INTERSPEECH 2009 BRIGHTON

Real-Time Correction of Closed-Captions

Patrick Cardinal, Gilles Boulianne

Centre de Recherche Informatique de Montréal (CRIM), Montréal, Canada

{patrick.cardinal,gilles.boulianne}@crim.ca

Abstract

Live closed-captions for deaf and hard of hearing audiences are currently produced by stenographers, or by voice writers using speech recognition. Both techniques can produce captions with errors. We are currently developing a correction module that allows a user to intercept the real-time caption stream and correct it before it is broadcast. We report results of preliminary experiments on correction rate and actual user performance using a prototype correction module connected to the output of a speech recognition captioning system.

Index Terms: Real-time correction, closed-captions.

1. Introduction

Speech recognition technology has widely contributed to improving the quality of life of the deaf and hearing-impaired population. An example is CRIM's automatic speech recognition system that has been applied to live closed-captioning of French-Canadian television broadcasts [1]. Our technology has been deemed acceptable by several Canadian broadcasters (RDS,CPAC,GTVA and TQS) who have adopted it over the past few years for captioning sports, public affairs and newscasts. The marked success of our approach rests on a low error rate, that depends notably on the integration of the re-speak method [2] for a controlled acoustic environment, automatic speaker adaptation and dynamic updates of language models and vocabularies.

In spite of the real-time constraints involved in these applications, satisfactory performances are obtained: average word accuracies for hockey games, House of Commons debates and news broadcasts are, respectively, 95%, 94% and 89%. Moreover, the closed-captions of all these projects were presented to panels of deaf and hard-of-hearing viewers and received favorable comments. Nevertheless, there is room for improvement.

Another area of application that will prove beneficial to the hearing-impaired community is phone services, that will allow access to many common services such as hotel room reservation or any other day to day communication which can be impossible if the proper equipment is not available. A possible approach to this problem requires the use of a third party to relay information between the hearing-impaired individual and the employee that books the reservation: the impaired individual may use a teletype machine to communicate the request to the third party, who in turn verbally relays the information to the hotel employee. The third party must relay back information to the impaired individual through a manual transcription. Clearly, this is a painstaking process associated with an unduly waste of time, since a simple hotel reservation can take up to 30 minutes. Here, an automatic speech recognition system could be used for cutting down transcription time. Also, in this situation, transcription errors are unforgivable.

Both of these applications have different requirements. In closed-captioning, the delay must be minimized, at the expense of a lower accuracy in transcription. In the telephone relay system, it is acceptable to tolerate a longer delay, but the error rate should be pratically zero. To this end, we are currently developing a real-time correction interface. In essence, this interface allows a user to intercept, in real-time, the word stream produced by speech recognition, so it can be corrected before being sent to the output device.

This paper is organized as follows. In section 2, we begin by offering a brief description of existing real-time correction systems; section 3 describes the weaknesses associated with our preliminary version of this application. The fourth section describes the interface and the major features of our approach, while section 5 shows experimental results from our system. We conclude by discussing future work.

2. Background

Real-time correction must be done within difficult constraints : with typical captioning rates of 130 words per minute, and 5 to 10% word error rate, the user must correct between 6 and 13 errors per minute. In addition, the process should not introduce more than a few seconds of additional delay over the 3 seconds already needed by respeaking and speech recognition.

In a previous work, Wald *et al.* [3] explored how different input modalities could reduce the amount of time required for correction. The authors describe 3 different interfaces for experimenting different ways of selecting an erroneous word. The first approach uses the standard mouse/keyboard combination which allows the user to select a word with the mouse pointer and to edit it with the keyboard. The second approach uses only the keyboard for both selection and editing to avoid alternating between different input devices. In their last approach, they present words in a table for which every column is associated with a function key. A word in a specific row can be reached by multiple presses of the "column" key.

They also suggest the use of 'hot' keys for easily correcting some types of errors such as word plurality or verb tenses.

They have experimented their interfaces on five users. The results show that the preferred interface was the traditional mouse/keyboard combination. However, they suggest that every user could become more efficient with any interface given adequate training.

In [4], the correction interface consisted in a scrolling window which can be edited by the user using a text editor style-interface. They introduced the idea of a controllable delay during which the text can be edited. They control the ouput delay by 3 scrolling modes: the elastic mode waits for the correction before sending the text while the bulldozer mode sends the text automatically, corrected or not. The third mode is a compromise between the two previous ones.

10.21437/Interspeech.2009-443

Our approach combines characteristics of the two previous systems. We use a delay parameter, which can be modified online, for controlling the output rate but the user can send out text as soon as it is corrected. We also use the standard mouse/keyboard combination for selecting and editing words. It's also possible to navigate through the application using only the keyboard.

To these functionalities, we added a list of alternate words that can be selected by a simple mouse click or a hot key to simplify the editing process and speed up correction time. Manual word editing is still available.

Another distinctive feature of our approach is the fixed word position: when a word appears on screen, it will remain in its position until it is sent out. This allows the user to focus on the words and not be distracted by word-scrolling or any other word movement.

3. A first version

In [5], we introduced our first version of a real-time, closed-captioning stream correction interface. Users have offered useful criticism of their experience with this original version. We will draw on these observations for the purpose of designing a more user-friendly and efficient version. Significant performance-hindering issues of the first version were:

1. Too much movement. In this first version, words were presented individually with a list of alternatives for each one. This approach takes up a lot of space on the screen. The number of words on the screen was limited and thus, the user would move his eyes between the bottom and top of the screen too often.
2. Free editing was possible, but difficult. Some types of correction were difficult. For example, when the speech recognition system replaces a word by many smaller words, the correction involved editing a word and suppressing the inserted ones. Since each word was in a different edit box, the operation took too long.
3. Fixed delay. Fixed delay was a major drawback. Some section of a transcription may contain numerous errors while another one may be nearly perfect. When there were too many errors, it was impossible to correct all of them and, conversely, when the transcription was very good, it was not possible to take advantage of it.
4. The interface supported only the mouse/keyboard combination to apply corrections; this mode of operation is not necessarily the most efficient for everybody.

4. Correction Software

The correction software allows editing the transcriptions by intercepting them while they are being sent to the external device (such as an encoder for a closed-captioning application). Both assisted and manual corrections can be applied to the word stream.

Assisted correction reduces the number of operations by presenting a list of alternate words for the word under consideration (the word with the caret on it), so that a correction can be done with a simple mouse click or by using the associated hot key. Manual correction requires editing the word to be changed and is more expensive in terms of delay. As a consequence, the number of these operations should be reduced to a strict minimum. However, it is not always possible to have the right word in the alternate word list, especially in the case of out-of-vocabulary words which are often replaced by a sequence of smaller words. In this situation, manual editing is the easier way to do the correction.

The user interface shown in Figure 1 has been designed with this consideration in mind. The principal characteristic of the interface is that there is no scrolling. Words never move; instead the screen is filled from left to right, top to bottom, with words coming from the speech recognition, in synchronisation with the audio. When the bottom right of the screen is reached, filling-in starts from the upper left corner again. Note that words are hidden after a certain delay. This feature avoids superposition of words on the screen while giving the necessary context for the user to make the correction.

Words are presented in an edition box which supports usual editing tools such as double-clicking to select a word. It is even possible to do copy & paste in the case of a very common error. Words appear in black while they are editable, and in gray (control is disabled) once they have been sent to the external device. Thus an active "window", corresponding to the interval during which words can be edited, moves across the word groups, while the words themselves remain fixed. Words sent to the external device are displayed in an output box (upper right corner of Figure 1).

Words are editable for a specific delay which may be adjusted according to the situation. However, the text is automatically sent to the external device as soon as the user has finished the verification/correction of a word group. A word group is considered to be completely verified when the user presses the enter key or when the caret is moved out of the edition box by using the arrow key. At the top right corner, the average delay is displayed. Depending on this value and the application, the user can bypass some word groups in order to decrease the accumulated delay.

The selected word can also be deleted by double-clicking on it followed by depressing the delete key. A shortcut, which deletes the word at the caret position has also been added for this purpose. Different shortcut corrections, as suggested in [3], can also be applied by using hot keys: the F1 and F2 keys change the gender (masculin or feminin) of the word while the F3 and F4 keys change the plurality (singular or plural) of the word. These available choices are in principle excluded from the list box choices. These shortcuts are also available on the keyboard by using the function keys.

Two users can run correction interfaces in parallel, on alternating sentences. This configuration avoids accumulation of delays. This functionality may prove useful if the word rate is so high that it becomes too difficult to keep track of the word flow. In this mode, the second user can begin the correction of a new sentence even if the first has not yet completed the correction of his/her sentence. Only one out of two sentences is editable by each user. The synchronisation is on a sentence basis.

4.1. Alternate word lists

The lists of alternate words are an important factor in the performance of our system. As described in the previous section, the gender/plurality forms of the word are implicitly included and accessible through a simple left/right mouse click or by pressing function keys on the keyboard. Other available forms explicitly appear in the list box. This approach has two major benefits. First, when a gender/plurality error is detected by the user, no delay is incurred from scanning the choices in the list box. Second, since the gender/plurality forms are not included in the

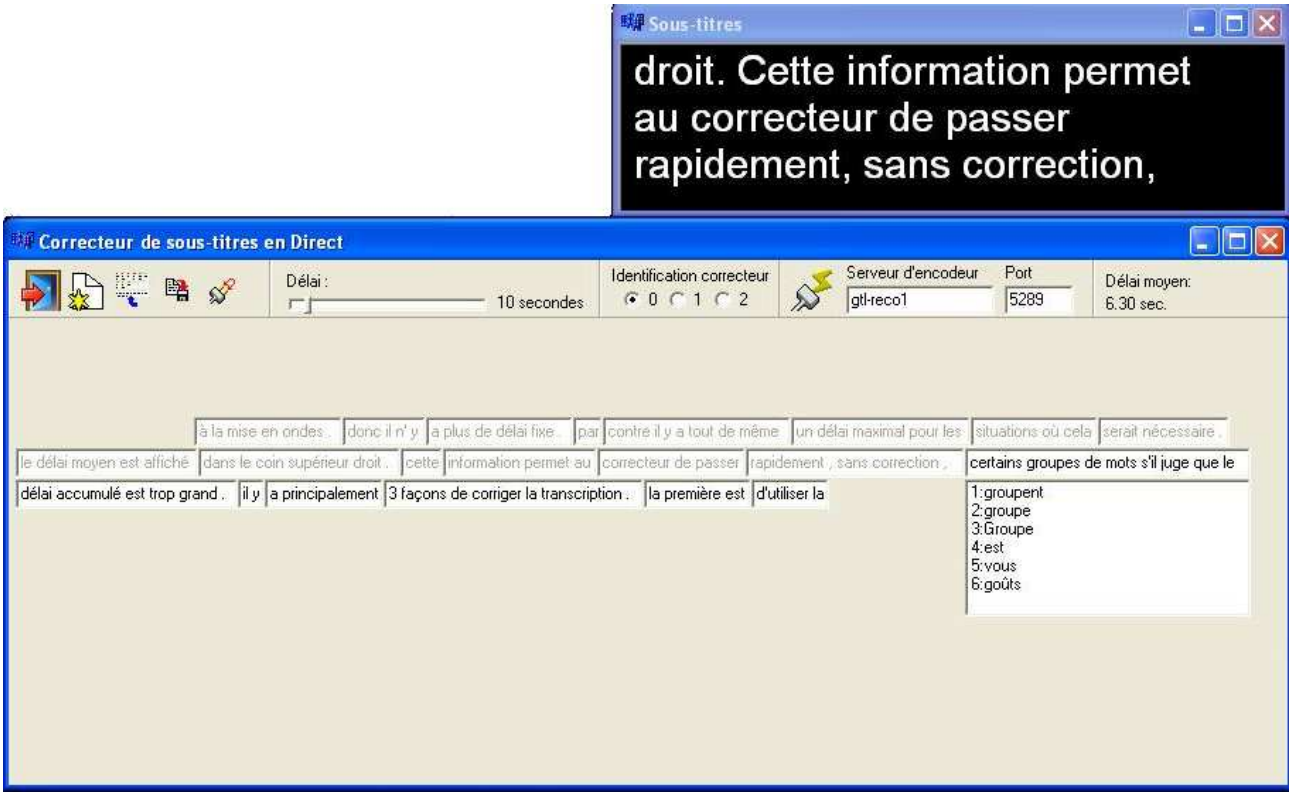

Figure 1: Real-time corrector software.

list box, their place becomes available for additional alternate words.

The main problem is to establish word lists short enough to reduce scanning time, but long enough to contain the correct form. For a given word output by the speech recognition system, the alternate words should be those that are most likely to be confused by the recognizer.

We experimented with two pre-computed sources of alternate word lists:

1. A list of frequently confused words was computed from all the available closed-captions of our speech recognition system for which corresponding exact transcriptions exist. The training and development sets were made up of 1.37M words and 0.17M words, respectively.
2. A phoneme based confusion matrix was used for scoring the alignment of each word of the vocabulary with every other word of the same vocabulary. The alignment program was an implementation of the standard dynamic programming technique for string alignment [6].

Each of these techniques yields a list of alternate words with probabilities based on substitution likelihoods. Table 1 shows how many times substitutions in the development set could be corrected with a word in the list, for each list and their combination.

To combine both lists, we take this coverage into consideration and the fact that 48% of the words were common to both lists. On this basis, we have constructed an alternate list of 10 words comprised of the most likely 7 words of case 1; the remaining 3 words are the most probable substitutions from the remaining words of both lists.

Source of alternates	coverage (%)
Word confusion matrix	52%
Phoneme confusion matrix	37%
Combined	60%

Table 1: *Coverage of substitutions on development set, for list of 10 words.*

5. Results

In this section we present the results of two preliminary experiments. In the first one, we simulated a perfect correction, as if the user had an infinite amount of time, to determine the best possible results that can be expected from the alternate word lists. In the second experiment, we submitted a prototype to users and collected performance measurements.

5.1. Simulation Results

The simulation is applied to a test set consisting of a 30 minute hockey game description for which closed-captions and exact transcripts are available. The test set is distinct from the training set and development set used in section 4.1. We aligned the produced closed-captions with their corrected transcripts and replaced any incorrect word by its correct counterpart if it appeared in the alternate list. In addition, all insertion errors were deleted. Table 2 shows the word error rate (WER) obtained for different alternate word lists.

The word confusion matrix captures most of the substitutions. This behavior was expected since the matrix has been

Source of alternates	WER
Original closed-captions	5.8%
Phoneme confusion matrix	4.4%
Word confusion matrix	3.1%
Combined	2.9%

Table 2: *Error rate for perfect correction.*

trained explicitely for that purpose. The performance should increase in the future as the amount of training data grows. In comparison, the contribution of words from the phoneme confusion matrix is clearly limited.

The corrected word was the first in the list 35% of the time, while it was in the first three 59% of the time. We also simulated the effect of collapsing words in insertion-substitution sequences to allow corrections of insertions : the increase in performance was less than 0.5% absolute.

5.2. User Tests

Experiments were performed by two unacquainted users of the system on closed-captions produced by our speech recognition system. They were instructed to operate as fast as possible without missing mispelled words. The experiments were performed on transcriptions produced for closed-captioning of public affair shows. The results are shown in Table 3

	User 1	User 2
Test duration (minutes)	59 min	35 min
Word rate (words/minute)	138 w/min	150 w/min
# of words	8122	5271
# of editions	274	221
WER before	6.6%	7.0%
WER after	3.2%	2.8%
Gain (relative %)	51%	60%
Delay (seconds)	9.4 sec	9 sec

Table 3: *Error rate before and after user correction.*

The results show that a significant WER decrease is achieved for an additional delay of approximatively 9 seconds. However, we can expect users to outperform these preliminary results with appropriate training.

Compared to our first version, the correction performance is similar but reduces the induced delay by 6 seconds.

5.3. User comments

Both users interact the same way with the interface. They use the mouse to select words and the keyboard to edit them. They intensively use the tabulation key to send out words before the delay is expired. However, they have pointed out that it is difficult to use hot keys but with practice, they estimate that time could be saved through their effective use. We anticipate that first-time users accustomed to keyboard interaction could exploit hot-keys more efficiently. On the other hand, both testers advantageously used the suggestion list to make corrections, especially for the purpose of correcting a grammatical errors.

Errors remaining after correction are mainly due to high word rate. Since they were required to minimize induced delay, they missed some misspelled words or spotted them after they had been sent out. In addition, correcting major errors such as out-of-vocabulary words takes a long time which causes them to lose track of the audio input. In situations where reducing delay is not so important, they anticipate a better correction rate.

It is important to note that this exercise imposes an important cognitive load on the user. A great deal of concentration is needed to follow the word stream. We observed that fatigue sets in after 15 minutes. Consequently, two correctors should be in place in order to relay the task. However, they feel that this load could become less important with practice but the presence of two correctors will probably remain unavoidable.

Finally, users commented on how they were more confortable with this new interface compared to the older one.

6. Conclusion and Future Work

We are currently developing a user interface for correcting live closed-captions in real-time. The interface presents a list of alternatives for each automatically generated word. When larger delays are allowed, manual edition of words for which there is no acceptable suggested alternative can yield further improvements.

We tested the application for real-time text correction produced in a real-world application with two users. The WER dropped from 7.0% to 2.8% and 6.6% to 3.2% with an induced delay of approximately 9 seconds.

In the future, users will be trained on the system and we expect an important improvement in accuracy and reduction in delay. We plan to integrate an automatic grammar checker for improving or highlighting the alternate word list. Inspired from translation domain, we want to implement dynamic alternate lists, changing according to previously applied corrections.

7. Acknowledgements

This project was funded in part by the Canadian Heritage New Media Research Networks Fund, which promotes innovation in new media or interactive digital content that pertain to the Canadian cultural sector.

8. References

[1] G. Boulianne, J.-F. Beaumont, M. Boisvert, J. Brousseau, P. Cardinal, C. Chapdelaine, M.Comeau, P. Ouellet, and F. Osterrath., "Computer-assisted closed-captioning of live TV broadcasts in french," in *Proceedings of the 2006 Interspeech - ICSLP*, Pittsburg, US, September 17-21 2006.

[2] T. Imai, A. Matsui, S. Homma, T. Kobayakawa, O. Kazuo, S. Sato, and A. Ando, "Speech recognition with a respeak method for subtiling live broadcast," in *Proceedings of the 2002 ICSLP*, Orlando, US, September 16-20 2002.

[3] M. Wald, "Creating accessible educational multimedia through editing automatic speech recognition captioning in real time." *Interactive Technology and Smart Education*, vol. 3, no. 2, 2006.

[4] A. Bateman, J. Hewitt, A. Ariyaeeinia, P. Sivakumaran, and A. Lambourne., "The quest for the last 5%: Interfaces for correcting real-time speech-generated subtitles," in *Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000)*, The Hague, Netherlands, April 1-6 2000.

[5] P. Cardinal, G. Boulianne, and M. Comeau, "Real-time correction of closed-captions," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, Prag, Czech Republic, June 2007.

[6] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms, 2nd edition.* MIT Press, Cambridge, MA, 2001.