The 2013 KIT Quaero Speech-to-Text System for French

Joshua Winebarger, Bao Nguyen, Jonas Gehring, Sebastian Stüker, and Alexander Waibel

Institute for Anthropomatics
Karlrsuhe Institute of Technology, Karlsruhe
{joshua.winebarger|quoc.nguyen|jonas.gehring|sebastian.stueker|alex.waibel}@kit.edu

Abstract

This paper describes our *Speech-to-Text* (STT) system for French, which was developed as part of our efforts in the Quaero program for the 2013 evaluation. Our STT system consists of six subsystems which were created by combining multiple complementary sources of pronunciation modeling including graphemes with various feature front-ends based on deep neural networks and tonal features. Both speaker-independent and speaker adaptively trained versions of the systems were built. The resulting systems were then combined via confusion network combination and cross-adaptation. Through progressive advances and system combination we reach a word error rate (WER) of 16.5% on the 2012 Quaero evaluation data.

1. Introduction

1.1. The Quaero Speech-to-Text Task

Quaero (http://www.quaero.org) is a French research and development program with German participation. The focus is to develop multimedia and multilingual tools with professional and general public applications in such domains as automatic extraction, analysis, classification, and exploitation of information. The vision of Quaero is to provide public and professional users with the means to access various information types and sources in digital form. Quaero proposes to achieve this by creating a framework for collaboration between complementary technological ventures such as businesses, public research institutions, and universities through competitive evaluations and sharing of the research thereby created in a process called "coopetition." Partners also collaborate on advanced demonstrations and prototypes and work to develop and commercialise the resulting applications and services.

One of the technologies researched within Quaero is *Automatic Speech Recognition*, i.e. the automatic transcription of human speech into written form. This is known as the speech-to-text task. In line with the concept of coopetition, evaluation of ASR technological development in Quaero is done once a year. The domain is a mix of broadcast news and broadcast conversational speech, the latter of which is more challenging for automatic recognisers than read speech due to the presence of disfluencies and non-speech events such as music and spontaneous human noises. The number of languages included in the program has increased, as has the expected state-of-the-art recognition system performance. Being a French project with European orientation, French is naturally among the languages evaluated. The fall 2013 evaluation was the fifth and final full-scale evaluation of ASR within Quaero. The test data for the evaluation consisted of audio from various web sources including broadcast news, video blogs, and lectures. At the time of this writing the evaluation was not yet completed. Therefore this paper reports our results on the 2012 evaluation data, which we used as our development set.

1.2. Paper Structure

The paper is structured as follows. Section 2 describes the acoustic data and training techniques of our system. We describe the front-end processing used, including deep neural networks and tonal features. Our efforts to develop diversified pronunciation modeling are the focus of Section 3. We give special attention to the use of graphemes. Then, we make a detailed description of our language model and its development in Section 4. In Section 5 we describe our overall recognition setup used in the evaluation and give performance figures for the 2012 evaluation data. Section 6 describes experiments done in the development of our system. Finally we discuss opportunities for future work and conclude the paper in Section 7.

2. Acoustic Modeling

We trained several acoustic models based on different pronunciation dictionaries and feature front-ends. The pronunciation modeling aspect is described in detail in Section 3. Each pronunciation model (dictionary) was essentially based on its own phoneme set. The feature front-ends can be generally described as deep bottlneck features based on either *i)* 40 log mel filter bank coefficients (lMEL) or *ii)* a stacked combination of 20 mel-frequency cepstal coefficients (MFCC,) 20 warped minimum variance distortionless response coefficients (MVDR,) and tonal features (a setup we call *M2+T*), described in Section 2.3.

All acoustic models are based on HMMs, whose states correspond to generalized quinphones with three states per phoneme, and a left-to-right topology without skip states. The generalized quinphones were found by clustering the quinphones in the training data using a decision-tree. We found 8,000 acoustic models performed best in all subsystems except the grapheme subsystem, for which 12,000 models performed best. The models were trained using incremental splitting of Gaussians (also known as merge and split or MAS training.) For all models we then estimated one global semi-tied covariance (STC) matrix after LDA [1], and refined the models with two iterations of viterbi training. All models use vocal tract length normalization (VTLN.) For a second-pass decoding (see Section 5) speaker adaptive models are trained using feature space constrained MLLR [2, 3]. For certain systems we extended the expected maximisation of the models with discriminative training based on the boosted Maximum Mutual Information Estimation (bMMIE) criterion [4], which saw reductions in word error rate (WER) of between 2-2.5% (relative) compared to the maximum likelihood systems.

2.1. Training Data

Each subsystem was trained on 268 hours of speech coming from the Broadcast News (BN) and Broadcast Conversation (BC) domain. We used the Quaero training data from 2009-2011 as well as data from the Ester campaign [5]. Both datasets provide manual transcripts and speaker clustering. The Quaero data can be divided into portions for which the transcripts are "fast" or carefully annotated. Those using careful annotations have speakers identified by name, even across shows and audio files, whereas "fast" annotated transcripts use automatic speaker annoatations. Also, the carefully annotated transcripts have a more comprehensive and detailed transcription of noises, disfluencies, and hesitations. We used a technique for filtering this acoustic data by decoding on it, which is described in Section 6.1. Before filtering we had 194.1 hours of Quaero data and 107.8 hours of Ester data. After filtering we had 187.7 hours of Quaero data and 80.3 hours of Ester data, which was used for training.

2.2. Deep Neural Networks as Features

Bottleneck features (BNFs) from multilayer perceptrons have become a staple component in ASR, due to their discriminative power and robustness to speaker and environment variations. Gehring et. al. recently introduced a deep bottlneck feature (DBNF) archictecture based on deep neural networks (DNNs) consisting of many hidden layers, which was shown to achieve significant reductions in WER. [6, 7]

For our French system we trained deep bottleneck feature networks for each subsystem from the best existing MFCC system alignment. Input to the network takes place on an input layer accepting stacked MFCC, MVDR and Tonal features or log mel coefficients. The network consists of five

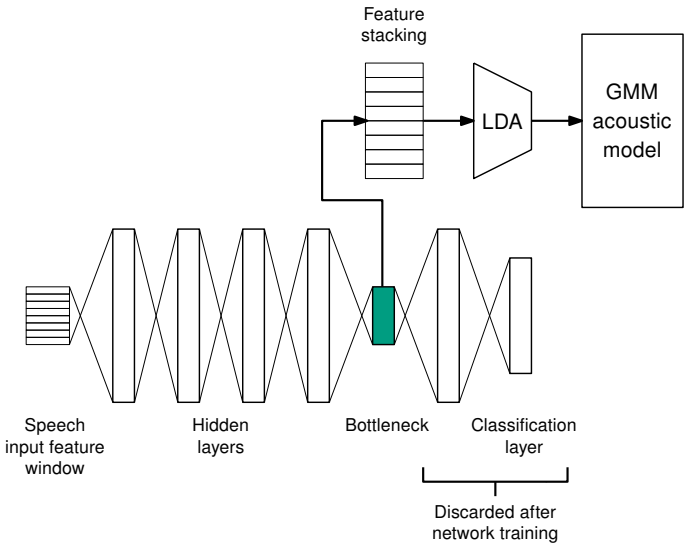

Figure 1: *Deep bottleneck architecture used for feature extractation in our systems*

fully-connected hidden layers containing 1,200 units each, followed by the bottleneck layer with 42 units, a further hidden layer and the final layer, as can be seen in Figure 1.

Layers prior to the bottleneck are pre-trained in a layer-wise, unsupervised manner as a stack of denoising auto-encoders [8]. After the stack of auto-encoders has been pre-trained, the bottleneck layer, the next hidden layer, and the classification layer are intialized with random weights and connected to the hidden representation of the top-most auto-encoder. The network is then trained with supervision to estimate the context-dependent HMM polyphone states. Fine tuning is performed for 14-18 epochs using the "newbob" learning rate schedule which starts with a high learning rate until the increase in accuracy on a validation set drops below a set threshold. The learning rate is then halved for each epoch until improvement in validation accuracy drops below a second threshold, at which point learning is stopped. The activations of the 42 bottleneck units are stacked over an 11- to 13-frame context window and reduced to a dimensionality of 42 using LDA.

Our context-dependent systems were then trained with these networks using the existing polyphone tree and alignment computed without DBNFs. Relative to MFCC, we saw an average word error rate reduction of approximately 22% for systems using lMEL DBNFs and 24% for those using DBNFs based on MFCC and MVDR. This is comparable to gains we saw in development of recognisers for other languages, where the same or similar DBNF architecture was employed.

2.3. Tonal Features

Conventional wisdom in ASR asserts that pitch or "tonal" information is not helpful in building speech recognisers for

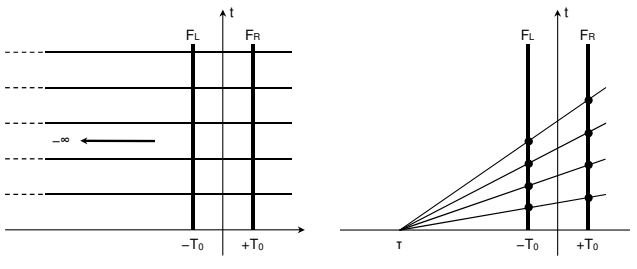

Figure 2: *Visualization of the vanishing point product employed in FFV. When $\tau \rightarrow \pm\infty$, the vanishing point product reduces to the standard inner product, shown on the left. On the right, we see F_R dilated by a negive value of τ.*

non-tonal languages (such as French.) However it was recently shown that pitch information can be integrated into an ASR in a manner that improves recognition accuracy for both tonal and non-tonal languages [9]. Fundamentally different from spectral features, which capture the envelope of the speech signal, pitch features capture variations in the fundamental frequency of the speaker's voice. Our DBNFs based on a concatenation of MFCC and MVDR coefficients incorporated two such tonal features derived from the pitch of the speech signal. These are the pitch and Fundamental Frequency Variation (FFV.)

We extract pitch features according to the method in [10]. First, a cepstrogram is computed with a window length of 32 ms. We detect the position of the maximum of all cepstral coefficients starting with the 30^{th} coefficient. Dynamic programming is then used to find a path that maximises the correlation between coefficients subject to constraints such as the maximum pitch change per unit time. Additionally we consider the position of the three left and right neighbours, as well as their first and second derivatives, resulting in seven pitch coefficients.

FFV features [11], typically used for tasks such as speaker verification, have the advantage that no explicit segmentation into speech and silence (for which pitch is undefined) is necessary. The change in fundamental frequency is not computed by tracking a single value of F_0 over time. Rather a "vanishing point product" is computed between two feature vectors obtained from two asymmetric windows covering the left-half and right-half portion of the general feature window. This vanishing point is equivalent to an inner product between left and right spectrums F_L and F_R, where F_L or F_R are dilated with respect to one another by positive or negative values of τ, respectively. This vanishing point product is depicted graphically in Figure 2. Afterwards, a filterbank is applied which attempts to capture meaningful prosodic information. The filterbank reduces the input space to seven scalars per frame. Previous experiments showed that the best way to integrate these features is through their concatenation with the MFCC and MVDR coefficients in the input vector for DBNF training. [9]

By concatenating tonal features in a 32 millisecond window with MFCC and MVDR coefficents (*M2+T*) for the input layer of our DBNFs, we reduced our error rate on the 2011 Quaero evaluation set by an additional 3% relative to MFCC and MVDR (*M2*) alone. This is comparable to the 3% relative improvement seen for KIT's English system developed for IWSLT. This 3% for non-tonal languages can be compared with the 5% relative improvement seen for Vietnamese, a tonal language.

3. Pronunciation Modeling

In an effort to develop subsystems that produce diverse, complementary output for system combination and cross adaptation we employed different pronunciations modelings. For training and testing, we used pronunciation dictionaries based on four sources:

1. The popular BDLex lexicon, which gives a wide range of pronunciation variants
2. The Globalphone dictionary
3. A rule-based pronunciation generation called text2phone, typically used for TTS applications [12]
4. A pseudo-grapheme-based approach

After a straightforward mapping, the BDLex and Globalphone dictionaries shared essentially the same phone set. That of BDLex contains 45 phones, among them five noise phonemes: hesitation, incomplete words, human noises such as coughing, non-human noises such as music, and a catch-all noise. The Globalphone phone set adds a voiceless glottal fricative *h*, an additional open-middle vowel, and a breath noise. It also differs slightly in the classification of phones. For the text2phone system we used a different, 41-element set of phones, among them the same noise phones as that of BDLex.

For the first two sources of pronunciation, missing pronunciations were generated automatically using grapheme-to-phone models as described in [13]. Acoustic models for systems using the first two pronunciation sources were initialized by bootstrapping from German models using a manually created mapping.

While subsystems based on the BDLex dictionary generally yielded the best performance, we found that the system combination benefited from the inclusion of the output of each additional subsystem in the combination.

3.1. Pseudographeme System

In a traditional grapheme-based system, the symbols of the written word are used as the sub-units of pronunciation rather than phonemes. The feasibility of using graphemes instead of phonemes in ASR has been shown in several different works [14, 15, 16]. It was also shown that the combination

of a grapheme system with phoneme systems lead to a significant reduction in word-error rate [17].

While French orthography is relatively regular vis à vis a language like English, the mapping between sounds and graphemes is not bijective, which is to say that the correspondence between graphemes and phonemes can be weak. Often, clusters of graphemes produce the same sounds as other, shorter ones, such as "-ai" and "-é", which both correspond to the IPA [e]. We handle this weakness by using single or multiple graphemes as the base units of pronunciation. Our set of grapheme-phones contained 49 elements, among them the same five noise phones as in the BDLex system. Using knowledge of French pronunciation we wrote simple mappings determining whether a certain sequence of graphemes should map to one unit or stand as separate units of pronunciation. These mappings consist of a list of grapheme sequences to be merged. We scan from left to right in a word and seek the longest matches possible in the list. The rules are kept simple in that we do not attempt to merge the graphemes in a way that each group has a unique pronunciation, nor do we factor in long-range context in the determination of merging or splitting. Instead, we write the rules with the expectation that the context-dependent nature of the acoustic models to learn the difference for grapheme groups having a context-dependent pronunciation. For example, we expect the acoustic models to learn from polyphone context that "ent" is voiced following an "m" and preceding a word boundary, as in "foncièrement," but that it is silent following "gn" as in "joignent." The following is a selection of some of these rules and examples of the effects of their use on the grapheme sequence of words.

é	←	{ é }, { a i }, { é e }, { u é }
ent	←	{ e n t }
e	←	{ e }, { è }, { ê }
au	←	{ a u }, { e a u }
on	←	{ o n }
oin	←	{ o i }
gn	←	{ g n }

Table 1: *Some selected rules for merging graphemes*

délaissées (*adj. fem. pl.* abandoned)	→	d é l é s é s
faisceaux (*n. m.* bundles)	→	f é s c au x
foncièrement (*adv.* fundamentally)	→	f on c i e r e m ent
joignent (*v. 3p. pl.* join)	→	j oi gn ent
pointée (*adj. fem.* pointed)	→	p oin t é

Table 2: *Selected entries from the grapheme dictionary with accompanying English translations*

Because there was no prior (pseudo-)grapheme system, we trained the system using a flat-start technique based on six iterations of expectation-maximisation (EM) training and regeneration of training data alignments. When clustering quinphone models for the graphemes we used only questions about the identity of graphemes in the context of the polygraphemes, as this is known to perform quite well [15].

3.2. Performance Comparison

The following is a comparison of the performance associated with the use of the various pronunciation models previously mentioned. The context-dependent system training is identical in every way, including feature front-end (MFCC). Only the dictionary or source of pronunciation is varied. The results are given in Table 3. The relatively higher error rate of the grapheme system relative to the phoneme systems is typical of our experience with other ASR languages [17, pg. 202].

Table 3: *Case-insensitve WER resulting from the use of various pronunciation models. The results are from systems using MFCC features and 8000 acoustic models, and are tested with the same language model.*

Dictionary	WER
BDLex	25.4
text2phone	25.6
globalphone	26.6
grapheme	27.0

4. Language Modeling

A 4-gram case-sensitive language model with modified Kneser-Ney smoothing was built for each of the text sources listed in Table 4. This was done using the SRI Language Modeling Toolkit [18]. We cleaned the Quaero acoustic training transcripts and used half as part of the training set; the other half was used as a tuning set. The language models built from these text sources were interpolated using weights estimated on this tuning set; these weights were estimated with a tool in the SRILM toolkit which uses an expectation maximization algorithm with fixed underlying mixture distributions to minimize the perplexity of the LM mixture on the tuning set. The result was a language model with 38.34 million 2-grams, 113.2 million 3-grams, and 233.8 million 4-grams.

4.1. Development

Our baseline language model was trained with newswire text from the Gigaword corpus as well as the Quaero acoustic training transcripts from all years up to 2012.

We achieved improvements in perplexity by including additional data in our training. In subsequent iterations of the language model, we added the Quaero 2012 additional language model training sources, among which are blog data, the newspaper *l'Humanité*, and various other news sources. We also added the transcripts of the Ester corpus [5] as

[1] From the Gigaword corpus

Source	Type	Words	Weight
Quaero transcripts	BC & BN	1M	0.459
Quaero l'Humanité	Newspaper	752M	0.127
AFP, APW, [1] and Ester	Newswire, BN	391M	0.121
Quaero Blog	Blog	62M	0.111
Quaero News Div	Newspaper	150M	0.091
AFP 2000s [1]	Newspaper	335M	0.032
CFPP2000	Interviews	417K	0.029
European parliament	Debate	100M	0.019
Est Républicain	Newspaper	104M	0.011

Table 4: *Summary of the cleaned language model (LM) training texts, including training data, type, word count per corpus, and interpolation weight in the final LM*

well as several other sources of text we found. Among these are the newspaper *Est Républicain*, transcriptions of the European Parliament, and the small conversational corpus CFPP2000 from the University of Paris 3, composed of a collection of interviews of Parisian residents. [19].

We also reduced perplexity through normalisation of elisions, which is described in section 6.2. Last, further improvements were made by normalising the casing of our text sources using smart case models trained on large corpuses of text.

We tested the effect of these development steps by measuring the perplexity of the language model on a text set composed of several Quaero acoustic transcripts: development and evaluation 2009, evaluation 2010, and evaluation 2011. The results are shown in Table 5.

Improvement	Perplexity
Baseline	174.1
+Fast cleaned Quaero 2012 material +elision normalisation with top 50 list	153.8
+Carefully cleaned Quaero 2012 material +Ester transcripts	136.0
+Additional data sources	135.0
+Smart casing	130.3

Table 5: *Perplexity scores of successive iterations of language model development, measured on the combined 2009-2011 Quaero dev. and eval. acoustic transcripts.*

4.2. Vocabulary Selection

For selection of the search vocabulary we employed the same tuning set as used for the estimation of the LM interpolation weights. For each of the aforementioned text sources, we built a Witten-Bell smoothed unigram language model. The vocabulary of this LM was taken as the union of the vocabularies of all text sources. Using the maximum likelihood count estimation described in [20] we found the best mixture weights for representing the tuning set's vocabulary as a weighted mixture of the word counts of the sources. This gave us a ranking of all words in the union vocabulary in terms of their relevance to the tuning set. We found that a vocabulary of 250,000 words gave consistently the best performance in terms of word error rate.

5. Recognition Setup

The decoding was performed with the *Janus Recognition Toolkit* (JRTk) developed at the Karlsruhe Institute of Technology and Carnegie Mellon University [21]. Our decoding strategy is based on the combination and cross-system adaptation of many subsystems trained with different dictionaries and feature front-ends. System combination works on the principle that different systems commit different errors which cancel each other out. Cross-system adaptation profits from the fact that unsupervised acoustic model adaptation works better when performed on output that was created with a different system with comparable performance [22]. By combining subsystems with several diverse configurations, we ensure a variety of outputs upon which subsystem combination can work effectively.

5.1. Segmentation

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 [23]. 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 corresponding in the ideal case to one individual speaker) we used an hierarchical, agglomerative clustering technique based on TGMM-GLR distance measurement and the Bayesian Information Criterion (BIC) stopping criterion [24]. The resulting speaker labels were used to perform acoustic model adaptation in the multipass decoding strategy described below.

5.2. Subsystem Combination and ROVER

We use two passes of decoding. The output lattices of the subsystems in each stage are used to produce an improved output through confusion network combination (CNC) [25]. The second pass decodings are performed using SAT models and unsupervised adaptation. At this stage the subsystem makes use of the confidences of the CNC from the previous stage. As a final step we apply a ROVER for further reduction of error [26].

For each pass of CNC and ROVER we tested several combinations of subsystems in order to find the best CNC or ROVER performance on the development (2012 evaluation) set. Generally speaking, this meant leaving the one or two subsystems with the highest WER out of the CNC. However,

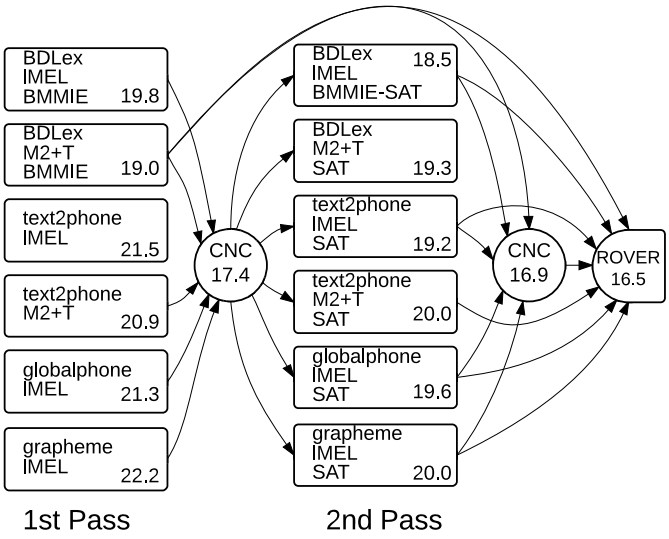

Figure 3: *Decoding strategy of the KIT 2013 evaluation system with WER for each subsystem and step.*

for the ROVER we found that simply including all subsystems gave the best results. Finally, compared to the best CNC without the grapheme subsystem, inclusion of this subsystem consistently improved the CNC by approximately 0.1% absolute WER reduction. Our decoding strategy, along with WER figures for each subsystem, CNC, and the ROVER, is shown in Figure 3.

6. System Development and Experiments

In this section we describe two experiments in the development of our systems which yielded useful improvements. An overview of certain steps and their effect on the performance of the best single system is given in Table 6.

6.1. Filtering of Acoustic Training Material via Decoding

We obtained improvements in WER by undertaking a filtering of the acoustic training material. Rather than using rule-based methods, such as segment duration or relative phone duration, we used our single best system to decode on the training material. We then computed a WER for each utterance in the training database. A new training database was formed wherein those utterances scored with an error rate over a certain limit were not included. We then retrained our system on this slightly smaller, updated training database. This process was done in an iterative method with regard to the incorporation of new training material. Our acoustic training sources were divided into four sets which were each filtered separately:

- "Quaero core" : Quaero partial 2010, 2011 training data
- "Quaero fast" : Quaero 2009, partial 2010 training data with "fast" transcripts

Development step	Improvement
baseline MFCC	27.8
+ various LM improvements	26.5
+ elision normalisation with top50	26.2
+ inclusion of filtered data	25.5
+ 8K models	25.1
lMEL DBNF	19.2
M2 DBNF	19.4
M2+T DBNF	18.7
+ smart casing in LM training	18.6
lMEL DBNF + bMMIE	18.6
lMEL DBNF + SAT	17.7
lMEL DBNF + bMMIE-SAT	17.4

Table 6: *Development steps for the best single system and the resulting reduction in WER (Case Insensitive) relative to the preceding step. The test set is automatically segmented eval. 2011 data.*

Training material	WER
baseline (Quaero core)	27.0
baseline + filt.'d @ 75% additional (fast & careful) Quaero	26.9
baseline + filt.'d @ 50% additional (fast & careful) Quaero	26.5
baseline + filt.'d @ 50% add. Quaero + filt.'d @ 38% Ester	26.3

Table 7: *Improvements in system performance with addition of filtered data. The test set is automatically segmented eval. 2011 data. WER given is Case Insensitive.*

- "Quaero careful" : Quaero 2009, 2010 training data with carefully annotated transcripts
- Ester 1
- Ester 2
- Ester 2-dev

Through successive trials of experimentation we developed the following strategy. First, we adopted a rule to reject those top 10% of utterances having the highest WER. This corresponded roughly to rejecting utterances with WER $> 75\%$. As an alternative we tried rejecting the top 25% of utterances, corresponding roughly to 50% WER. Equivalent systems trained on data filtered with the 50% threshold (more strict) outperformed those trained on data filtered with the 75% threshold (less strict), as is shown in Table 7. As for the Ester data, it differs in that it is solely broadcast news and contains a good deal of telephone-quality speech. Thus we decided to apply a stricter rule of thumb that 38% WER would be the maximum for utterances from Ester.

Table 7 shows the results of successive inclusions of training material, while Table 8 shows the effects of the filtering on the amount of utilised training material.

Training material	Unfilt. utts.	%	Unfilt. hrs.	Filt. hrs.	%
Quaero core	32.5K	100	140.0	-	100
Quaero fast	3.17K	73.4	17.5	14.8	84.76
Quaero careful	9.07K	67.9	36.7	32.9	89.6
Ester 1	11.6K	48.5	61.6	48.3	78.3
Ester 2	6.58K	50.4	40.6	28.2	69.6
Ester 2 dev	1.32K	48.6	5.65	3.76	66.7
Total			302	268	88.7

Table 8: *Effects of decoding-based filtering on training material.*

Elision treatment	S. Vocab oov	WER
join	5.0%	29.6%
sep.	0.61%	26.7%
sep. w/ top50	0.68%	26.2%

Table 9: *OOV rate for 250K-word vocabularies on eval2011 data.*

6.2. Elision normalisation

French contains a phenomenon called elision, wherein the final vowel of one word immediately before another word beginning with a vowel is omitted and by convention the two words are joined with an apostrophe in the written form. For example "ce est" becomes "c'est" ("this is,") "la apparance," becomes "l'apparance" ("the appearance,") "de être" becomes "d'être," ("to be") and so on. When we consider these joined words to be one unit for language modeling (a treatment we call *"join"*) we are faced with a challenge due to a large number of out-of-vocabulary (OOV) words which are simply the elision of two words already in our search vocabulary. The natural solution would then seem to be to consider these units as two separate words (a treatment we call *"separate"* or *"sep."*) While this reduces OOV frequencies, it decreases the language model context. We took a compromise approach by treating the fifty most common elided word combinations in our training transcripts as one word and the rest as two (a treatment we call *"sep. w/ top50."*) This was reflected in our text normalisation.

We selected three 250,000-word search vocabularies from the same data, appropriately filtered in each case to treat elision differenty according to the schemes described above. Table 9 shows the OOV rate of these vocabularies as tested on the 2011 evaluation Quaero transcripts (appropriately processed to reflect the same treatment of elisions.) We see that treating elided combinations as two separate words dramatically reduces OOV. As a further experiment, we trained three otherwise identical recognisers, each reflecting one of these treatments of elision and tested them with a corresponding language model. The effects are shown in the same table (Table 9). The separated approach outperfoms the joined approach, and joining the fifty most common elisions gives further gains.

7. Conclusions

In this paper we presented the KIT French Quaero Speech-to-Text system. We described details of its development through the integration of enhancements such as deep neural networks for features, tonal features, multiple pronunciation models including a modified grapheme scheme, and other development techniques used to improve recognition accuracy. The combination of these techniques was shown to significantly reduce error on this task. Future system development should focus methods on advanced language modeling techniques such as neural networks and techniques developed for inflectional languages in an effort to reduce homophone confusability. Further refinements may also be made to our acoustic neural networks by using multilingual training data and training from new alignments written with neural network systems.

8. Acknowledgements

This work was realised as part of the Quaero Programme, funded by OSEO, the French state agency for innovation.

9. References

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

[2] T. Anastasakos, J. McDonough, and J. Makhoul, "Speaker adaptive training: A maximum likelihood approach to speaker normalization," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 1043–1046.

[3] M. J. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[4] 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.

[5] S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, and G. Gravier, "The ester phase ii evaluation campaign for the rich transcription of french broadcast news." in *Interspeech*, 2005, pp. 1149–1152.

[6] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *ICASSP2013*, Vancouver, CA, 2013, pp. 3377–3381.

[7] Q. B. Nguyen, J. Gehring, K. Kilgour, and A. Waibel, "Optimizing deep bottleneck feature extraction," in *RIVF2013*, Hanoi, Vietnam, 2013.

[8] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[9] F. Metze, Z. Sheik, A. Waibel, J. Gehring, K. Kilgour, Q. Nguyen, and V. Nguyen, "Models of tone for tonal and non-tonal languages," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2013, to appear.

[10] K. Schubert, "Grundfrequenzverfolgung und deren anwendung in der spracherkennung," master's thesis, Universität Karlsruhe (TH), Germany, 1999, in German.

[11] K. Laskowski, M. Heldner, and J. Edlund, "The fundamental frequency variation spectrum," *Proceedings of the Swedish Phonetic Conference (FONETIK 2008)*, pp. 29–32, June 2008.

[12] D. Haubensack, "Text2phone."

[13] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[14] 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.

[15] 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.

[16] 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.

[17] K. Kilgour, C. Saam, C. Mohr, S. Stüker, and A. Waibel, "The 2011 kit quaero speech-to-text system for spanish," *IWSLT-2011*, pp. 199–205, 2011.

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

[19] S. Branca-Rosoff, S. Fleury, F. Lefeuvre, and M. Pires, "Discours sur la ville," *Corpus de français parlé parisien des années*, 2000.

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

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

[22] 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 *INTERSPEECH*, 2006.

[23] S. Stüker, C. Fügen, F. Kraft, and M. Wölfel, "The isl 2007 english speech transcription system for european parliament speeches." in *INTERSPEECH*, 2007, pp. 2609–2612.

[24] 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.

[25] 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.

[26] 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.