Identifying reading strategies employed by learners within their oral french self-explanations

Marilena Panaite, Mihai Dascalu, Philippe Dessus, Maryse Bianco, Stefan Trausan-Matu

Identifying Reading Strategies Employed by Learners within their Oral French Self-Explanations

Marilena PANAITE, Mihai DASCALU
Computer Science Department, University Politehnica of Bucharest, Splaiul Independentei, No 313, Bucharest, Romania
marilena.panaite@gmail.com, mihai.dascalu@cs.pub.ro

Philippe DESSUS, Maryse BIANCO
LaRAC, Univ. Grenoble Alpes, Grenoble, France
philippe.dessus@univ-grenoble-alpes.fr, maryse.bianco@univ-grenoble-alpes.fr

Stefan TRAUSAN-MATU
Computer Science Department, University Politehnica of Bucharest, Splaiul Independentei, No 313, Bucharest, Romania
stefan.trausan@cs.pub.ro

Abstract*: Natural Language Processing has massively evolved during the last years and many up-to-date applications integrate different speech tools in order to create an enhanced user experiences. For obtaining a seamless integration of existing speech recognition systems, there is a trending interest for developing and improving existing speech-to-text algorithms. The aim of this paper is to improve user interaction with the ReaderBench platform, by developing and integrating a speech recognition module designed so that young pupils can dictate their self-explanations to a given text. Afterwards, the ReaderBench framework is used to automatically evaluate the employed reading strategies based on the resulted speech transcriptions. A dataset containing 160 self-explanations from students ranging from 9 to 11 years old was analysed using both original transcripts, and the ones automatically generated by our custom speech recognition system. Multiple methods designed to perform speech recognition are also compared, while a new dedicated model was trained in order to improve the quality of the existing French model for speech recognition from CMUSphinx speech recognition system. Our revised model includes a pronunciation dictionary obtained after training a Long Short-Term Memory (LSTM) Grapheme-to-Phoneme neural network. The accuracy of our system is benchmarked in relation to the automated process of identifying reading strategies implemented in our ReaderBench framework, which is applied on both manual transcriptions and automated speech-to-text inputs. The obtained results argue for the adequacy of our method as the slight decrease in terms of identification accuracy is justifiable in contrast to the effort of manually transcribing each self-explanation.*

Keywords: *Speech recognition; Automated identification of reading strategies; Natural Language Processing; ReaderBench framework.*

I. INTRODUCTION

Natural Language Processing (NLP) is a growing domain that can be integrated into multiple day-to-day activities in which speech tools have recently gained an extended applicability. The main idea in using speech recognition systems is to improve the user experience and the application's usability, in tight correlation with transition from text data to orally expressed inputs. However, we need to be careful when choosing the workflow for this type of applications so that the language model is adapted to the users' needs. After integrating the speech tools, systems must be accepted by end users who can provide feedback based on the system's performance.

While narrowing the scope of the previous systems in order to ensure educational outcomes, of particular interest is the usage of speech recognition modules to capture learner's oral self-explanations. Self-explanations are strongly encouraged among students in order to improve their comprehension because they better understand a text when generating out loud thoughts and impressions regarding the input text. Moreover, the employed reading strategies are indicative of their comprehension levels as in-depth reading strategies presume inferences, as well as the integration between the presented information and learner's prior knowledge. However, this does not apply for students who do not have prior knowledge on the information expressed in the text, or learners who are poor readers and cannot understand the notions presented in the text [1]. In contrast, people who are expert or strategic readers are able to determine whether they can make inferences based on the information presented to them and are capable of self-evaluating their level of comprehension over the read information. Special techniques presuming students' aloud lecture and their self-explanation of the read text are recommended for average readers who are not capable of tracking their understanding.

Current implementation models for speech recognition and existing learning applications are analyzed in section 2. Section 3 introduces the steps used for integrating a speech recognition system in the Self-Explanation module from *ReaderBench,* including the training of a G2P (Grapheme-to-Phoneme) neural network for the basic Sphinx French dictionary. Section 4 provides a comparative perspective over the results of the Self-Explanation module employed after integrating different speech recognition systems. The paper ends with discussions and conclusions over the obtained results and presents future experiments that can be made.

II. STATE OF THE ART

2.1 Speech Recognition

During the last years, the Natural Language Processing domain was influenced by the growing interest of big companies that want to integrate speech recognition solutions in their software products in order to obtain better user experience. In this way, the majority of the speech recognition research papers focus on addressing specific problems, like integrating speech tools in accessibility software products or adapting to different speakers depending on their context.

As most written languages are based on an alphabet, written symbols are associated with sounds; thus, a standardized phonetic alphabet makes the connections between a letter and its phonetic transcription. A known standardization for sounds that are part of an alphabet is the one for English, where there is a reference standard called ARPAbet, which is standard subset for IPA [2]. In most of the speech recognition algorithms, the key point in finding the best match is to have an efficient implementation of the search algorithms for finding the most probable paths. Among these methods, one of the most used is the Viterbi algorithm, which provides a fairly good performance in recognizing the most probable sequence of words that matches the spectrogram. In this case, their performance is quite good in terms of recognition and the running time is also not to be neglected as the time complexity is close to $O(N^2T)$, where T is the length of the list of observation and N the number of states in the Markov graph [2].

Before using Long Short Term Memory neural networks (LSTM) for doing the Grapheme-to-Phoneme transformation, most of the speech recognition systems were using Weighted Finite State Transducers automates (WFST). This method needed additional steps for aligning the grapheme used as input according to the context in which the translation was made. The most popular tool that implements this transformation is Phonetisaurus [3]. This method used in generating pronunciations for new words uses a combination of alignment with various n-gram models and a WFST system in order to determine the best phoneme match for a grapheme as described in here [4]. Since April 2016, the CMUSphinx team has released a brand-new tool for training a grapheme-to-phoneme translation using deep learning, specifically the Sequence-to-sequence Long Short Term Memory Recurrent Neural Network [5].

2.2 Automated Identification of Reading Strategies

Usually, a single tutor is present in a classroom who has to take care of multiple students and also has to make sure that everyone gets evaluated and is offered appropriate feedback for their results. In this case, the participants will benefit more from having an automatic method that detects their reading strategies and gives them feedback in real time. Depending on the measured reading strategies, different heuristics can be used. For the most basic strategies, like paraphrasing and text-based inferences that are both based on the words used in the initial story, similarity measures for words and paragraphs are used. One measure would be to identify the lexical similarity using lexicalized ontologies (e.g., WordNet). Another measure would consider the count of words referring to causality (e.g., “because”) in order to describe the way in which students use inference-based words to explain the story [6].

For achieving good results, specific heuristics for detecting semantic cohesion are implemented, as the lexical distance was insufficient. One such NLP technique is Latent Semantic Analysis (LSA) [7], where the focus is to figure what is the similarity between a big corpus of text and some terms using the idea that words which occur similarly in the text are closer to each other (for example “chair” is close to “table” according to LSA). Another measurement that is used for finding semantic similarities is the Latent Dirichlet Allocation (LDA) [8], which implies figuring out a distribution over the topics from a document, supposing there is a great density of different subject.

In addition to these methods, there are other heuristics that can also provide good results for different types of corpora. One example is the Entailment Index [9], where the idea is to extract the initial text that is read aloud to the students and the self-explanation that was given by each one of them. Then we take this parts of texts and run them through a preprocessing step, where we perform some basic operation like parsing and tokenization and then in the final steps we create a dependency graph between the terms for each of the text and explanation part. Then, we perform various operations for matching the two graphs with the partial graphs obtained, in order to find out the coefficient for the semantic similarity between the two texts.

III. METHOD

Our goal was to integrate a speech recognition tool within the *ReaderBench* Self-Explanation module. This would increase the system's usability and make it more efficient in evaluating learner's reading strategies. In the following sections, we are going to describe some methods for integrating a speech recognition module along with the Self-Explanation API implemented in *ReaderBench*. The best way to plug-in the new reading strategy model is right after the speech recognition module has finished recognizing some sentence that was given as an audio input. After the full-length text is obtained, the results can be given to the reading strategy module that performs a comparison between the given text and the original one that the students have listened to.

3.1 Selected Corpora and Phonetic Transcription

The dataset for the *ReaderBench* Self-Explanation module was gathered using an experiment in which a group of students read out loud a story individually, split in 6 parts, and was asked to tell what they have understood from each part, just after its reading. Each of these explanations were then transcribed and used for further analysis. During the experiment [9], performed on approximately 80 students in third and fourth grades read and explained two stories of approximately 300 words. The stories used in the experiment are French stories for kids, called *Matilda* and *L'avaleur de nuages*. Also, the story was specially chosen to fit the comprehension level of the children involved in the experiment.

The *ReaderBench* framework implements a series of APIs for using the functionality implemented and among that, the Self-Explanation module also has its own API that we will use later in our system. Usually, this API is used for finding out the reading strategies used by the children in order to find out the level of comprehension over the story that was presented to them. Similar to other reading strategies systems, *ReaderBench* offers multiple features for evaluating the reading strategies of students like the level of paraphrasing, text-based inferences and knowledge-based inferences [11].

The actual self-explanation corpus is obtained from previous records of lessons with the 9-11-year-old pupils that where performing the task of explaining the text that was just read to them. The format of the input data for the speech recognition model are audio files that contain the recording for all the interaction between the tutor and the participants, including pauses and reformulations. Also, there are the transcripts for each of these conversations that were pre-processed in order to analyze them and build the self-explanation module from *ReaderBench*.

3.2 Sphinx baseline evaluation

CMUSPhinx [11], one of the most popular open-source applications for speech-to-text, is developed by the Carnegie Mellon University and provides an API with support for the Java programming language to develop applications for recognition of speech adapted to a particular language. Thus, *CMUSphinx* is an open-source framework that can be used for developing personalized speech-to-text applications with the possibility of building a language model and an acoustic model adapted to speech recognition system that we want to develop. Furthermore, we can even introduce your own dictionary that provides a link between phoneme and grapheme.

After cleaning up the text and dividing it into sentences, the input for the statistical language model is prepared. Moreover, an important toolkit that helps us in building the language model using the N-gram model is CMU-CAMBRIDGE Toolkit, which results in an ARPA format language model. This special file format is used as an output representation for the results of the language model training and its main format exposes the probabilities for each set of n-grams calculated during the modeling process, like the 1-grams, bi-grams, tri-grams, etc. More information about the implementation of Hidden Markov Model Toolkit can be found in the HTK Book [12].

Regarding the Self-Explanation module from *ReaderBench*, the current implementation involves giving a text corpus as an input and obtaining metrics regarding the level of comprehension over the story that is presented to the students. The data that was used during the training of this module is made of several transcripts of the children's explanations. This involves recording the students and then making a transcript of their opinion in order to feed it to the *ReaderBench* Self-Explanation API. Then, speech synthesis can be used for reading the story to the participants and speech recognition tools for capturing the pupils' explanations, resulting in real time evaluation of the level of comprehension.

3.3 Enhanced G2P Speech Model

When integrating speech recognition tools in applications that are used in a specific domain (e.g., health care, education, etc.), an important aspect is to make sure that domain-specific words can be recognized by the speech module. In this case, we have two options for ensuring quality in terms of speech recognition: one is to build from scratch a new dictionary with all the pronunciations, while the second uses an existing dictionary as a baseline and tries to improve it by adding specific terminology. The first option is by far more difficult to implement as all pronunciations need to be created. In order to obtain a new model for a language by extending an existing model, it is necessary to find an open source set of tools that is flexible in training new models and gives support for new extensions of already existent dictionaries. Thus, it is important to look after a complete set of language tools that will grant the possibility of extending different components of the speech recognition tool, including the existing language models, the pronunciation dictionaries or even retraining the existing acoustic model for better recognition. The desirable outcome is that the provided acoustic model does not depend on the environment or on different types of voices and perturbations.

As described in the previous section, the experiments made after integrating the speech recognition module did not provide adequate results in identifying the children's reading strategies as all the measured features were affected by the quality of the speech recognition output. Therefore, the next step was to improve the recognition for French language. Thus, we opted to improve the language model by training a LSTM and using it to decode other new words that were not present in the initial French dictionary – i.e., we performed a grapheme-to-phoneme (G2P) training for a specific language pronunciation dictionary that maps the learner words with their corresponding pronunciation. Afterwards, the trained dictionary is given as an input to the speech recognition *CMUSphinx* workflow.

The *CMUSphinx* implementation for training the G2P transformation using a LSTM is based on a research proposal made by Microsoft in [13] in which they also proposed the two solutions starting from previous research on these neural networks [14]. Also, they made an experiment with a simple encoder-decoder architecture that did not performed well compared with the combined version presented in [14]. In consequence, the *CMUSphinx* implementation was only inspired by these articles, as the same architecture that was implemented with *TensorFlow* had better results on the CMU English dictionary, meaning a rate of 24% [15] with 512 hidden units and 2 hidden layers.

Based on the training made by the *CMUSphinx* team with their English dictionary, we decided to create a similar model for the French language, which can be used further in the recognition process that is integrated with the *ReaderBench* Self-Explanation module. In order to perform the training, we took the current French dictionary offered by the *CMUSphinx* official website and feed it to the neural network for training. As described in the English dictionary experiment, we trained two different models, one with 64 hidden units and one with 512 hidden units and then evaluated them using all the 105,000 words from the CMU French dictionary. Also, in order to train the neural network, we required a machine with a graphical processor that was capable of performing the GPU training computation because the *CMUSphinx* implementation was using *TensorFlow*. For this experiment, we used an Amazon custom AMI with both *TensorFlow* and *Cuda* libraries preinstalled that enabled us to train the network in almost 12 hours, i.e. about 30,000 steps with the default learning rate of 0.5 that was decreasing with 0.8 after each decay. As it can be observed from Table 1, the network for the French language obtained comparable results with the one from the English language in terms of WER (Word Error Rate).

LSTM Model	French *CMUSphinx* (105,003 words)	English *CMUSphinx*
LSTM with 2 hidden layers and 64 hidden units	57.89% (WER)	32.03% (WER)
LSTM with 2 hidden layers and 512 hidden units	23.19 % (WER)	24.23% (WER)

Table 1. Results after training the G2P LSTM.

IV. RESULTS

Table 2 depicts several differences between the results obtained from the baseline (automatically identified strategies using transcripts) and those obtained after the recognition performed using the Sphinx recognition module. This model does not include our G2P optimizations and it reflects the drastic decrease in the accuracy of automatically identifying reading strategies using our speech-to-text functionality versus correct transcriptions. We opted to present these correlations in contrast to using the reading strategies annotated by human experts in order to highlight the noise induced by the speech-to-text component. The differences come from the recognition part, as the language used is French and the dictionary from *CMUSphinx* is a small set that is not tuned and trained to recognize a wide spectrum of terms. Whereas the correlations differ along strategies, this difference cannot be causally attributed to the kind of strategy (e.g., text inferences are not especially more difficult to recognize from speech excerpts). Moreover, the differences between texts may be caused by the recording context of the verbalizations, since each text has been subject to a separate experimental session.

Reading Strategy automatically identified by *ReaderBench*	*L'avaleur de nuages* (*N* = 71)			*Matilda* (*N* = 64)		
	α	ICC	*r*	α	ICC	*r*
Paraphrase	.842	.373	.730**	.837	.466	.767**
Causality	.781	.462	.714**	.354	.164	.388**
Text inference	.022	.005	.011	.456	.188	.552**
Bridging	.826	.550	.719**	.288	.088	.465**
Inferred knowledge	.797	.468	.676**	.340	.128	.275*
Metacognition	.463	.257	.357**	.569	.363	.569**

Table 2. Correlations between baseline results for Self-Explanations and results after using the CMUSphinx recognition module, detailed for each Reading Strategy score.

The next experiment included the *improved version* of the speech recognition system that contains the trained G2P phoneme transcription described in Section 3.2. In order to apply the new model to the existing speech corpus, we added new transcriptions that were not present in the French pronunciation dictionary. The updated experiments for the Self-Explanation modules are presented in

Table 3. We can observe that the noise induced by the automated speech-to-text conversion has dramatically decreased and that the results (with very few exceptions) are very similar to the ones obtained using the expert transcriptions. This improvement makes the recognition process from speech very closely related to this of human's, especially for one text, *L'avaleur de nuages*. The correlations remain lower for *Matilda*, which can be attributed to the recording context.

Reading Strategy automatically identified by *ReaderBench*	*L'avaleur de nuages* (*N* = 71)			*Matilda* (*N* = 64)		
	α	ICC	*r*	α	ICC	*r*
Paraphrase	.999	.671	.998**	.968	.922	.943**
Causality	.960	.895	.923**	.872	.764	.798**
Text Inference	.945	.894	.898**	.257	.123	.381**
Bridging	.999	.998	.998**	.581	.365	.493**
Inferred Knowledge	.991	.981	.983**	.733	.561	.602**
Metacogntion	.836	.716	.722**	.851	.716	.801**

Table 3. Correlations between baseline results for Self-Explanation and results after G2P enhanced model

V. CONCLUSIONS

Integrating a speech recognition module into *ReaderBench* improves user experience as students can find out in near real time the reading strategies employed within their oral self-explanations. Although adding a speech-to-text component to the evaluation workflow affects the performance of the Self-Explanation API, the proposed Grapheme-to-Phoneme transformation dramatically improved the quality of speech recognition. The results were sensitive to several points: the recording settings, the integrated acoustic and language models because the size of the French dictionary from *CMUSphinx* was smaller compared to the English equivalent. In addition, there was no previous training for the French dictionary regarding the G2P transformation relying on a Long Short Term Memory neural network (LSTM).

For further research, a seamless integration of the LSTM Grapheme-to-Phoneme training module would be beneficial to automatically introduce new pronunciations for terms that are not present in the dictionary without a manual update. Another approach would be to train different architectures and configurations of LSTM networks, like implementing a bidirectional LSTM or even combine it with an N-gram model, in order to diminish the WER (Word Error Rate) and obtain an optimal Grapheme-to-Phoneme transformation for French Language tailored for the learning task.

Acknowledgements

This work was funded by the 644187 EC H2020 Realising an Applied Gaming Eco-system (RAGE) project, and by the DEVCOMP project of the French *Agence Nationale de la Recherche* (DEVCOMP-ANR-10-BLAN-1907-01).

Reference Text and Citations

[1] McNamara, D. S., & Scott, J. L. (1999). Training Self Explanation and Reading Strategies. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 43(21), 1156-1160.

[2] Jurafsky, D., & Martin, J. (2009). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2nd ed.). Upper Saddle River, N.J.: Pearson Prentice Hall.

[3] Novak, J. R., Yang, D., Minematsu, N., & Hirose, K. (2011). Phonetisaurus: A wfst-driven phoneticizer. The University of Tokyo, Tokyo Institute of Technology, 221-222.

[4] Novak, J. R., Minematsu, N., & Hirose, K. (2012, July). WFST-based grapheme-to-phoneme conversion: open source tools for alignment, model-building and decoding. In 10th International Workshop on Finite State Methods and Natural Language Processing (p. 45).

[5] Sundermeyer, M., Ney, H., & Schluter, R. (2015). From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing IEEE/ACM Trans. Audio Speech Lang. Process., 23(3), 517-529. doi:10.1109/taslp.2015.2400218

[6] Dascalu, M., Dessus, P., Bianco, M., & Trausan-Matu, S. (2014). Are Automatically Identified Reading Strategies Reliable Predictors of Comprehension? Intelligent Tutoring Systems Lecture Notes in Computer Science, 456-465.

[7] Dumais, S. T. (2004). Latent Semantic Analysis. Annual review of information science and technology, 38(1), 188-230.

[8] Zhang, Z., Miao, D., & Gao, C. (2013). Short text classification using latent Dirichlet allocation. Journal of Computer Applications, 33(6), 1587-1590.

[9] Dascalu, M. (2014). ReaderBench (1)–Cohesion-Based Discourse Analysis and Dialogism. Analyzing Discourse and Text Complexity for Learning and Collaborating Studies in Computational Intelligence, 137-160.

[10] Dascalu, M., Dessus, P., Trausan-Matu, S., Bianco, M., & Nardy, A. (n.d.). ReaderBench, an "Environment for Analyzing Text Complexity and Reading Strategies. Springer Berlin Heidelberg

[11] Lamere, P., Kwok, P., Walker, W., Gouvêa, E. B., Singh, R., Raj, B., & Wolf, P. (2003, September). Design of the CMU sphinx-4 decoder. InINTERSPEECH.

[12] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D. & Woodland, P. (2002). The HTK book. Cambridge University Engineering Department. Cambridge, UK.

[13] Yao, K., & Zweig, G. (2015). Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. arXiv preprint arXiv:1506.00196.

[14] Rao, K., Peng, F., Sak, H., & Beaufays, F. (2015). Grapheme-to-phoneme conversion using Long Short-Term Memory recurrent neural networks. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). doi:10.1109/icassp.2015.7178767

[15] Zen, H. (2015). Acoustic modeling in statistical parametric speech synthesis–from HMM to LSTM-RNN. *Proc. MLSLP*.