GRAPHEME-TO-PHONEME CONVERSION METHODS FOR MINORITY LANGUAGE CONDITIONS

Mengxue CAO[1], *Steve Renals*[2], *Peter Bell*[2], *Aijun LI*[1], *Qiang FANG*[1]

[1]Phonetics Lab, Institute of Linguistics, Chinese Academy of Social Sciences, Beijing, China
[2]School of Informatics, University of Edinburgh, Edinburgh, UK
cmxsmile@gmail.com, s.renals@ed.ac.uk, peter.bell@ed.ac.uk, liaj@cass.org.cn, fangqiang@cass.org.cn

ABSTRACT

This study attempts to investigate the grapheme-to-phoneme conversion approaches for minority language conditions. Instead of isolated-word data for major languages, sentence-form data is defined to be a proper form of training data for minority languages. Joint-multigram Model and Hidden Markov Model were examined in this study. The “treat-sentence-as-word” training method and the forced-alignment process were proposed to extend the Joint-multigram Model and the Hidden Markov Model respectively to meet the minority language conditions. Results get from the sentence-form training data using our proposed methods are as good as the results get from the isolated-word training data using previous proposed methods. The Joint-multigram Model performs better for well-designed training data, while the Hidden Markov Model has more error capacity and is more proper for minority language conditions.

Index Terms — Grapheme-to-phoneme, Joint-multigram Model, HMM, treat-sentence-as-word, forced-alignment

1. INTRODUCTION

Grapheme-to-phoneme conversion represents the process of converting the orthographic form (grapheme) of a word to its symbolic pronunciation representation (phoneme). This process is essential for both speech synthesis and speech recognition. Statistically-based approaches, which require minimum dependencies of hand-written rules, are the state-of-the-art methods [6].

Taylor [5] proposed an approach based on modified Hidden Markov Model (HMM), which successfully applied the high-efficiency HMM algorithm to the entire grapheme-to-phoneme training and conversion process. Different from HMM in speech recognition, phonemes are hidden states, and transitions between phonemes are the probabilities that one phoneme will follow another; graphemes are observations, which are generated according to the probability distributions attached to the states. In addition, looping is not allowed at any state. This framework has proved its highly efficient training process.

Bisani and Ney [2, 3] proposed an approach using joint-multigram to model the grapheme-phoneme sequences. The “variable-length” property of the multigram model enables the Joint-multigram Model to deal with *m-to-0* kind alignment between phonemes and graphemes. The ability to accept parallel sequences of observations makes the training algorithm more powerful. This approach produced significant accuracy-rate advantages over previous methods.

Training data used in previous researches [2, 3, 5, 7, 8] are all based on lexicons directly picked from existing dictionaries. We call this kind of data “isolated-word” training data. Previous training methods work quite well for languages where high-quality pronunciation dictionaries are available. However, the conditions for minority languages are very different. It is impossible for us to have a well-transcribed pronunciation dictionary in advance. In other words, the “isolated-word” approach cannot be applied to grapheme-to-phoneme model training under minority language conditions.

We suggest that, for minority languages, one possible form of training data could be consisted of parallel data of the grapheme transcription sequences of continuous audio sentences and their corresponding phoneme transcription sequences generated from a phone-based speech recogniser. We call this kind of data “sentence-form” training data.

In this study, based on the “isolated-word” approaches proposed by previous researchers, we extended the Joint-multigram Mode by proposing the “treat-sentence-as-word” training method, and enhanced the Hidden Markov Model by involving a forced-alignment process when building high-order N-gram phone models. The minority language condition was simulated using English data sets.

2. CONDITIONS

2.1. Assumptions

Since we are simulating the condition of minority languages using English data, some strict restrictions have to be made to keep our simulation always meaningful. In this study, we assume that we seldom know the language we are training.

Under that assumption, three restrictions have been proposed in this research:

- No linguistic knowledge of the relations between graphemes and phonemes should be involved in the training process.
- No pre-processing should be applied in the training process.
- Words occur in the training data should not be included in the testing data.

2.2. Data Sets

2.2.1. WSJCAM0

WSJCAM0 (WSJ) is a British English corpus based on Wall Street Journal text that was recorded at Cambridge University. There are 90 utterances from each of 92 speakers that are designated as training material [1]. All the training data in WSJ are manually well-transcribed, so it is a very good resource for us to evaluate different models and configurations.

Isolated words and their corresponding pronunciation transcriptions can be extracted from the WSJ data set. There are 141,207 word records in total, among which 11,081 lexicons are found, and 658 of the 11,081 are influenced by the co-articulations in the sentences.

2.2.2. BEEP dictionary version 1.0

The BEEP (British English Example Pronunciations) dictionary is derived in part from the Oxford Text Archive releases 710 and 1054. It is a comprehensive pronunciation dictionary which contains more than 257,000 records. Based on the fact that the BEEP dictionary uses the same phone set as that of the WSJ data set, it is used to derive phoneme transcriptions for the 20K word list of WSJ data set.

2.3. Performance metric

The performance metric in all experiments is Phone Error Rate (PER) which is defined by the fowling formula, where S is the number of substitution errors, D is the number of deletion errors, I is the number of insertion errors, and N is the number of phonemes in the reference.

$$PER = \frac{S + D + I}{N} \qquad (1)$$

2.4. Experiment environment

All of the experiments were performed on the DICE server of the University of Edinburgh. The hardware configurations of the server were: Dell PowerEdge R610s with 48GB of memory and 2.66GHz processors. Since there were other students running their jobs on the server too, the actual resources we could use were smaller than that.

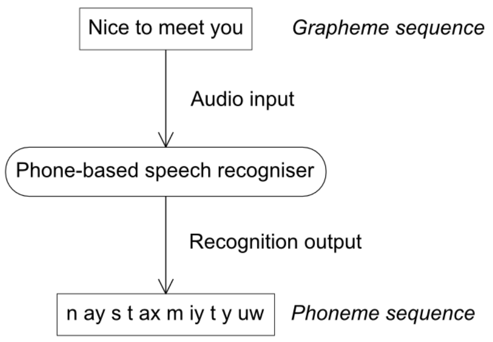


Fig. 1. In minority languages conditions, phoneme transcriptions are gained from a phone-based speech recogniser, but the word boundaries in the phoneme transcription cannot be determined without using linguistic knowledge.

3. TREAT-SENTENCE-AS-WORD FOR JOINT-MULTIGRAM MODEL TRAINING

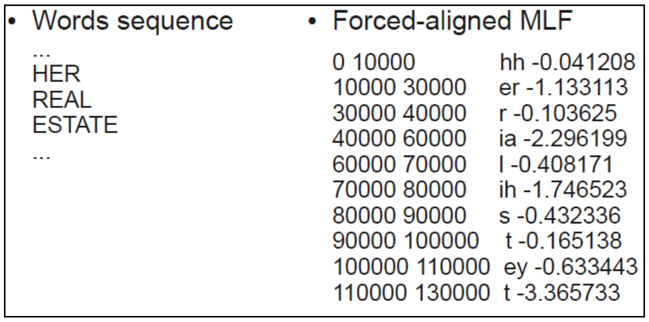
To meet the training condition of minority languages, "treat-sentence-as-word" training algorithm is implemented for the Joint-multigram Model.

Under minority language conditions, training data are gained directly from the transcriptions outputted from a phone-based speech recogniser, as demonstrated in Fig. 1. However, in the output phoneme transcription sequence, the boundaries corresponding to its grapheme transcriptions cannot be determined without using any linguistic knowledge. In this case, our solution is to treat the whole sentence as a "big word". The basic form of the training data looks like followings:

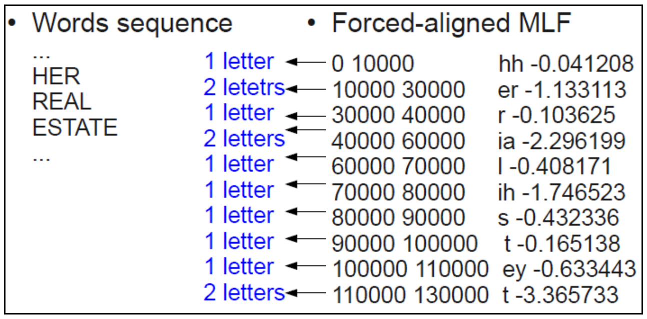
```
...
nicetomeetyou              n ay s t ax m iy t y uw
whathappensnow             w oh t hh ae p ax n z n aw
...
```

As demonstrated, words in the sentence "Nice to meet you" are combined together and formed into a new "big word" — "nicetomeetyou". In this case, sentence becomes "word", and words in the original sentence can then be treated as "syllables" of the "big word". For example, "nice", "to", "meet" and "you" can be treated as four "syllables" of the "big word" "nicetomeetyou". Based on that kind of transposition, we have deleted all "sil" phonemes in the original transcription, since "sil" would bring context interference, and also, in reality, "sil" should not be existed in the phoneme transcription of a word. Finally, we denote /n ay s t ax m iy t y uw/ as the phoneme transcription of the "big word" — "nicetomeetyou".

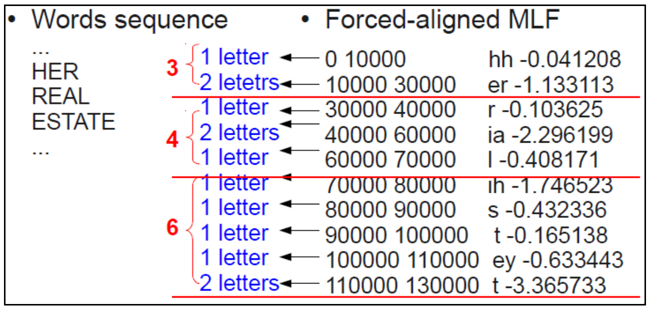
By treating sentence as word, the length of the "big word" could become a problem. In some long sentences, the letters in the "big word" can be 100 or even more, so we suppose it would be a lot harder for the model to find correct



(a) Time alignments in the forced-aligned MLF are used to decide the boundaries in the phoneme transcription. 1000 is a time unit.


• Words sequence
…
HER
REAL
ESTATE
…
• Forced-aligned MLF
1 letter 0 10000 hh -0.041208
2 letetrs 10000 30000 er -1.133113
1 letter 30000 40000 r -0.103625
2 letters 40000 60000 ia -2.296199
1 letter 60000 70000 l -0.408171
1 letter 70000 80000 ih -1.746523
1 letter 80000 90000 s -0.432336
1 letter 90000 100000 t -0.165138
1 letter 100000 110000 ey -0.633443
2 letters 110000 130000 t -3.365733

(b) One time unit represents one letter (grapheme) in the word.


• Words sequence
…
HER
REAL
ESTATE
…
• Forced-aligned MLF
3
1 letter 0 10000 hh -0.041208
2 letetrs 10000 30000 er -1.133113
4
1 letter 30000 40000 r -0.103625
2 letters 40000 60000 ia -2.296199
1 letter 60000 70000 l -0.408171
6
1 letter 70000 80000 ih -1.746523
1 letter 80000 90000 s -0.432336
1 letter 90000 100000 t -0.165138
1 letter 100000 110000 ey -0.633443
2 letters 110000 130000 t -3.365733

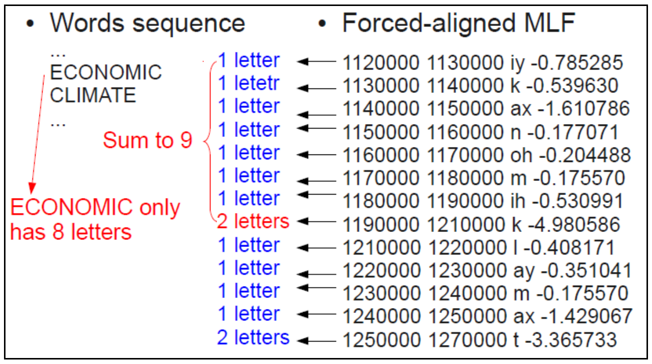
(c) When letter counts equals to the word length in the grapheme transcriptions, a boundary in the phoneme transcription is set.

Fig. 2. The process of extracting isolated-word data from forced-alignment results.

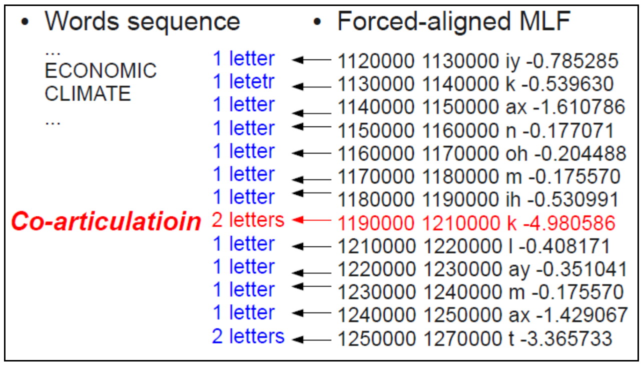
alignments between graphemes and phonemes. Based on that concern, we made a modification on the foundation of "treat-sentence-as-word" model by adding a boundary mark "_" to the end of each word in the sentence, shown as:

```
…
nice_to_meet_you_          n ay s t ax m iy t y uw
what_happens_now_          w oh t hh ae p ax n z n aw
…
```

The "big word" then becomes "nice_to_meet_you_". A special character "_" is used to mark the word boundaries in the original sentence which are equivalent to the "syllable boundaries" in the "big word". The mark "_" is still treated as a common grapheme in the "big word", and we suggest


• Words sequence
…
ECONOMIC
CLIMATE
…
Sum to 9
ECONOMIC only has 8 letters
• Forced-aligned MLF
1 letter 1120000 1130000 iy -0.785285
1 letetr 1130000 1140000 k -0.539630
1 letter 1140000 1150000 ax -1.610786
1 letter 1150000 1160000 n -0.177071
1 letter 1160000 1170000 oh -0.204488
1 letter 1170000 1180000 m -0.175570
1 letter 1180000 1190000 ih -0.530991
2 letters 1190000 1210000 k -4.980586
1 letter 1210000 1220000 l -0.408171
1 letter 1220000 1230000 ay -0.351041
1 letter 1230000 1240000 m -0.175570
1 letter 1240000 1250000 ax -1.429067
2 letters 1250000 1270000 t -3.365733

(a) In some cases, we cannot find correct letter counts for a word.


• Words sequence
…
ECONOMIC
CLIMATE
…
Co-articulatioin
• Forced-aligned MLF
1 letter 1120000 1130000 iy -0.785285
1 letetr 1130000 1140000 k -0.539630
1 letter 1140000 1150000 ax -1.610786
1 letter 1150000 1160000 n -0.177071
1 letter 1160000 1170000 oh -0.204488
1 letter 1170000 1180000 m -0.175570
1 letter 1180000 1190000 ih -0.530991
2 letters 1190000 1210000 k -4.980586
1 letter 1210000 1220000 l -0.408171
1 letter 1220000 1230000 ay -0.351041
1 letter 1230000 1240000 m -0.175570
1 letter 1240000 1250000 ax -1.429067
2 letters 1250000 1270000 t -3.365733

(b) That means a co-articulation has occurred here.

Fig. 3. The way we find co-articulations during the extracting process.

that the frequency and the position of the occurrence of the boundary mark could help the Joint-multigram Model to gain better alignments between graphemes and phonemes.

4. FORCED-ALIGNMENT FOR HMM TRAINING

An important step in building Hidden Markov Model is to train high-order N-gram phone models. However, we suggest that the widely existed cross-word n-grams in the sentence-form training data could lead to poor performance of the model, since most of those n-grams are invalid that will not occur in common words. Our experiments show that when building high-order N-gram phone models directly from sentence-form training data, the cross-word contexts caused the Kneser-Ney smoothing method failed to work; with the Witten-Bell and the Good-turing smoothing methods still able to work, the PER (Phone Error Rate) stays high and start to increase from 3-gram.

To guarantee good performance, N-gram phone model must be built on word level. We propose a solution that forced-alignment should be performed on the sentence-form training data first, and then isolated-word data can be extracted from the forced-alignment results, finally, high-order N-gram phone models can be built based on those isolated-word training data.

Fig. 2 describes the extracting process in detail. From Fig. 2 (a), we can see that, in the forced-aligned MLF (Master Label File), each phoneme has its time duration. We treat 10000 as a time unit, and we define that one time unit maps to one letter (grapheme) in the word transcription, as shown in Fig. 2 (b). Since the word boundaries in the word transcription is known, the letter counts can then be used to determine the boundaries in the phoneme transcription, as shown in Fig. 2 (c). The output from Fig. 2 (c) is:

HER	hh er
REAL	r ia l
ESTATE	ih s t ey t

We are also able to judge co-articulations in the phoneme transcription. As shown in Fig. 3 (a), the word "ECONOMIC" only has 8 letters, but the corresponding phoneme transcription generates 9 letters. If this situation happens, we define that a co-articulation has occurred, since the co-articulation caused the combination of the last time unit of the leading word and the first time unit of the following word. As shown in Fig. 3 (b), the phoneme /k/ is co-articulated. Since we do not want to keep co-articulations in our training data, the phoneme /k/ is then assigned to both "ECONOMIC" and "CLIMATE":

ECONOMIC	iy k ax n oh m ih **k**
CLIMATE	**k** l ay m ax t

One essential point needs to be clarified is that the processes above do not mean that we have involved any linguistic knowledge into the training process. The forced-alignment process is done automatically based on the prediction of the previous trained monophone model. The extraction of isolated-word data is simply based on the time alignments generated in the forced-alignment result MLF.

It is important to be aware that the forced-alignment process is not perfect. The quality of the forced-alignment results depends on the best-trained monophone model. Also, some errors will be produced during the alignment process. In addition, if there are more phonemes than graphemes in a sentence, the forced-alignment of that sentence will be failed. Moreover, the extracting algorithm is not perfect either, so when we extract isolated-word data from the forced-alignment results, additional error will be generated. Even though, we find those errors are within the acceptable range and forced-alignments can finally improve the performance.

5. EXPERIMENTS AND RESULTS

5.1. Experiments on isolated-word training data

These experiments are used to set reference PER scores for our experiments on sentence-form and audio speech training data. The training set was built based on words extracted from the WSJ data, and the testing set was built based on the 20K word list of WSJ data set.

For Joint-multigram Model, the PER drops dramatically with the increase of the N-gram order, and the drop gradually becomes stable after 4-gram. With co-articulation effects in the training data, the minimum PER produced by 9-gram model with Expectation-Maximization (EM) training algorithm is 9.64%. When the co-articulation effects are removed, the PER drops to 9.30%. From those results, we conclude that co-articulations have few draw back effects on the Joint-multigram Model. This is an important conclusion for minority language conditions, since co-articulations cannot be fixed in the sentence-form training data. By applying the Viterbi training algorithm to the training data with co-articulation effects, the minimum PER at 9-gram is 9.77%. With this acceptable error increase, we decided to use Viterbi training algorithm for the sentence-form training data to guarantee efficiency.

For HMM training method, 4-emittting-state HMM was used to build the monophone model, 10 iterations were performed. High-order N-gram phone model was smoothed using Kneser-Ney algorithm. Context-dependent triphone model was built using the clustering method proposed by Taylor [5], so that all the triphones that had more than 20 tokens in the training data were clustered. Together with the contribution of the 4-gram phone model and the context-dependent triphone model, the system achieved a minimum PER of 21.19%.

The difference of the minimum PER between the two models is significant. Published results [2, 3] also show that the Joint-multigram Model is excellent for isolated-word training data. However, Bisani and Ney have also mentioned in their paper [2, 3] that the Joint-multigram Model training method was very time consuming. Our experiments show significant benefit for Hidden Markov Model method of its extremely efficient training process. It took about 30 hours to train the 9-gram model for Joint-multigram Model, but it took only a few minutes for Hidden Markov Model to finish the whole training process. Considering the complexity context effects in the sentence-form training data and unpredicted errors in the audio speech training data, we will keep trying on Hidden Markov Model to see whether it could bring better results in more challenging situations.

5.2. Experiments on sentence-form training data

Step by step, to simulate the condition for minority language, we change the training data into sentence-form, but still using their text format. The "treat-sentence-as-word" training method was applied to Joint-multigram Model training, and the "forced-alignment" was involved in HMM training. Training set was built based on the WSJ data set, and testing set was built based on the 20K word list of WSJ data set.

For Joint-multigram Model, we have tried experiments of training the model with and without the boundary mark "_". When boundary marks are involved in the training data, same boundary marks need to be added to the end of each grapheme sequence in the testing data to keep consistency.

Results show that without boundary mark the PER at 9-gram is 10.69%, and by adding the boundary mark the PER at 9-gram drops to 10.17%. Compared with the result we get from isolated-word training data under the condition of co-articulation plus Viterbi training algorithm 9.77%, we conclude that by applying the "treat-sentence-as-word" training method to sentence-from training data, the model can achieve very close performance as that of the isolated-word experiment.

For HMM training method, we applied "forced-alignment" while training high-order N-gram phone models. Results show that the minimum PER is 20.86%. Compared with the result we get from isolated-word training data 21.19%, we conclude that by involving the "forced-alignment" process in sentence-form data training, the model can achieve as good performance as that of the isolated-word experiment, and even better.

The PER difference between two models is still very significant. However, the training process has become a lot harder for the Joint-multigram Model since the length of the "big word" significantly increased the computational complexity in finding alignments. Even by using Viterbi algorithm, it took more than 60 hours to train a 9-gram model. On the other hand, the advantage of Hidden Markov Model becomes dramatically obvious. The whole training process only took about 20 to 30 minutes. For both models, the results get from the sentence-form training data using our proposed methods are as good as those from the isolated-word training data.

5.3. Experiments on speech training data

These experiments simulated the minority language conditions we proposed at the beginning. We adopted an English phone-based speech recogniser from Lincoln [4] to generate the phoneme transcriptions, and we used those transcriptions to train our two baseline systems. The speech recogniser was based on HTK Toolkit and the acoustic model was a context-dependent triphone model. As shown in Fig. 1, WSJ data was used as the input to the speech recogniser, the outputted phoneme transcriptions together with the grapheme input formed the training data for our baseline systems, and the 20K word list was used as the testing data. Phone set differences between the adopted acoustic model and the WSJ data set were fixed before training our baselines. As we have expected, the phoneme transcriptions get from the speech recogniser contains large number of unexpected errors with a PER of 38.30%.

For Joint-multigram Model, boundary marks were inserted into both the training and testing data, and Viterbi algorithm was used to train the model. However, due to the large amount of errors in the training data, the alignment process became much harder and longer. Within our hardware limitation, we only managed to train the model up to 4-gram.

Table 1. The results gained using the recognised phoneme transcriptions as training data — Joint-multigram Model.

Order of the N-gram model	PER
1-gram	43.04%
2-gram	41.98%
3-gram	41.78%
4-gram	44.51%

From Table 1, we find that the PER increases from 4-gram instead of going to convergence. Based on our analysis, since the model was trained on a flat data with large number of errors, with the increase of N-gram order, high probabilities were more constantly assigned to wrong data as more context information been considered.

For Hidden Markov Model, we used 4-emitting-state monophone model as the foundation monophone model, then performed forced-alignment and built high-order N-gram phone models, finally context-dependent model was built trying to further improve the performance.

Table 2. The results gained using the recognised phoneme transcriptions as training data — Hidden Markov Model. Context-dependent model produce worse results because of large number of wrong context information in the training data.

Order of the N-gram model	4-state monophone model, 10 iterations (PER)	Context-dependent triphone model (PER)
2-gram	40.15%	43.56%
3-gram	33.32%	34.78%
4-gram	33.01%	34.44%
5-gram	33.26%	34.79%

As mentioned in section 4, while performing forced-alignment, if there are more phonemes than graphemes in a sentence, the forced-alignment of that sentence will be failed. As a result, by automatically dropping those failed sentences, most data with badly errors were dropped out from the training data. In other words, when we used the extracted isolated-word data to build N-gram phone models, lots of badly errors were already dropped during the forced-alignment step. From the results shown in Table 2, we can see significant performance improvements from 2-gram to 3-gram. We suggest that the PER increase at 5-gram is caused by data sparse. However, not as we expected, the context-dependent model produce worse results than the monophone model. Considering the fact that there are lots of errors existed in the training data, we suggest that the worse results are caused by the effects of wrong context information.

The Joint-multigram model achieved a minimum PER of 41.78%, while Hidden Markov Model achieved 33.01%. This time, the Hidden Markov model produced better results. The errors in the training data brought very high computational cost to the Joint-multigram Model and caused the model to give high probabilities to lots of wrong data. As a consequence, it took the Joint-mutigram Model about 100 hours to train a 4-gram model. On the other hand, with the help of high-order N-gram models, the Hidden Markov Model achieved better results with only about 30 minutes training time. The forced-alignment step contributes a lot by dropping lots of data with badly errors. We conclude that the Joint-multigram Model has weaker capacity for large number of errors in the training data, and that makes it hard to deal with audio speech training data, while Hidden Markov Model can handle that kind of training data better.

6. DISCUSSIONS AND CONCLUSIONS

Table 3 gives a summary of our main results. As shown by the results, the Joint-multigram Model can produce much better performance than the Hidden Markov Model for both isolated-word training data and sentence-form training data. But the Joint-multigram Model suffers badly from its high computational complexity, and the Hidden Markov Model is proved to have significant efficiency advantages.

For both models, the results get from the sentence-form training data using our proposed methods are as good as the results get from the isolated-word training data. We conclude that by applying "treat-sentence-as-word" training method and by involving the forced-alignment process, the Joint-multigram Model and the Hidden Markov Model can be successfully extended to sentence-form data respectively.

For the experiments using audio speech training data, the forced-alignment process can help the Hidden Markov Model to drop lots of poor data and produce better results. We find that the Joint-multigram Model has weaker capability to the training data with large number of errors and long grapheme-phoneme sequences.

Based on the results above, we suggest that the Joint-multigram Model is better for those languages with high quality training data, while the Hidden Markov Model is more proper for common minority language issue. The performance of the Hidden Markov Model can be improved further by developing better clustering strategies for context-dependent triphone models. We would like to explore that in our further studies.

Table 3. Summary of the main results of our experiments.

Experiment / **Model**	**Isolated-word**	**Sentence-form**	**Audio speech**
Joint-multigram	9.77%	10.17%	41.78%
HMM	21.19%	20.86%	33.01%

7. ACKNOWLEDGEMENTS

This research was funded by the NSFC Project with No. 60975081, and the CASS Innovation Program. Thanks to the supports from the school of PPLS and the school of Informatics of the University of Edinburgh.

8. REFERENCES

[1] J. Fransen, D. Pye, T. Robison, P. Woodland, and S. Young, "WSJCAM0 Corpus and Recording Description", Cambridge, UK, 1994.

[2] M. Bisani and H. Ney, "Investigations on joint-multigram models for grapheme-to-phoneme conversion", *Proc. Internat. Conf. on Spoken Language Processing, Vol. 1, pp. 105–108*, Denver, CO, USA, 2002.

[3] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion", *Speech Communication 50 (2008) 434-451*, 2008.

[4] M. Lincoln, "Speech Separation Challenge 2: baseline recogniser and scoring scripts", *Pascal Speech Separation Challenge Part II*, 2006.

[5] P. Taylor, "Hidden Markov Models for Grapheme to Phoneme Conversion", *Proc. Interspeech 2005*, Lisbon, Portugal, 2005

[6] R. I. Damper, Y. Marchand, M. J. Adamson and K. Gustafson, "A comparison of letter-to-sound conversion techniques for English text-to-speech synthesis", *Proc. Institute of Acoustics 20 (6)*, 1999.

[7] S.F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion", *Proc. European Conf. on Speech Communication and Technology, pp. 2033–2036*, Geneva, Switzerland, 2003

[8] S.F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling", *Computer Speech Lang. 13 (4), 359–394, Oct*, 1999.

[This paper was published on Oriental COCOSDA, 2012]