ISCA Archive
http://www.isca-speech.org/archive

International Workshop on
Spoken Language Translation 2011
San Francisco, CA, USA
December 8-9, 2011

The 2011 KIT QUAERO Speech-to-Text System for Spanish

Kevin Kilgour[1,2], *Christian Saam*[1,2], *Christian Mohr*[1],
Sebastian Stüker[1,2], *and Alex Waibel*[1]

[1]Institute of Anthropomatics
[2]Research Group 3-01 'Multilingual Speech Recognition'
Karlsruhe Institute of Technology
Karlsruhe, Germany
{kevin.kilgour|christian.saam|christian.mohr|sebastian.stueker|alexander.waibel}@kit.edu

Abstract

This paper describes our current Spanish *speech-to-text* (STT) system with which we participated in the 2011 Quaero STT evaluation that is being developed within the Quaero program. The system consists of 4 separate subsystems, as well as the standard MFCC and MVDR phoneme based subsystems we included a both a phoneme and grapheme based bottleneck subsystem. We carefully evaluate the performance of each subsystem. After including several new techniques we were able to reduce the WER by over 30% from 20.79% to 14.53%.

1. Introduction

In this paper we describe our Spanish *speech-to-text* (STT) system with which we participated in the 2011 Quaero STT evaluation. Our STT makes extensive use of system combination and cross-adaptation, by utilizing acoustic models which are trained with different acoustic front-ends and are, in addition to the normally used phonemes, also based on graphemes.

1.1. Quaero

Quaero (http://www.quaero.org) is a French research and development program with German participation. It targets to develop multimedia and multilingual indexing and management tools for professional and general public applications such as the automatic analysis, classification, extraction, and exploitation of information. The projects within Quaero address five main application areas:

- Multimedia Internet search
- Enhanced access services to audiovisual content on portals
- Personalized video selection and distribution
- Professional audiovisual asset management
- Digitalization and enrichment of library content, audiovisual cultural heritage, and scientific information.

Also included in Quaero is basic research in the technologies underlying these application areas, including automatic speech recognition, machine translation, and speech-to-speech translation. The vision of Quaero is to give the general public as well as professional user the technical means to access various information types and sources in digital form, that are available to everyone via personal computers, television, and hand-held terminals, across languages.

One of the technologies under investigation within Quaero is *automatic speech recognition* (ASR), i.e. the automatic transcription of human speech into written record. Within Quaero research is driven by competitive evaluation and sharing of results and technologies employed. This process is called *coopetition*. Evaluations are conducted once a year on a predefined domain and a set of languages. As the project continues the number of languages to address will grow. Also the performance of the recognition systems developed within the project is expected to improve.

The evaluation conducted in fall of 2011 was the third full-scale evaluation of ASR technology within Quaero and has shown considerable progress in the systems for the Quaero domain on the languages that have been developed for the last years. The test data for the evaluation consisted of various audio files collected from the World Wide Web, including broadcast news, lectures, and video blogs.

1.2. Structure

The rest of this paper is structured as follows. Section 2 provides a description of the front-ends used in our set-up and we evaluate our new bottleneck feature front-end. An overview of the techniques and data used to build our acoustic models is given in Section 3. Details of language models are provided in Section 4. Section 5 describes the grapheme based system and its influence on the WER. Our decoding strategies are explained in Section 6 with Section 7 detailing the evaluation results.

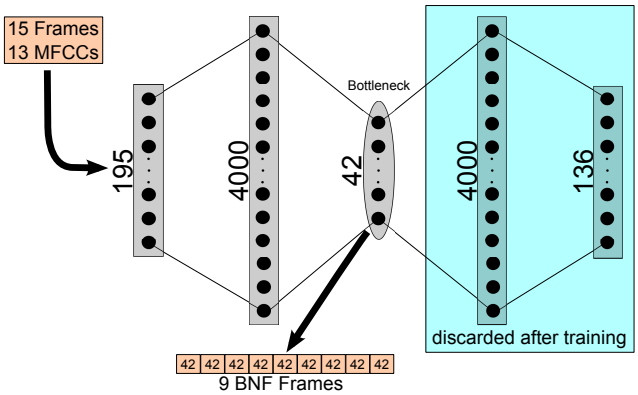

Figure 1: *The MLP architecture (4kx4k) that performed best in our experiments: A 15 frame context window, with 13 MFCCs each, was used as the input feature; the 136 node target layer (one node per sub-phone) and the 4k 3rd hidden layer were discarded after the MLP was trained. A 9 frame context window of the MLP output at the 42 node bottleneck layer is then used as the new 378 dim BNF feature.*

2. Front-ends

We trained systems for two different kinds of acoustic front-ends. One is based on the widely used *mel-frequency cepstral coefficients* (MFCC) obtained from a discrete Fourier transform and the other on the *warped minimum variance distortionless response* (MVDR). The second front-end replaces the Fourier transformation with a warped MVDR spectral envelope [1], which is a time domain technique to estimate an all-pole model using a warped short time frequency axis such as the mel-scale. The use of the MVDR eliminates the overemphasis of harmonic peaks typically seen in medium and high pitched voiced speech when spectral estimation is based on linear prediction.

For training, both front-ends have provided features every 10 ms. During decoding this was changed to 8 ms after the first stage. In training and decoding, the features were obtained either with the Fourier transformation followed by a mel-scale filter bank or the warped MVDR spectral envelope. For the MVDR front-end we used a model order of 22 without any filter bank since the warped MVDR already provides the properties of the mel-scale filter bank, namely warping to the mel-frequency and smoothing. The advantage of this approach over the use of a higher model order and a linear filter bank for dimensionality reduction is an increase in resolution in low frequency regions which cannot be attained with traditionally used mel-scale filter banks. Furthermore, with the MVDR we apply an unequal modelling of spectral peaks and valleys that improves noise robustness, due to the fact that noise is mainly present in low energy regions.

Both front-ends apply *vocal tract length normalization* (VTLN) [2]. For MFCC this is done in the linear domain, for MVDR in the warped frequency domain. The MFCC front-end uses 13 cepstral coefficients, the MVDR front-end uses 15. The mean and variance of the cepstral coefficients were normalized on a per-utterance basis. For both front-ends 15 adjacent frames were combined into one single feature vector. The resulting feature vectors were then reduced to 42 dimensions using *linear discriminant analysis* (LDA).

2.1. Bottle Neck Features

In recent years neural network based features have been shown to improve ASR systems [3]. A typical setup involves training a neural network to recognize phones (or phone-states) from a window of ordinary (e.g. MFCC) feature vectors. With the help of a hidden bottleneck layer the trained network can be used to project the input features onto a feature with an arbitrarily chosen dimension; a schematic of our setup can be seen in Figure 1 [4]. The input vector is derived from a 15 frame context window with each frame containing 13 MFCCs. Previously we used LDA to reduce the dimensionality of this input vector which limits the resulting LDA-features to linear combinations of the input features. A *multi layer perceptron* (MLP) with the bottle-neck in the 2nd hidden layer can make use of non-linear information.

The MLP was trained with all 313 hours of audio data available. After using our Janus ASR toolkit to extract and align the required features the quicknet tool [5] was used to train the MLP. Our basic topology consisted of a 195 node input layer, a first hidden layer having 2000 to 4000 nodes, and the 42 nodes bottleneck layer. Between the bottleneck layer and the 136 node output layer lies an optional 2000–4000 node 3rd hidden layer. We refer to a network with 2000 nodes in the first hidden layer and without a 3rd hidden layer as *2k*, networks including a 3rd hidden layer, e.g. also with 2000 nodes, are then named *2kx2k*.

Running a 2nd MLP training pass with only the 100 hours of 2011 Quaero training data gave us a constant 0.2-0.4 percentage point (pp) improvement. After a series of experiments using different topologies we came to the conclusion that larger (non-bottleneck) hidden layer(s) perform better (see Table 1). The inclusion of the 3rd hidden layer between the bottleneck layer and the output layer resulted in further improvements. We also examined how topology changes affected the final system combination result. The bottleneck features provided us with further gains when we included the first pass output of our BNF system in a first CNC and adapted all 2nd pass systems on its results. With the 4k BNF system this resulted in a 0.45 pp reduction in WER compared to adapting the 2nd pass systems only on their first pass output and then combining them.

3. Acoustic Modeling

For a given front-end our standard method of training an acoustic model requires first performing an LDA to reduce the input dimension. All models are context dependent quinphone systems with three states per phoneme, and a left-to-right topology without skip states. All models use 6,000 dis-

Topo	EM Training	System Combination
2k	19.29%	17.27% / -
3k	18.99%	-
4k	18.99%	17.12% / 16.67%
2kx2k	19.10%	-
4kx4k	18.66%	- / 16.63%

Table 1: *Comparison of different bottleneck features. The EM Training column refers to a single BNF system trained to that stage. The System Combination column displays the WER of the final CNC of all 3 2nd pass systems, either self adapted or adapted on the CNC of the first pass.*

tributions and codebooks compared to 2,000 in our 2010 setup. Simply increasing the number of distributions and codebooks improved our initial system from a WER of 23.71% to 22.57%. The models were trained using *incremental splitting of Gaussians* (MAS) training, followed by *semi-tied covariance* (STC) [6] training using one global matrix, and 2 iterations of Viterbi training. All models use *vocal tract length normalization* (VTLN). In addition to that *maximum likelihood linear regression* (MLLR) [7] and *feature space MLLR* (fMLLR) *speaker adaptive training* (SAT) [8] was applied on top.

While in the past we improved the *expectation maximization* (EM) trained models further with the help of *maximum mutual information estimation* (MMIE) training [9], this year we performed discriminative training with *boosted MMIE* (bMMIE) training (see Section 3.3). We applied bMMIE training firstly to the models after the 2 Viterbi training iterations, and secondly to the models after the fMLLR-SAT training, taking the adaptation matrices from the last iteration of the fMLLR-SAT training and keeping them unchanged during the bMMIE training.

3.1. Training Data

In addition to the supplied 200 hours of Quaero audio training data we also used 11 hours of broadcast news data and 95 hours of EPPS training data. This data was filtered based on absolute segment duration (discard if > 300 s) and relative phone duration (discard if < 0.03 s or > 10 s). Relative phone duration is measured as utterance duration divided by number of letters in utterance, as there is a close relation between letters and sounds in Spanish. The 10 s allow for noise phones to have long durations. Also utterances containing *+noise_nontrans+* were discarded. Figure 2 presents an overview of the used training data and the results of the filtering.

3.2. Shared Memory Training

Computational speed for training the ASR systems was increased by using the RAM disk temporary file storage facility (tmpfs) on the clusters computing nodes.

Corpus	unfiltered	filtered
Broadcast News Speech Corpora	11:00:21	10:59:42
TC_STAR EPPS Transcriptions	100:17:45	100:14:58
Quaero 2009 dev data	2:28:18	2:16:21
Quaero 2010 training data	95:23:37	58:46:55
Quaero 2011 training data	104:08:58	99:47:22
total	313:19:01	272:05:21

Table 2: *Acoustic Model training data before and after filtering*

Step	tmpfs	NAS (1 node)	NAS (4 nodes)
LDAs	$\approx$ 16 min	$\approx$ 20 min	$\approx$ 5 min
Samples	$\approx$ 9 min	$\approx$ 68 min	$\approx$ 17 min
MAS	$\approx$ 9 min	$\approx$ 104 min	$\approx$ 26 min
OFS	$\approx$ 116 min	$\approx$ 184 min	$\approx$ 46 min
Viterbi	$\approx$ 54 min	$\approx$ 80 min	$\approx$ 20 min
total	$\approx$ 204 min	$\approx$ 456 min	$\approx$ 114 min
SAT	$\approx$ 194 min	$\approx$ 252 min	$\approx$ 63 min

Table 3: *Runtime of different training steps comparing use of tmpfs (RAM disk) and shared network memory (NAS). All training steps using tmpfs are run on a single node, training steps using NAS are run on 4 nodes. The middle column, NAS (1 node), was computed from the NAS (4 node) column in order to better demonstrate the resources saved by using tmpfs. SAT being optional is not included in the total time.*

In order to achieve an acceptable computation time the training steps are parallelized by splitting the training data. Most steps only require a single final merging step. For past systems a shared network memory partition (network-attached storage (NAS)), which all nodes can access, was used since this merging can only be done by a single process and all the fragmental results need to be available to that process. This method however leads to enormous cluster network traffic and therefore large memory access times.

The new approach uses the nodes tmpfs, a small RAM disk partition, which can be used by all processes running on one node. Since the tmpfs is in local RAM the access times are short and there is no need to send data over the network. We limited the maximal number of processes per step to 16, the number of cores in a node. Table 3 compares the runtime of several training steps computed on one node with 16 cores using tmpfs to their runtime using shared network memory and more nodes, with an extra column showing the hypothetical runtime of the NAS setup if only a single node is used. It can be seen that most steps are absolutely faster using tmpfs or at least relatively faster according to the number of processes.

	MMIE	bMMIE
MFCC - 1st pass	20.04%	18.99%
MVDR - 1st pass	19.95%	19.45%
BNF - 1st pass	18.66%	18.02%
1st CNC	18.18%	17.22%
MFCC - 2nd pass	17.77%	17.33%
MVDR - 2nd pass	17.93%	17.20%
BNF - 2nd pass	17.68%	17.85%
2nd CNC	16.63%	15.99%

Table 4: *Improvments from replacing MMIE with BMMIE*

Text corpus	Word Count	sources
EPPS & news texts	245.7 million	9
Gigaword	1050.5 million	4
Quaero 2010 data	1180.0 million	5
Quaero 2011 data	459.4 million	17
google Ngrams	1.6 bln ngrams	1
total	2935.6 million	36

Table 5: *Language Model training data word count per corpus and number of text sources included in corpus. The total word count does not include the google ngrams.*

3.3. Boosted MMIE

Boosted MMIE (bMMIE) is an updated version of MMIE proposed by Povey [10] where the lattice confusions with the largest phone error are given more weight, *boosted*, in order to improve the discriminative capability of the acoustic model. We replaced MMIE training with boosted MMIE training in all three of our systems. As can be seen in Table 4, all of our individual systems improved with use of bMMIE, with improvements varying about 0.4% to 1.0% absolute.

4. Language Modeling

A 4gram case sensitive language model with modified Kneser-Ney smoothing was built for each of the text sources listed in Table 5. This was done using the SRI Language Modelling Toolkit [11]. The effects of the different text sources on the performance of language model can be seen in Table 6. The transcripts of the Quaero training data were cleaned and split into a $1,097k$ word training set and a $390k$ word tuning set. The aforementioned language models built from the text sources in Table 5 were interpolated using interpolation weights estimated on this tuning set resulting in a 20 GB language model with $59,293k$ 2grams, $153,979k$ 3grams and $344,073k$ 4grams.

4.1. Vocabulary Selection

To select the vocabulary we used the same tuning set that we used to estimate LM interpolation weights. For each of our Spanish text sources (see Table 5) we built a Witten-Bell smoothed unigram language model using the union of the text sources' vocabulary as the language models' vocabulary (global vocabulary). With the help of the maximum likelihood count estimation method described in [12] we found the best mixture weights for representing the tuning set's vocabulary as a weighted mixture of the sources' word counts thereby giving us a ranking of all the words in global vocabulary by their relevance to the tuning set. The top 150k words were selected as our vocabulary. Unknown pronunciations were automatically generated using a set of grapheme to phones rules.

System	discription	WER
LM01	Baseline	20.79%
LM04	+googleNgrams +gigaword	19.78%
LM05	+Quaero 2011 text data	19.80%
LM06	+retuning with Q2011 transcripts	19.72%

Table 6: *Language Model development. The the tuning set used to estimate the interpolation weights in LMs 1 through 5 did not contain the transcripts of the Quaero 2011*

4.2. Memory Mapped LM

The final LM, LM06, contains over 500 million n-grams and at 17 GB data size was more than 11 times larger than our 2010 language model. Even compressed in an easy to load binary format our language model required about 4.5 GB of RAM. Our ASR system deals with this by loading the language model into a region of shared memory and allows multiple decoder instances running on different cores to access it. On a fully utilized 16 core compute node for example the language model will only require about 0.3 GB per instance.

5. Grapheme Based System

In order to get acoustic models that contain more diverse information and that produce complementary outputs for system combination and cross-adaptation, we trained models that use graphemes as sub-word units, instead of phonemes.

The feasibility of using graphemes instead of phonemes in ASR systems has been shown in several different works [13, 14, 15]. How well the use of graphemes works, heavily depends on the language and the nature of its grapheme-to-phoneme relation. In general, however, grapheme based systems produce higher word error rates than phoneme based ones—also in our case here. However, when using the grapheme based models in a system combination and adaptation we see improvements.

Table 7 compares the results of the individual phoneme and grapheme based systems and how the inclusion of the grapheme based system into the system combination reduces the word error rate. It shows that including the grapheme based models in the combination lowers the word error rate

Single systems		
#	System	WER
1	MFCC	17.33%
2	MVDR	17.20%
3	BNF	16.85%
4	Grapheme based BNF system	18.38%
System combinations		
#	System	WER
1+2+3	CNC 1st pass	17.22%
1+2+3+4	CNC 1st pass	16.76%
1+2+3	CNC 2nd pass	15.99%
1+2+3+4	CNC 2nd pass	15.68%

Table 7: *Results of system combinations with and without grapheme based systems.*

by 0.46 percentage points in the 1st pass and 0.31 pp in the 2nd pass.

The training set-up for the grapheme based systems was the same as for the phoneme based systems. Since there was no former grapheme based system it was not possible to write fixed alignments and VTLN parameters similar to the phoneme based systems. We initialized the grapheme models by bootstrapping from the phoneme models using a rough, manually created mapping, instead.

When clustering quinphone models for the graphemes we used only questions about the identity of graphemes in the context of the poly-graphemes, as this is known to perform quite well [14].

6. Decoding Strategy

The decoding was performed with the *Janus Recognition Tool-kit* (JRTk) developed at Karlsruhe Institute of Technology and Carnegie Mellon University [16]. Our decoding strategy is based on the principle of system combination and cross-system adaptation. System combination works on the principle that different systems commit different errors that cancel each other out. Cross-system adaptation profits from the fact that the unsupervised acoustic model adaptation works better when performed on output that was created with a different system that works approximately equally well [17]. The set-up used for our evaluation system consists of two stages. In each stage multiple systems are being run, and their output is combined with the help of *confusion network combination* (CNC) [18]. On this output the acoustic models of the next stage are then adapted using *Vocal Tract Length Normalization* (VTLN) [2], *Maximum Likelihood Linear Regression* (MLLR) [7], and *feature space constrained MLLR* (fMLLR) [19].

6.1. Segmentation and Speaker Clustering

Segmenting the input data into smaller, sentence-like chunks used for recognition was performed with the help of a fast decoding pass on the unsegmented input data in order to determine speech and non-speech regions [20]. Segmentation was then done by consecutively splitting segments at the longest non-speech region that was at least 0.3 seconds long. The resulting segments had to contain at least eight speech words and had to have a minimum duration of six seconds and a maximum length of 30 seconds.

In order to group the resulting segments into several clusters, with each cluster, in the ideal case, corresponding to one individual speaker we used the same hierarchical, agglomerative clustering technique as last year which is based on *tied Gaussian mixture model generalized likelihood ratio* TGMM-GLR distance measurement and the *Bayesian Information Criterion* (BIC) stopping criterion [21]. The resulting speaker labels were used to perform acoustic model adaptation in the multi-pass decoding strategy described below.

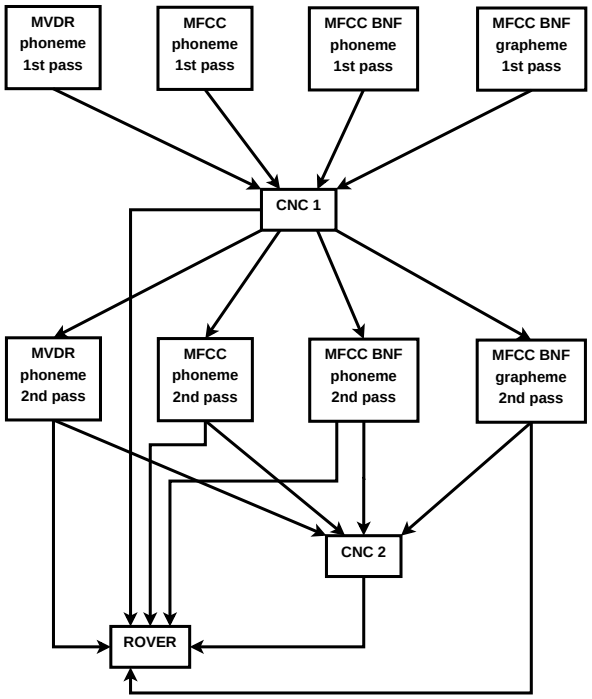

Figure 2: *Architecture of the KIT Spanish speech recognition system.*

6.2. ROVER Combination

The final step in our system decoding set-up is the ROVER combination of several outputs [22]. We optimized the selection of languages and combination method by trying out several set-ups on the development set. It turned out that a majority vote among the first, second CNC and all other system outputs from the second stage, gave the best results.

7. Evaluation Results

We evaluated our systems using both the Quaero 2010 development data as well as the Quaero 2010 evaluation data.

System	WER
KIT	20.79%
+ LM2011	19.80%
+ Quaero2011 data + 6000er Tree	18.03%
+ BNF(2k)	17.27%
+ BNF(4k)	17.12%
+ X-adapt with BNF(4k)	16.67%
+ BNF(4kx4k)	16.63%
+ BMMIE	15.99%
+ Grapheme System	15.68%
+ new Segmentation	14.87%
+ ROVER	14.53%

Table 8: *WER of our System as we gradually added new techniques*

The combined dataset contained about 6 hours of audio gathered from pod-casts with about 300 different speakers. Whereas our 2010 system only contained two subsystems (MFCC&MVDR) we now employ 4 subsystems (MFCC, MVDR, BNF & GRAPHEME) for each pass and adapt the 2nd pass on the CNC of the first. Table 8 presents the final system improvement of each new step that we performed. The first few steps (*+LM2011*, *+Quaero2011 data* & *+6000er Tree*) resulting in a total improvement of 2.66% mainly involve using more data and increasing some parameters. The addition of the BNF(2k) system initially contributes 0.75% to the total WER reduction and although topology changes provide some slight improvements it was not until we performed a 1. pass CNC including the BNF system on which all 2nd pass systems were adapted that we were able to get an absolute improvement of 1.4% out of our new BNF system.

8. Conclusion

In this paper we presented our Spanish LVCSR system, with which participated in the Quaero 2011 evaluation. We describe the incorporation of new features over the 2010 systems, such as the use of bMMIE training, bottle-neck features, and grapheme based systems. In combination the addition of the new features reduces the WER on the Quaero task by 30% relative.

9. Acknowledgements

This work was realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation. 'Research Group 3-01' received financial support by the 'Concept for the Future' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative.

10. References

[1] M. Wölfel and J. McDonough, "Minimum variance distortionless response spectralestimation, review and refinements," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, September 2005.

[2] P. Zhan and M. Westphal, "Speaker normalization based on frequency warping," in *ICASSP*, Munich, Germany, April 1997.

[3] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On using mlp features in lvcsr," in *Proceedings of ICSLP*. Citeseer, 2004.

[4] F. Metze, R. Hsiao, Q. Jin, U. Nallasamy, and T. Schultz, "The 2010 cmu gale speech-to-text system," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[5] D. Johnson *et al.*, "Icsi quicknet software package," *on-line] http://www. icsi. berkely. edu/Speech/qn. html*, 2007.

[6] M. Gales, "Semi-tied covariance matrices for hidden markov models," Cambridge University, Engineering Department, Tech. Rep., February 1998.

[7] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[8] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Cambridge University, Engineering Department*, May 1997.

[9] D. Povey and P. Woodland, "Improved discriminative training techniques for large vocabulary continuous speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, USA, May 2001.

[10] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4057–4060.

[11] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *ICSLP*, 2002.

[12] A. Venkataraman and W. Wang, "Techniques for effective vocabulary selection," *Arxiv preprint cs/0306022*, 2003.

[13] C. Schillo, G. A. Fink, and F. Kummert, "Grapheme based speech recognition for large vocabularies," in *Proceedings of the Sixth International Conference on*

Spoken Language Processing (ICSLP 2000). Beijing, China: ISCA, October 2000, pp. 584–587.

[14] M. Killer, S. Stüker, and T. Schultz, "Grapheme based speech recognition," in *Proceedings of the 8th European Conference on Speech Communication and Technology EUROSPEECH'03*. Geneva, Switzerland: ISCA, September 2003, pp. 3141–3144.

[15] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition," in *Proceedings the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, vol. 1. Orlando, Florida, USA: IEEE, 2002, pp. 845–848.

[16] H. Soltau, F. Metze, C. Fuegen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *ASRU*, 2001.

[17] S. Stüker, C. Fügen, S. Burger, and M. Wölfel, "Cross-system adaptation and combination for continuous speech recognition: The influence of phoneme set and acoustic front-end," in *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech 2006, ICSLP)*. Pittsburgh, PA, USA: ISCA, September 2006, pp. 521–524.

[18] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, October 2000.

[19] V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker adaptation using constrained estimation of gaussian mixtures," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 5, pp. 357–366, 1995.

[20] S. Stüker, C. Fügen, F. Kraft, and M. Wölfel, "The isl 2007 english speech transcription system for european parliament speeches," in *Proceedings of the 10th European Conference on Speech Communication and Technology (INTERSPEECH 2007)*, Antwerp, Belgium, August 2007, pp. 2609–2612.

[21] Q. Jin and T. Schultz, "Speaker segmentation and clustering in meetings," in *Proceedings of the 8th International Conference on Spoken Language Processing (Interspeech 2004 — ICSLP)*. Jeju Island, Korea: ISCA, October 2004.

[22] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proceedings the IEEE Workshop on Automatic Speech Recognition and Understanding*. Santa Barbara, CA, USA: IEEE, December 1997, pp. 347–354.