JOINT DECODING OF COMPLEMENTARY UTTERANCES

Mickael Rouvier, Benoit Favre, Frederic Bechet

Aix-Marseille University, CNRS, LIF UMR 7279, 13000, Marseille, France

ABSTRACT

Errors in open-domain ASR can be corrected by asking the speaker to rephrase targeted segments in utterances where they have been detected. The utterance merging problem consists in generating a better transcript from the utterance where errors have been detected and a clarification utterance. We introduce an alignment-decoding algorithm for jointly processing the two utterances and benefit from the complementary information they contain. The algorithm aligns word lattices in the WFST framework with a probabilistic cost model. Results on the BOLT-BC speech-to-speech translation task show an improvement of 2.84 points of accuracy compared to aligning the one best without joint decoding.

Index Terms— Joint probabilistic model, lattice alignment, ASR error correction, dialog systems

1. INTRODUCTION

Automatic speech recognition is a building block of more and more speech-enabled applications. However, recognition errors due to out of vocabulary terms, disfluencies, mispronounciations, or difficult acoustic conditions can cripple the generated transcript and deteriorate user interactions. Mimicking humans, dialog systems have been equipped with clarification modules that leverage domain knowledge in order to conduct interactions and recover the message that the user intended to communicate [1]. Open-domain applications, such as speech-to-speech translation or voice search, can also benefit from clarification interactions by deploying a dialog system that detects ASR errors, asks targeted clarification questions in order to elicit a rephrase of the corrupted part of the utterance transcript, and recover a better transcript from both the original and the clarification utterances [2].

The problem of merging complementary utterances is difficult for two main reasons. First, users of a clarifying dialog need to use context in order to rephrase parts of a sentence, and will often rephrase words outside of the error segments as well. Second, short utterances, such as clarifying utterances, are harder to recognize because the ASR system cannot take advantage of context through its language model. Previous work on the problem [3, 4] has mainly covered the first aspect, by proposing alignment strategies between the utterances accounting for task-specific constraints: the utterances might have different lengths, contextual words might be rephrased in addition to targeted words, ASR errors might occur in the clarification utterance transcript or in the original utterance transcript outside of the error segment boundaries.

Previous work was limited in that it assumed the ASR system had enough information to decode each utterance independently of each other. Here, we reconsider that hypothesis and tackle the problem of jointly decoding multiple utterances. Joint decoding is attractive because, on the one hand, decoding the original utterance depends on what was said in the clarification utterance, and on the other hand, the clarification speech lacks the context of the words it replaces, which can void the benefits of using a language model. This approach is not limited to clarification dialogs but could be applied to other tasks, such as to transcribe disfluent speech, where a sentence spans multiple turns, and overlapped speech [5], or to benefit from the effects of speaker entrainment [6].

In this paper, we propose a model for joint decoding of two comparable utterances which accounts for discrepancies between an erroneous utterance and its clarification and test it in the same framework as [4]. Our contributions are as follows:

- We propose a probabilistic model of joint decoding and alignment of speech utterances.
- This model is cast in the WFST framework in order to jointly decode word lattices and confusion networks.
- We experiment on a realistic corpus of clarification subdialogs collected for the speech-to-speech translation task of the DARPA-BOLT project.

The paper is organized as follows. Section 2 details the proposed model, Section 3 shows how to cast this model for the task of transcript clarification, Section 4 details and analyses the experiment results on the BOLT corpora, Section 5 compares our approach to related work and Section 6 discusses the conclusions.

This work was partially funded by DARPA HR0011-12-C-0016 as an AMU subcontract to SRI International.

2. JOINT ALIGNMENT AND DECODING

Traditionally, ASR is modeled as the probability of a word sequence given the observed acoustic recording. Here, we wish to jointly model two utterances which come from two different acoustic sequences. The two utterances shall not lead to the same transcript, but rather are complementary to each other in that one may bring evidence on the words of the other and conversely. This requires the notion of alignment between the two utterances. If they match completely, their decoding should be identical to that of decoding a single utterance, while if they are totally different, their decoding should be independent.

2.1. Model

A probabilistic model for sequence alignment was proposed in [7]. It assumes that edit operations are independent and can be estimated with the EM algorithm. This model was extended to Conditional Random Fields by [8] to allow feature-based training of a discriminative aligner. The scarcity of training data available for merging clarification utterances leads us to the following generative model.

First, we model the alignment of sequences of words. Let X be the original word sequence. Let Y be the clarification word sequence. Let a be a sequence of edits from the set of all sequences of edits A. Following [7], the joint sequence XY can be expressed in term of all the alignments between X and Y.

$$P(XY) = \sum_{a \in A} P(a|XY) \tag{1}$$

with $P(a|XY)$ the probability of a sequence of edits that maps X to Y, $a = a_1 \dots a_n$. Assuming independent edit operations leads to the following formulation:

$$P(a|XY) = \prod_{i=1}^{n} P(a_i|XY) \tag{2}$$

where $P(a_i|XY)$ is the probability of one of the edit operations:

$$P(a_i|XY) = \begin{cases} P(del|x_j) & \text{deletion} \\ P(ins|y_k) & \text{insertion} \\ P(sub|x_j, y_k) & \text{substitution} \end{cases} \tag{3}$$

with (j, k) the couple of words which are being aligned (omitting the missing word in case of insertion or deletion).

We propose to introduce word posterior probabilities associated with each word in X and Y. For that we assume that the edit operations are independent of the word generation process.

$$P(del|x_j) = P(x_j)^{\lambda_x} P(del|x_j)^{\theta_{\text{del}}} \tag{4}$$

$$P(ins|y_k) = P(ins|y_k)^{\theta_{\text{ins}}} P(y_k)^{\lambda_y} \tag{5}$$

$$P(sub|x_j y_k) = P(x_j)^{\lambda_x} P(sub|x_j y_k)^{\theta_{\text{sub}}} P(y_k)^{\lambda_y} \tag{6}$$

Here, λ_x and λ_y are hyper-parameters that control the contribution of the original and clarification sequences of words to the model. θ_{del}, θ_{ins} and θ_{sub} control the contribution of edit operations to the alignment.

In this model, the best decoding-alignment of the two sequences can be retrieved through Viterbi decoding:

$$a^\star = \underset{a \in A}{\text{argmax}}\, P(a|XY) \tag{7}$$

2.2. Lattices

Without loss of generality, we consider that instead of building a new kind of decoder that can process multiple utterances at once from the ground up, we can take advantage of the search space of a decoder running independently for each utterance in the form of word lattices.

Given that a word lattice represents a collection of word sequences, the joint alignment and decoding between two lattices corresponds to finding the minimum cost edit script between the cartesian product between the sequences represented by both lattices. If $L_x = \cup_l \{X_l\}$ is the set of word sequence hypotheses for X (the word lattice), resp. $L_y = \cup_m \{Y_m\}$ for Y, the best decoding-alignment $a^\star_{XY}$ of the two lattices L_x and L_y is:

$$a^\star_{XY} = \underset{\substack{XY \in L_x \times L_y \\ a \in A}}{\text{argmax}}\, P(a|XY) \tag{8}$$

In order to find $a^\star_{XY}$ efficiently, this model can be cast in the transducer alignment paradigm as proposed by [9]. Assuming all weights are in the tropical semiring, let T_X, T_Y be acceptors that represent word lattices weighted by $-\lambda_x \log P(x_j)$, resp. $-\lambda_y \log P(y_k)$. Let T_e be a single state edit transducer that maps each word x_j to each word y_k for a cost of $-\theta_{sub} \log P(sub|x_j y_k)$, maps each word x_j to ε (the empty symbol) with a cost of $-\theta_{del} \log P(del|x_j)$, and maps ε to y_k with a weight of $-\theta_{ins} \log P(ins|y_j)$. $a^\star_L$ can be derived from the composition:

$$a^\star_{XY} = bestpath(T_X \circ T_e \circ T_Y) \tag{9}$$

An interesting potential extension is that this approach can be used to jointly decode more than two utterances. Since T_e is symmetric, the operation is commutative, i.e. $T_X \circ T_e \circ T_Y = T_Y \circ T_e \circ T_X$. Therefore, by alternatively composing word lattices and the edit transducer, one creates a search space weighted with the sum of the costs for the individual alignments.

$$a^\star_{XYZ\dots} = (((T_X \circ T_e) \circ T_Y) \circ T_e) \circ T_Z \dots \tag{10}$$

Even though this search space is large, recent advances in WFST implementations allow the lazy evaluation of the compositions when searching for the best hypothesis which is both efficient in term of time and memory.

This paper does not cover the joint decoding of more than two utterances. Therefore, we will make use of Equation 9 in the context of the transcript clarification task, as explained below.

3. TRANSCRIPT CLARIFICATION

3.1. Task description

In this section we describe our approach to transcript clarification and explain how it can be solved within the joint decoding-alignment model.

The BOLT-BC task consists in machine mediated speech-to-speech translation. The machine takes the role of an interpreter that translates a conversation between speakers of different languages. Like an interpreter, the machine can take the initiative to clarify some words it did not understand or did not know, in order to avoid the typical diverging conversations of non-interactive translation systems subject to ASR errors. Clarification sub-dialogs, initiated when a system detects an error in ASR output, consist in a question targeting the error, followed by a clarifying answer by the user. From the original utterance and the clarification utterance, the system shall generate a better transcript of the user intent, which generally is not the transcript of one of the utterances, but derived from both utterances. Even though the system might detect multiple error segments, only one is clarified at a time (if too many errors tarnish the utterance, the system asks to rephrase or repeat it completely). After being repaired, the transcript is sent to the translation system further along the pipeline. Here is an example of a subdialog:

1. Speech: They traveled across the desert on camel back
2. ASR: They [travel to cross to] desert on camel back
3. Error: [travel to cross to]
4. Question: Can you rephrase AUDIO(travel to cross to)?
5. Clarification: They visited the desert
6. Intent: They visited the desert on camel back

In this dialog, we call *original* the ASR transcript of the original utterance (2). The *error segment* is the sequence of words targeted by the clarification dialog (3). The *clarification* is the answer from the user to the clarification question (5). The *intent* is the sequence of words that the system should generate as a result of the merging process (6). In the next sections, we build a system which inputs an original utterance with an error segment and a clarification utterance. We expect this system to output the intended utterance transcript.

When asked to rephrase part of their utterances, the users might completely rephrase them, rephrase only the targeted segment, or contextualize the edit operation with other words, which might be subject to ASR errors, or be rephrased versions of words outside the error segment. The proposed system accounts for all these behaviours.

3.2. Intended utterance generation

In order to solve the task of merging the original utterance and the clarification utterance, we take advantage of the model described in Section 2 for joint decoding-alignment of utterances. Here, we jointly decode the original and clarification utterances. Equation 9 gives a decoder for the model in which X and Y are respectively the original and the clarification, T_X and T_Y are the word lattices generated by the ASR system for those utterances, weighted by the posterior probability of each word, and T_e is the edit transducer, weighted by the probability of the edit operations.

T_e essentially performs a Levenshtein alignment between the word lattices, with edit operations balancing (scaled) word probabilities. [4] has proved that Levenshtein alignment must be modified to account for task-specific issues such as the fact the the clarification is generally shorter than the original, and the fact that words in the error segment are incorrect. Without loss of generality of the model, T_e can be modified to account for this specificity. This yields the following systems (extended from [4]):

- Replace baseline: words from the original are replaced by the clarification words as if the user had completely rephrased his utterance.
- Insert baseline: words from the clarification are inserted in place of the error segment.
- Levenshtein: T_e is used as described in Section 2.
- Error-loop + affine gap: words in the error segment are replaced by a loop of dummy symbols (with posterior probability of 1) that can match as many words from the clarification as needed. Affine-gap adds a cost to start a sequence of insertions / deletions (with different probabilities for the start and the continuation).
- Phonetic match: the edit probability for substituting two words depends on the distance of the phonetic transcription of the words.
- Word embedding: the edit probability for substitution depends on the distance between the words in an embedding space [10].

Since Equation 9 yields an alignment between X and Y, to generate the transcript from the alignment, we give priority to the clarification words over the original words. Compared to [4], our approach can be viewed as a generalization of utterance alignment to lattices. The main difference is that our model accounts for the balance between word probabilities

and alignment probabilities. In the hereunder experiments descriptions, we call *mergers* the systems for intended utterance generation.

4. EXPERIMENTS AND RESULTS

In order to assess the quality of our system, we used two clarification dialog corpora. The first corpus, which we call *BOLT-DEV*[1], is a set of clarification situations where the user is given an original utterance with an error segment and has to rephrase it. This corpus was collected for the development of the BOLT speech-to-speech translation system. The second corpus, which we call *BOLT-P2*, contains real clarification dialogs recorded during the BOLT phase 2 evaluation. The first and second corpora contain respectively 70 and 141 dialogs. In both corpora the original utterance contains at least one ASR error segment (the targeted error) and might contain additional errors. For all the experiments, we tune the hyper-parameters of the system on one corpus and test it on the other (and vice versa).

ASR transcripts were obtained by running a DNN-based ASR systems developed by SRI in the course of the BOLT project [11]. On Bolt-P2, its WER (Word Error Rate) is 15.76% on the original and 17.78% on the clarification utterances. Note that in all experiments, we consider that targeted error segments have been correctly located by the error detection module.

Joint decoding for utterance merging performance is evaluated with two metrics: merging accuracy represents the rate of complete recovery compared to the human-written reference, and merging Word Error Rate (WER) is the word error rate of the hypothesis compared to the intended transcript that should have been produced (it's not the WER relative to the original reference transcript).

In the following, we propose to use and compare different kind of ASR search space representation (one-best hypothesis, word lattice and confusion network) as input of the utterance merger described in Section 3.

4.1. One-best hypothesis vs word lattices

In Table 1 we compare the baseline mergers (*Insert*, *Replace* and *Levenshtein*), with the task-specific merger (*Err loop + affine gap*). The latter is used in a first version with one-best hypothesis given by the ASR (called *(1-Best) Err loop + affine gap*) and a second version with word lattice search space (called *(Lattice) Err loop + affine gap*). We observe that the two different versions of the task-specific merger perform better than the baselines, resulting in a reduction of WER and improvement of accuracy. We observe that taking advantage of lattices on the *BOLT-P2* corpus yield an improvement of 2 points of WER compared to relying on the 1-best. In general the lattice-oriented system provides slightly better results compared to its 1-best counterpart.

[1]This corpus was called *Speech* in [4].

	BOLT-DEV		BOLT-P2	
Method	Acc.	WER	Acc.	WER
(Baseline) Replace	14.29	54.47	09.93	75.42
(Baseline) Insert	08.57	46.24	17.02	27.55
(Baseline) Levenshtein	20.00	23.55	18.44	20.36
(1-Best) Err loop + affine gap	24.29	20.99	22.70	18.70
(1-Best) Oracle	28.57	16.17	23.21	19.27
(Lattice) Err loop + affine gap	24.29	20.57	24.11	18.49
(Lattice) Oracle	32.64	14.05	27.26	15.98

Table 1. Accuracy and WER results on the *BOLT-DEV* and *BOLT-P2* corpus according the one-best and word lattice ASR search spaces.

4.2. Hyper-parameters

In the joint alignment and decoding model, we propose to optimize two hyper-parameters: λ_x and λ_y from Equation 6. These hyper-parameters control the contribution of the original and clarification word posteriors to the model. We simplify the problem and balance the parameters as follows: $\lambda_x = \lambda$ and $\lambda_y = (1 - \lambda)$. Figure 1 shows accuracy and WER results using different λ values on the *BOLT-DEV* and *BOLT-P2* corpora using the *Err-loop + affine-gap* merger. The best results are obtained with a λ_x and λ_y fixed respectively to 0.9 and 0.1. These values mean that the model gives more importance to the clarification words.

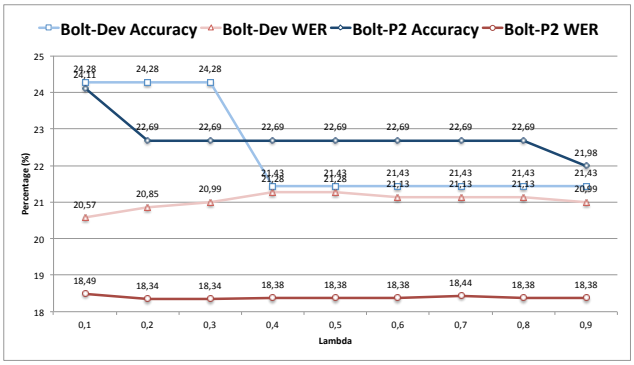

Fig. 1. Accuracy and WER results using different λ hyper-parameter values on the *BOLT-DEV* and *BOLT-P2* corpora with the *Err-loop + affine-gap* merger.

In a second experiment, we look at the variation of performance according to the beam width of the ASR decoder in order to tighten or loosen the ASR search space through hypothesis pruning. The beam width controls how large the word lattices and confusion networks get when they are generated. Figure 2 shows accuracy and WER results using different beam widths on the *BOLT-DEV* and *BOLT-P2* corpora

with the *Err-loop + affine-gap* merger. The best results are obtained with a beam width of $1,000$.

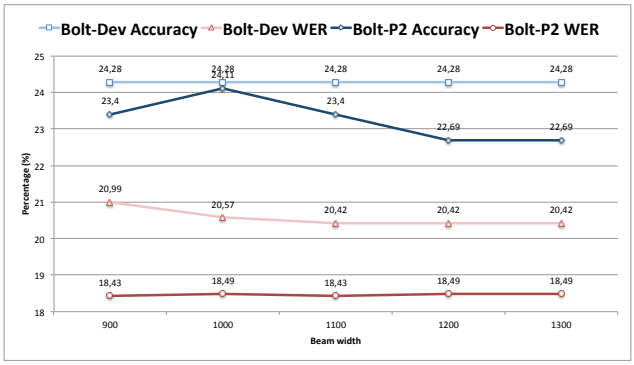

Fig. 2. Accuracy and WER results using different beam values on the *BOLT-DEV* and *BOLT-P2* corpus with the *Err-loop + affine-gap* merger.

4.3. Improved merging

In Table 2 we analyse the impact of using the word lattice with different versions of the edit transducer. Among all proposed mergers in [4], we focus on the two strategies that obtain the best results on the *BOLT-DEV* corpus: *Phonetic + words* and *Phonetic + embedding*. As expected, the mergers using the word lattice obtain the best results compared to the mergers using the one-best hypothesis. This experiments tend to prove that the gains obtained by using word lattices are not linked to a specific edit distance computation strategy.

	BOLT-DEV		BOLT-P2	
Method	Acc.	WER	Acc.	WER
(1-Best) Phonetic + words	24.29	20.57	21.99	18.70
(1-Best) Phonetic + embedding	25.71	20.85	23.40	18.18
(Lattice) Phonetic + words	25.71	20.28	23.40	18.65
(Lattice) Phonetic + embedding	25.71	19.29	24.11	18.33

Table 2. Accuracy and WER results on the *BOLT-DEV* and *BOLT-P2* corpus according the one-best hypothesis and word lattice.

4.4. Confusion networks

Confusion networks are a different representation of the ASR search space, generated from word lattices. They consist of a sequence of slots with each slot containing a list of word alternatives weighted by their posterior probability. Thus, Confusion Networks contain more paths than Word Lattices and should achieve theoretically more robust results.

In the following experiment, we propose to use confusion networks in place of the lattices as input search space for the join decoding algorithm. To achieve this and in order to keep the same joint alignment and decoding model we simply represent the confusion networks as WFTs in the same way as the word lattices.

In Table 3, we report the results obtained using confusion networks with the three alignment variations: *Err loop + affine gap*, *Phonetic + words* and *Phonetic + embedding*. First, we observe an improvement of the oracle of 3.19% points in term of accuracy compared to the oracle for word lattices. This is expected because the size of the search space is increased. Secondly, on both corpora and for each merger we observe a slight accuracy improvement compared to the use of word lattices.

	BOLT-DEV		BOLT-P2	
Method	Acc.	WER	Acc.	WER
(CN) Err loop + affine gap	22.86	19.86	25.53	18.23
(CN) Phonetic + words	24.29	19.15	24.82	18.39
(CN) Phonetic + embedding	25.71	19.57	26.24	17.92
(CN) Oracle	34.93	12.78	30.45	14.87

Table 3. Accuracy and WER results on the *BOLT-DEV* and *BOLT-P2* corpora according the confusion networks (*CN*) given by the ASR output.

5. RELATED WORK

Besides the work on utterance merging performed in the context of the BOLT project, the following work is related to our contributions.

When low quality or diverging transcripts are available, it has been proposed to align decoding hypotheses to imperfect transcripts [12]. In that particular work, the authors use subtitles and prompts in order to improve ASR hypotheses. While the decoding is generating hypotheses, they align the imperfect transcripts to the current hypothesis, and boost the LM probabilities according to how many words in a N-gram can be aligned to the imperfect transcript. This leads to improvements in term of word error rate compared to not using any external source of information. This line of work is similar to ours but it relies on textual transcript instead of jointly decoding utterances.

Short-term adaptation, such as MLLR adaptation [13] from previous transcripts, or cache language models [14], accounts for information from previous utterances for decoding the current utterance, and are able to boost the probability of a sequence of phonemes or words seen in the recent past. The large body of work in adaptation could be used in our work (in fact the ASR system used in our experiments performs such adaptation), but it does not directly apply in the context of targeted error clarification because words of the intended utterance come from different speech signals.

Jointly decoding multiple speech signals is also of interest of the robust ASR community. For instance, [15] propose an extension of the HMM/DTW framework to decode jointly

multiple occurrences of the same isolated word. They consider that the alignment shall be performed according to an additional dimension for each audio recording and propose a dynamic programming solution in order to find the optimal alignment. Even though attractive, this approach is limited to cases where multiple observations of the same utterance are available, a special case of multiview learning.

The idea of forcing multiple examples to have the same prediction has also been explored in the natural language processing community, as the task, among others, of tagging multiple instances of the same unknown word with the same tag. This can be achieved in the dual decomposition framework which maximizes the sum of constrained decoders [16]. It would be interesting to apply the same approach to the joint decoding of multiple utterances.

6. CONCLUSION

In this paper, we propose a model for jointly decoding multiple utterances that share complementary meaning. It jointly models the probability of two utterance transcripts from their respective audio recording, at the same time as their alignment probability in term of edit distance. The model can be implemented in the weighted finite state transducer framework, and inputs word lattices or confusion networks as ASR search space. Tested for the clarification utterance merging task in the framework of the DARPA BOLT project, the proposed approach leads an improvement of 2.84 points of accuracy compared to aligning the ASR one-best hypotheses without joint decoding. In addition to being a success for the utterance merging task, the model could be used for jointly decoding multiple utterances that convey a single message, such as when a speaker begins an utterance and another finishes it, or when disfluent speech spans multiple utterances but actually corresponds to a single message.

7. REFERENCES

[1] Svetlana Stoyanchev, Alex Liu, and Julia Bell Hirschberg, "Clarification questions with feedback," in *Interspeech*, 2012.

[2] Necip Fazil Ayan, Arindam Mandal, Michael Frandsen, Jing Zheng, Peter Blasco, Andreas Kathol, Frédéric Béchet, Benoit Favre, Alex Marin, Tom Kwiatkowski, et al., "Can you give me another word for hyperbaric?: Improving speech translation using targeted clarification questions," in *ICASSP*. IEEE, 2013, pp. 8391–8395.

[3] Frederic Bechet and Benoit Favre, "ASR Error Segment Localization for Spoken Recovery Strategy," in *ICASSP*, 2013.

[4] Benoit Favre, Mickael Rouvier, and Frederic Bechet, "Reranked aligners for interactive transcript correction," in *ICASSP*, Florence, Italy, May 2014.

[5] Erich Zwyssig, Friedrich Faubel, Steve Renals, and Mike Lincoln, "Recognition of overlapping speech using digital mems microphone arrays," in *ICASSP*. IEEE, 2013, pp. 7068–7072.

[6] Rivka Levitan, "Entrainment in spoken dialogue systems: Adopting, predicting and influencing user behavior.," in *HLT-NAACL*, 2013, pp. 84–90.

[7] Eric Sven Ristad and Peter N Yianilos, "Learning string-edit distance," *TPAMI*, vol. 20, no. 5, pp. 522–532, 1998.

[8] Andrew Mccallum and Kedar Bellare, "A conditional random field for discriminatively-trained finite-state string edit distance," in *UAI*, 2005.

[9] Mehryar Mohri, Fernando Pereira, and Michael Riley, "The design principles of a weighted finite-state transducer library," *Theoretical Computer Science*, vol. 231, no. 1, pp. 17–32, 2000.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *ICLR*, 2013.

[11] Horacio Franco, Jing Zheng, John Butzberger, Federico Cesari, Michael Frandsen, Jim Arnold, Venkata Ramana Rao Gadde, Andreas Stolcke, and Victor Abrash, "Dynaspeak: Sri's scalable speech recognizer for embedded and mobile systems," in *HLT*. Morgan Kaufmann Publishers Inc., 2002, pp. 25–30.

[12] Benjamin Lecouteux, Georges Linares, Pascal Nocéra, and Jean-François Bonastre, "Imperfect transcript driven speech recognition.," in *Interspeech*, 2006.

[13] Mark JF Gales and Philip C Woodland, "Mean and variance adaptation within the mllr framework," *Computer Speech & Language*, vol. 10, no. 4, pp. 249–264, 1996.

[14] Roland Kuhn and Renato De Mori, "A cache-based natural language model for speech recognition," *TPAMI*, vol. 12, no. 6, pp. 570–583, 1990.

[15] Nishanth Ulhas Nair and TV Sreenivas, "Joint evaluation of multiple speech patterns for speech recognition and training," *Computer Speech & Language*, vol. 24, no. 2, pp. 307–340, 2010.

[16] Alexander M Rush, David Sontag, Michael Collins, and Tommi Jaakkola, "On dual decomposition and linear programming relaxations for natural language processing," in *EMNLP*. Association for Computational Linguistics, 2010, pp. 1–11.