggered with an average delay of 9 s (± 8 s).

The system is integrated into a widely used real-time sound processing software, Max/MSP, which is here used to trigger audio events, but could also be used to trigger other kinds of events such as lights, videos, etc.

Index Terms: tale following, text-dependent speech recognition, real-time, live performance

1. INTRODUCTION

This paper describes the application of a speech recognition system to a live performance. The kind of live performance we are interested in in this case involves acting and musical interpretation. There may be several actors – and they may speak at the same time – but the text should be previously known, but not necessarily in the exact interpretation – we indeed want the actors to have some acting freedom.

The aim of this project is to equip the computer with a *tale follower* which listens to the actors’ performance in order to trigger basic audio events in real time. This performance situation problem is related to the well-known score following problem in the domain of computer music [1–4].

Thus, the designed system may be similar to what one would call *augmented tale telling*, where automatically triggered audio illustrations emphasise the actors’ performance. Using such a system, actors are not directed by the musical score but are in command of the show. Additionally, a musician or a band may improvise on the audio track triggered by the actors’ performance.

This paper is organised as follows: first, we present the motivations for building such a system and which requirements should be met. In section 3 we briefly describe the speech recognition engine and how we adapted it to our problem. The next section is dedicated to the text alignment procedure. The integration of the system into a audio processing environment is addressed in section 6. Application to a live performance is described in section 7.

2. OBJECTIVES

Score following has been studied since the early eighties in order to use the computer as a virtual musician able to play a score and accompany a musician in real time. The computer knows the score that is played by the musician and also knows the score it has to play to accompany the musician. Following the tempo of the musician, the computer anticipates the events coming from the musician in order to optimise the synchronisation between it and the musician, just like real musicians.

Our objective is to address the same problem by replacing the musician by an actor, thus considering as input the voice of an actor reading a text instead of a melody played by a musician. In our project, the computer knows the text that will be told by the actor as well as the score it has to play. The computer has to analyse the voice to extract phonemes, build words and align them with the text in order to know where the actor is situated in the text at any time.

The problems addressed are the following: efficiency, robustness, precision and reusability. Firstly, as the system is to be executed during live performances, it has to be efficient enough to work in real-time (as a score follower) for a good synchronisation between inputs and outputs – triggering a sound too early or too late is to be avoided at all cost. Secondly, in live situations, the system has to be robust to mispronunciations, forward jumps, as well as repetitions and everything that may occur in the context of actors interpretation. Thirdly, the precision of the temporal synchronisation between inputs and outputs depends of course on the algorithms global efficiency but also on the confidence of the alignment. For example, as will be detailed later in the paper, when the signal corresponds to a portion of a word, confidence is low, whereas confidence increases after the completion of a word or a sentence. Thus, this consideration provides constraints on the score to be played by the computer in order to optimise the precision of the synchronisation. Finally, we want to design a system as general as possible in order to be able to reuse it for quite different artistic contexts. For example, it has to be independent from the voice of the actor, whether male or female, so that different actors may perform on different occasions. It also has to be easily connected to the musical tools that are usually used in live performances in order not to add too much complexity for sound engineers or musicians.

3. SPEECH RECOGNITION

The speech recognition engine we choose to use is CMU Sphinx [5] for its real-time capabilities and the availability of French acoustic models. The aim of the system is to recognise words as they are said, find their position in the (known) text, and trigger audio events that are used to support the actors' playing and help the musician to focus on her improvisation.

3.1 Sphinx

We chose to use Sphinx as it is freely available and a real-time implementation – pocketsphinx – exists [6]. We will however not extensively describe the Sphinx system as it is fairly complex. Nevertheless, we remind that a speech recognition system has two main processing steps:

- An acoustic processing step, which uses phoneme models in order to transcribe audio features in a string of phones. The model used in this step is language-specific since phonemes differ in each language, and may or may not be speaker specific - it depends on how many speakers were used to train the models, though models may be speaker adapted in order to achieve better recognition.

- A "phoneme to words" step, which aims at transforming the string of phones into sequences of words. This step make use of a pronunciation dictionary, which indicates to the system how the words may be pronounced, and a language model which works as a sort of grammar by defining which word should follow another. The language model is usually trained on a large database of texts (i.e. newspaper articles) and statistics are extracted corresponding to the most frequent sequences of words. The length of those sequences may vary from one to three, we then speak of unigrams (word occurences statitics), bigrams (sequences of one or two words) or trigrams (sequences of one, two or three words).

The Sphinx system was used with success on French data during the ESTER evaluation [7] by the Laboratoire d'Informatique de l'Université du Maine (LIUM). The LIUM developed French acoustic and language models to be used with Sphinx for this evaluation [8]. They managed to achieve a performance of 18.2% WER on broadcast data from a number of television and radio channels [9]. The LIUM models are available both from the Sphinx repository on Sourceforge [1] and from the LIUM website [2]. It is worth noting that the LIUM French acoustic models are speaker independent.

3.2 Text Adaptation

In spite of these available models, we need to create our own language model and add some words and their pronunciation to the dictionary. There is indeed a specific vocabulary that may be used in poetry but that is not frequently found in the sources usually employed for training language models – i.e. newspapers.

3.3 Preliminary experiment

We wanted to test this system on a read speech corpus, the read text being the ideal case of a fake newspaper article read by several native speakers. The data we used come from the PFC (Phonologie du Français contemporain) Corpus [10]. The text is composed of 406 words and for this experiment we used a total of 21 speakers (11 female and 10 males) from the towns of Brecey and Brunoy which are usually used to represent "standard french". The total duration of the 21 files is approximately 57 minutes (i.e. 2:42 per file).

On these recordings, the system, without any adaptation, achieved surprisingly poor performances – 91.1% Word Error Rate (WER), as shown on the first line of table 1. These poor performances may be due to the fact that the LIUM-Sphinx system was trained for broadcast news transcription, which is particularly important considering the language model, which was trained on newspaper data (e.g. excerpts from "Le Monde"). However, the text used in this experiment, which is considered to be similar to a newspaper article, may not reflect well the training data used for the language model. It is also worth noting that the best performing LIUM system mentioned earlier in the paper is fairly complex since it makes use of a speaker segmentation algorithm, a 4-gram language model and works in several passes, which are options that we have not considered here due to real-time constraints.

Considering these facts, we therefore decided to train a new statistical language model on a combination of concatenations of phrases from the text. We did not use a fixed grammar as language model, because actors may change the text slightly during a live performance.

Using this language model, specific to the text of the poem, we managed to decrease the WER to 41.1%. Restricting the language model and dictionary to the original

[1] http://sourceforge.net/projects/cmusphinx/files/AcousticandLanguageModels/

[2] http://www-lium.univ-lemans.fr/en/content/data

Adaptation Method	Corr	Sub	Del	Ins	Err
none	9.5	30.1	60.4	0.7	91.1
LM only	59.9	3.9	36.2	1.4	41.5
MAP+LM	63.8	2.6	33.6	1.2	37.4
MAP+MLLR+LM	72.1	1.4	26.5	1.7	29.7

Table 1. Performance obtained using Sphinx with the LIUM acoustic models and different kinds of adaptation on PFC data

Take	Corr	Sub	Del	Ins	Err
#1	81.9	14.5	3.6	19.7	37.8
#2	79.8	15.6	4.6	20.6	40.7

Table 2. Performance obtained using Sphinx with the LIUM acoustic models on rehearsal data

text also has the welcome effect of speeding the speech recognition process. Besides, this result could probably be further improved by training an acoustic model specific to the speaker or channel. Using Maximum A Posteriori (MAP) adaptation combined with MLLR (Maximum Log Likelihood Ratio), we obtained a much more decent WER of 29.7%. The speaker/channel adaptation works quite well since recording conditions vary greatly in this database: the microphone is most of the time placed on a table and the room is not always very quiet.

3.4 Experiments using rehearsal data

The aim of our system is however to be speaker independent, since we may want to switch actors if necessary. There should not be any channel effect in our setup since we use close capture microphones.

Using the LM-only setup, we thus have tested the performance of the system on rehearsal data. The text of the poem is in that case told by the two actors (one male, one female) that will perform during the live show. We have recorded two sessions of the performance. Each recording has a duration of approximately 40 minutes, the theoretical length of the text being 1779 words. Since one of the aims of the system is to leave as much freedom as possible to the actors, they obviously took advantage of it. Instead of simply reading the poem, they played with it, sometimes speaking together, repeating words that were mentioned only once etc. The performance of the system is described in table 2. Even though we did not experiment with several actors, we are confident that the results should be similar with any interpreter since the acoustic models were not adapted here.

The performances are surprisingly better than with the PFC data in terms of number of correctly transcribed words (around 80%, to be compared with 70%) , but the number of insertions (20%) is much greater than in the previous test (around 1.2%). Although some of these insertions may be caused by the actors interpretation, this is quite unfortunate because we certainly do not want to trigger an event at a wrong time. The figure for the deletions is however much better on the rehearsal recordings than on the PFC data, which is encouraging.

These results show that we cannot rely only on the speech recognition alone to perform the task we want: the quite high error rate will certainly have some undesired consequences on the triggering of the events. Thus, we have chosen to use an algorithm for automatically aligning the recognised words with the text of the poem that allow for incomplete matching. This algorithm is described in the next section.

4. ALIGNMENT

The alignment algorithm is issued from research on DNA sequences. The starting point is the algorithm described in [11], which allows to transform a character string u in a string v using different operations: insertions, deletions, substitutions. Dynamic programming is used to find the optimal series of operations. As an example, the two following sequences can be aligned using this algorithm :

```
A T − G T T A T
A T C G T − A C
```

The algorithm from [12] works on the same principle but at a local level. This algorithm can find the two sub-strings of stronger similarity as in the example below:

```
G T G G A T − G T T A T G T G G
C C A C A T C G T − A C A A C A
```

It has been successfully applied on audio data for music similarity purposes [13].

We decided to apply this algorithm to our problem as the output from the speech recognition system might not be the exact researched text.

As input to the alignment procedure, we use both the recognised text and the confidence score given by the speech recognition system. If this recognition score is over a certain threshold, we use the algorithm to see if the recognised text may be aligned with the original text. To this end, we define a search window on the original text, which has a size proportional to the recognised sentence length. This window is used to restrict the search space for the alignment. The best approximate match, given by [12] algorithm, gives an alignment score. If this alignment score is greater than a second threshold, we consider that we have a valid alignment of a valid recognition and thus advance the start of search window to the end of the aligned sequence. If the alignment score is below this threshold, the size of the search window is increased without changing its starting point. This procedure is designed for the reading of a text: the progression of the reading should be linear – i.e. the reader must not go back to a previous element in the text.

5. EVENTS TRIGGERING PERFORMANCE EVALUATION

The timing of the events that the system should trigger is a very important point in our system. In order to assess that every event we designed to be triggered is effectively detected by the system, we have measured the time delay

Take	Average Delay (s)	Standard Deviation (s)
#1	9.20 (s)	± 8.01 (s)
#2	10.42 (s)	± 12.04 (s)

Table 3. Average delay for ten events on the two rehearsal recordings (average for four trials)

between the real occurrence of the triggering word in the recording and the time at which this word is effectively detected by the system.

To do this, we devised a list of ten words distributed along the text on which we measured the mean square of the delays. The result of this test, for the two rehearsal recordings is given in table 3. Since the performances may vary slightly between two tests using the same recording, the measurements are averaged on 4 trials.

As seen on table 3, the average delay is quite important. However, the delay can vary greatly between events, as illustrated by the confidence intervals. For instance, the best performing trial on both recordings is given in table 4. Note that the measured delays may be negative – the event is triggered before the word is actually pronounced – because of errors in the recognised stream of words leading to a false alignment.

The difficult passages are indeed mainly linked to moments where the actors play a lot with the text. We hope to improve that point in the future, but the performances are very dependent on the actors pronunciation and interpretation. Thus, in the actual state of the system, we had to select the proper words to trigger events efficiently.

word#	Take 1 delay (s)	Take 2 delay (s)
#1	10.936	-0.510
#2	8.038	-28.745
#3	1.818	-1.036
#4	21.446	-11.447
#5	0.185	0.915
#6	0.901	0.448
#7	0.807	0.953
#8	4.589	9.749
#9	21.135	26.157
#10	18.469	13.064

Table 4. Delay measured for each triggering word on take 1 and take 2

6. INTEGRATION WITH A MUSIC PROCESSING SOFTWARE

Max/MSP is a visual data-flow programming language which is widely used to program sound and music processing for live performances. Indeed, most of the interactive music composers consider that this language is the standard to process music in real time. Thus, we made a connection between our tale follower and Max/MSP to integrate it in a convenient environment and we developed our own sound modules in MAX.

The integration of the system has been achieved using the framework described on figure 1. Two HF microphones are used as input for a first computer (noted “Text follower” on the figure). The microphones are connected to the sphinx engine via Jack. Recognised words are then fed into the text aligner which indicates the progress of the reading. This information is then transferred via OSC to an audio processing second computer running Max/MSP. On this computer, switches triggering events are activated according to the received information and a cue list.

The sound coming from the instrument(s) played by the musician(s) is also processed using the second computer, which renders different effects and spread the sound on eight loudspeakers – the effects and the sound spatialisation characteristics may also be changed using the information from the tale follower).

7. THE FLUXUS SHOW

The whole system has been used for two performances of “Fluxus”. “Fluxus” is the name of a poem in French by author Donatien Garnier. The performance consisted in the reading of this poem by two actors (a man and a woman), and the playing of a musical accompaniment by a musician and a computer. The musical part played by the computer was determined before the performance whereas the musician improvised his own part.

Composer and musician György Kùrtag Jr. composed a specific music for this poem. The musical illustrations designed to be played by the computer were previously recorded in a studio. The musician was also present and improvising during the show. Although we still have troubles with the accuracy of the speech recognition system, with carefully chosen target words, the system did perform almost flawlessly and the performance was a success – though this is unfortunately not quantifiable.

8. CONCLUSION

In this paper, we have described a *tale-following* system based on a speech recognition system and an automatic aligner. The performances of the speech recognition system in text-dependent mode are quite average – around 40 % WER – in studio conditions. These numbers are quite different from the results obtained by LIUM using the same acoustic models because we used the real-time implementation of Sphinx – pocketsphinx – and 3-gram language models (as opposed to 4-grams). Using the “regular” sphinx implementation (of sphinx III), we managed to obtain a WER of 6.4% on the PFC data – but not in real-time. This is to be compared to the 29.7% WER obtained using the same training method with pocketsphinx. We will have to investigate why the performance gap is so important between the two implementations.

Anyway, as we needed real-time speech recognition, the system takes advantage of a post-processing using an automatic alignment algorithm designed to be able to cope with these errors. But, even with the complete system, trial events were triggered with a delay that can vary from 0.1 to 20 seconds. Extensive testing allowed us to choose the most appropriate words to trigger the events, providing a great experience from the audience point of view.

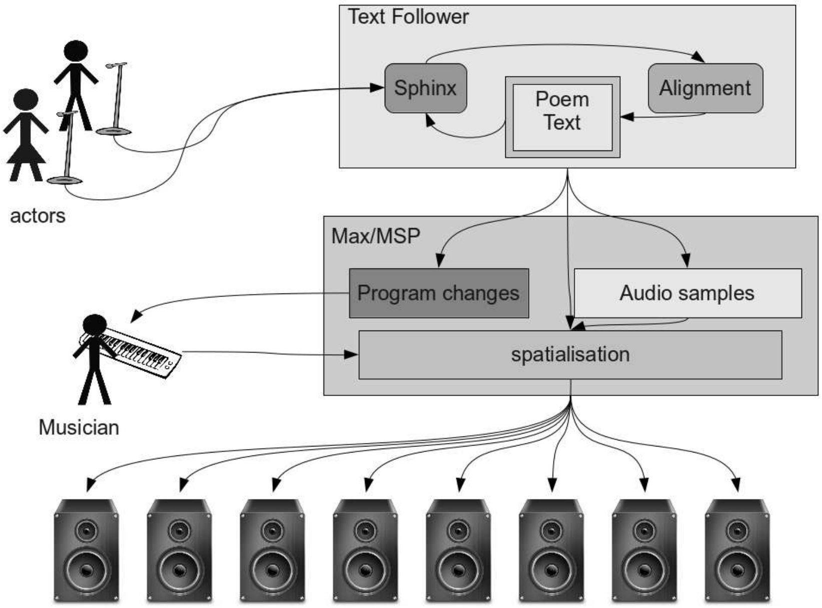

Figure 1. integration of the system

9. PERSPECTIVES

In this project, we have addressed only the temporal synchronisation between actors and musical accompaniment. An interesting perspective of this work is to extract voice and interpretation characteristics in order to use them for shaping the musical part. For example, we can consider adapting the energy of the music to the volume of the voice to enhance emotional impact. Intonation and speech rate could be used to modify musical and sound parameters and musical tempo. Effects could also be applied to the voice to transform it depending on the words and the sentences that are said by actors. For this purpose, we plan to use the iscore interactive sequencer [14, 15] which was developed during the virage project [16] to make the definition of the temporal organisation of musical events and the interconnection of different processes easier.

We also plan to adapt this system to singing voice following. The problem of speech recognition on singing voice is however still a challenge. We will adapt the speech recognition system to the singing voice by using specific data, but the intrinsic models may also need to be modified as singing voice characteristics differ from speech, particularly in terms of vowel durations, articulatory strategies and formant spanning. Nevertheless, by knowing the lyrics beforehand, we hope to be able to design an efficient system with real-time capabilities.

10. ACKNOWLEDGEMENTS

This work is partly supported by a grant from the ANR (Agence Nationale de la Recherche) with reference ANR-12-CORD-0022.

This research was carried out in the context of the SCRIME project (Studio de Création et de Recherche en Informatique et Musique Electroacoustique – Electroacoustic Music and Computer Science Research and Creation Studio, `scrime.labri.fr`) which is funded by the DGCA of the French Culture Ministry and the Aquitaine Regional Council. The SCRIME project is the result of a cooperation convention between the Conservatoire of Bordeaux, ENSEIRB-Matmeca (electronic and computer scientist engineering school) and the Bordeaux University of Sciences. It involves both electroacoustic music composers and scientific researchers and is managed by the LaBRI (Computer Science Research Laboratory at Bordeaux University, `www.labri.fr`). Its main missions are research and creation, diffusion and pedagogy thus extending its influence.

11. REFERENCES

[1] B. Vercoe, “The synthetic performer in the context of live performance,” in *Proceedings of International Computer Music Conference (ICMC)*, 1984.

[2] M. Puckette and C. Lipp, “Score following in practice,” in *Proceedings of International Computer Music Conference (ICMC)*, 1992.

[3] A. Cont, “Antescofo: Anticipatory synchronization and control of interactive parameters in computer music,” in *Proceedings of International Computer Music Conference (ICMC)*, 2008.

[4] N. Orio, S. Lemouton, and D. Schwarz, “Score following: State of the art and new developments,” in *In New Interfaces for Musical Expression (NIME)*, 2003.

[5] P. Lamere, P. Kwok, W. Walker, E. Gouvêa, R. Singh, B. Raj, and P. Wolf, “Design of the cmu sphinx-4 de-

coder," in *8th European Conf. on Speech Communication and Technology (EUROSPEECH 2003)*, 2003.

[6] H. D. Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2006.

[7] G. Gravier, J. Bonastre, S. Galliano, E. Geoffrois, K. M. Tait, and K. Choukri, "The ester evaluation campaign of rich transcription of french broadcast news," in *Language Resources and Evaluation Conference (LREC)*, 2004.

[8] P. Deléglise, Y. Estève, S. Meignier, and T. Merlin, "The lium speech transcription system: a cmu sphinx iii-based system for french broadcast news," in *9th European Conf. on Speech Communication and Technology (INTERSPEECH'2005 - EUROSPEECH)*, 2005.

[9] S. Galliano, E. Geoffrois, G. Gravier, J.-F. Bonastre, D. Mostefa, and K. Choukri, "Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news," in *Language Resources and Evaluation Conference (LREC)*, 2006.

[10] J. Durand, B. Laks, and C. Lyche, *Phonologie, variation et accents du français*. Hermès, 2009, ch. Le projet PFC: une source de données primaires structurées, pp. 19–61.

[11] S. B. Needleman and C. D. Wunch, "A general method applicable to the search for similarities in the amino acid sequences of two proteins," *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.

[12] T. Smith and M. S. Waterman, "Identification of commuon molecular subsequences," *Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981.

[13] J. Allali, P. Ferraro, P. Hanna, and M. Robine, "Polyphonic alignment algorithms for symbolic music retrieval," *Lecture Notes in Computer Science*, vol. 5954, pp. 466–482, 2010.

[14] M. Desainte-Catherine and A. Allombert, "Interactive scores: a model for specifying temporal relations between interactive and static events," *Journal of New Music Research*, vol. 34, pp. 361–375, 2005.

[15] A. Allombert, M. Desainte-Catherine, and G. Assayag, "Iscore : Writing the interaction," in *Proceedings of the 3rd Digital Interactive Media in Entertainment and Art (DIMEA)*, 2008.

[16] A. Allombert, P. Baltazar, R. Marczak, and M. Desainte-Catherine, "Virage : Designing an interactive intermedia sequencer from users requirements and theoretical background," in *International Computer Music Conference (ICMC)*, 2010.