Procesamiento del Lenguaje Natural, núm. 35 (2005), pp. 29-34 recibido 02-05-2005; aceptado 01-06-2005

Main issues in grapheme-to-phoneme conversion for TTS

Tatyana Polyakova
Universitat Politècnica de Catalunya
Jordi Girona 1-3, Barcelona, Spain
tatyana@gps.tsc.upc.es

Antonio Bonafonte
Universitat Politècnica de Catalunya
Jordi Girona 1-3, Barcelona, Spain
antonio.bonafonte@upc.es

Resumen: La conversión de letras a fonemas en inglés está siendo desarrollada para su futura integración en un sistema de síntesis de habla dentro del proyecto TC-STAR. En este trabajo se describen los experimentos realizados usando dos técnicas diferentes de aprendizaje automático. Se ha considerado la predicción de la pronunciación con y sin acento. Se analiza la influencia de los diferentes parámetros en la tasa de error en la conversión de letras a fonemas. También se estudia la distribución de la tasa de error en función de la longitud de las palabras ha sido obtenida.

Palabras clave: letras en fonemas, alineado, x-grams, traductores de estados finitos, CART.

Abstract: Grapheme-to-phoneme conversion system for English is being developed for further integration into speech synthesis system within TC-STAR project. In this work we describe experiments performed using two different machine learning techniques. The pronunciation was predicted both for stressed and unstressed lexicon and the results were compared. Analysis of different parameters that may influence the error rate in grapheme-to-phoneme conversion was performed. The error rate as a function of the word length was studied.

Keywords: grapheme-to-phoneme, alignment, x-grams, finite-state transducers, CART.

1 *Introduction*

Text-to-speech synthesis and continuous speech recognition are the two rapidly developing technologies that have many applications in many different areas of human life. It is impossible to have a good-working text-to-speech system without having a tool that generates correct pronunciation from the orthographic transcription.

Because of the writing system irregularities (absence of unambiguity between the phoneme and grapheme full sets, where, for example one letter may correspond to two phonemes or to no sound at all) the grapheme-to-phoneme conversion is not an easy task, especially for languages like English or French where the relation between letters and sounds is not very clear.

To train a grapheme-to-phoneme conversion system most algorithms require an alignment between grapheme and phoneme strings. Many different approaches have been elaborated to automatically perform grapheme-to-phoneme alignment for the translation systems; however, in grapheme-to-phoneme conversion usually a dictionary aligned by hand-seeded rules is used to train a classifier.

Different methods for building automatic alignments have been proposed, among which there are one-to-one and many-to-many alignments. In one-to-one alignment each letter corresponds only to one phoneme and vice versa. To match grapheme and phoneme strings, an “empty” symbol is introduced into either string.

In many-to-many alignments a letter can correspond to more than one phoneme and a phoneme can correspond to more than one letter.

An example of one-to-one alignment model is the epsilon scattering model (ESM) where “epsilon” or “empty” phoneme is introduced to either both grapheme and phoneme strings or only to phoneme string (Pagel, Lenzo and Black, 1998) Then the EM algorithm is applied to optimize their positions.

An example of many-to-many alignment was proposed by Bisani and Ney (2003). In their model the main idea is that the both orthographic form and pronunciation for each word are determined by a common sequence of *graphones* which is a pair $q = (\boldsymbol{g}, \boldsymbol{\varphi})$ of letter and phonemes sequences, respectively (where $q \in Q \subseteq G^* \times \Phi^*$; and G is the letter alphabet and Φ is the inventory of phonemic symbols). For a given sequence of letters $\boldsymbol{g} \in G^*$ the

ISSN: 1135-5948

probability to find most likely phoneme sequence $\boldsymbol{\varphi} \in \Phi^*$ is maximized.

This paper reports on a data driven-approach to pronunciation modeling. To align graphemes and phonemes we chose to use automatic epsilon-scattering. The experiments were performed on stressed and unstressed lexicon of English, the grapheme-to-phoneme results, obtained by CART decision trees and x-gram based Finite-State Transducers were compared for these lexicons for both cases.

2 Grapheme-to-phoneme methods

2.1 Epsilon-scattering based alignment

As the alignment method we use automatic epsilon-scattering method to produce one -to-one alignment. (Black, Lenzo and Pagel, 1998). For the cases when the number of letters is greater than that of phonemes we introduce, the "empty" phoneme symbol into phonetic representations to match the length of grapheme and phoneme strings. First we add the necessary number of "empty" symbols, into all possible positions in phonetic representations for each word in the training corpus. In the Table 1 we give an example of two of the possible alignment candidates for the word "meadows".

Let.	M	E	A	D	O	W	S
Phon.	m	E	_	d	o	_	z
Phon.	_	_	m	E	d	o	Z

Table 1: Some possible alignment candidates.

Such a probabilistic initialization gives us number of all possible imperfect alignments. Our goal is to maximize the probability that letter g matches phoneme φ and therefore choose the best alignment from the possible candidates. It is done by applying the expectation maximization algorithm (EM) (Dempster, Laird and Rubin, 1977). The EM associated with joint grapheme/phoneme probabilities. Under certain circumstances the EM guarantees an increase of the likelihood function at each iteration until convergence to a local maximum

2.1.1 Stress assignment

In this work we have used the phonetic alphabet that is called SAMPA (http://www.phon.ucl.ac.uk/home/sampa/american.htm).

The total number of the phoneme symbols involved in the SAMPA for American English is N_p=41. As we use different phonemes for accented and unaccented vowels, to perform training and test on the accented dictionary we have added N_a=17, accented vowel symbols and an "empty" phoneme, "_". As an example, in Table 2 the word "abilities" is shown together with its two phonetic representations: with stress and without stress. In the stressed version the digit "1" is added to identify the stressed vowel. In this example one "empty" symbol was introduced to match the strings.

Let.	A	B	I	L	I	T	I	E	S
Str. Phon.	@	B	I1	*l*	@	t	i	_	z
Unstr. Phon.	@	B	I	*l*	@	t	i	_	z

Table 2: Alignment of the graphemes and phonemes including the "empty" symbol. The digit 1 is the stress marker.

2.2 CART

Deriving the pronunciation automatically by using decision trees, such as Classification and Regression Trees, or CART is a commonly used technique in grapheme-to-phoneme conversion. Pagel, Lenzo and Black (1998) have introduced the CART decision-trees into pronunciation prediction. As the input vectors, the graphemic sliding window, containing three letters on the left and three letters on the right of each center letter was considered. The advantage of using CART method is that it produces compact models. The model size is defined by the total number of questions and leaf nodes in the generated tree.

2.3 Finite State Transducers

In this next approach we chose the pronunciation that maximizes

$$\arg\max_{\varphi} \left\{ p(\varphi / g) \right\} \qquad (1),$$

where φ is the sequence of phonemes, including the "empty" phoneme, and g is the sequence of letters.

To solve the maximization problem we use a Finite state transducer, which is similar to (Galescu and Allen, 2001). The equation (1) can be expressed as

$$\arg\max_{\varphi} \{p(\varphi, g)\} / p(g) = \arg\max_{\varphi} \{p(\varphi, g)\} \ (2).$$

This can be estimated, using standard n-gram methods.

In the training, first the alignment is performed. Then we define the graphones γ, as the pair consisting of one letter and one phoneme (including ε)

$$p(g,\varphi)=\prod_{i=1}^{N} p\left(g_i,\varphi_i / g_1^{i-1},\varphi_1^{i-1}\right)=\prod_{i=1}^{N} p(\gamma_i / \gamma_1^{i-1}), \quad (3)$$

where N is the number of letters in the word. This can be estimated using n-grams, but the symbols are not words, but the "graphones", i.e. pair (letter, phoneme) found in the aligned dictionary.

N-grams can be represented using a Finite-state automaton: for each history h we define a state and for each new graphone $\gamma_l < g_l,\varphi_l >$ we create an edge, with the label $< g_l,\varphi_l >$ and weight it with the probability $p(\gamma_l / h)$.

For example, for the word "aligned" we create the following graphone sequence:

<start> (A,a) (L,l) (I,aI) (G,_) (N,n) (E,_) (D,d) <end>

Fig. 1 shows some states and edges for the bigram-based FSA (in practice all of the training data is used to estimate probability).

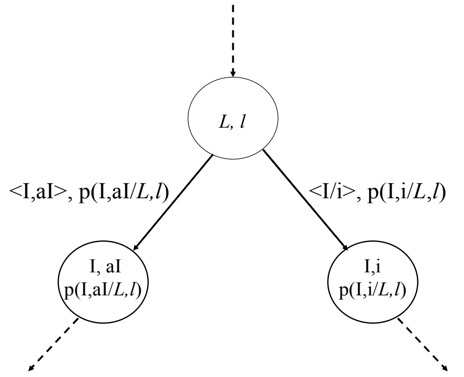

Figure 1: A Finite-state automaton

The FSA allows to compute the probability p(γ), but this requires an alignment. In order to solve the equation (2) we derive a finite state transducer in a straightforward way: the labels attached to the FSA are split: the letter becomes an input and the phoneme becomes an output. Fig. 2 shows the results of this after applying it to Fig. 1.

Note that the FST is not deterministic. For instance, from the state labeled as (*L,l*) there are two possible edges for the same input letter "I".

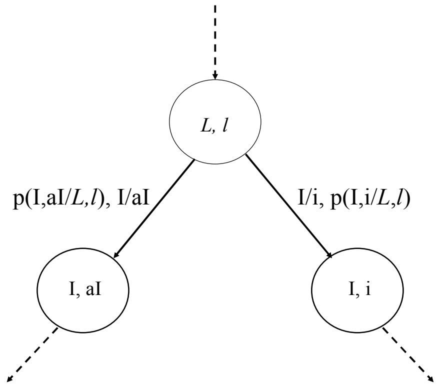

Figure 2: A Finite-state transducer

To find the pronunciation we have to solve equation (3). This is equivalent to finding the path of the FST with maximum probabilities. The input letters limit the number of edges which can be followed. The best path is found using the dynamic programming algorithm.

In order to derive correct pronunciation we use the flexible length *x*-gram model (Bonafonte and Mariño, 1996). The *x*-gram model assumes that the number of conditioning grapheme-phoneme pairs depends on each particular case. The main idea of the *x*-gram model lies in applying of a merging-state algorithm which allows us to reduce the number of states. The two criteria for merging were used. Firstly, the merging is applied if the number of times of a given history $<q_{i-m},,,,q_i>$ appearance on the training data with lexicon size *J*, is less than a threshold, k_{min} (where q_i is a grapheme-phoneme pair). The probability to such grapheme-phoneme pairs is derived by smoothing. Then the states are merged if their distributions $\mathbf{p}=p(q/q_{i-m},\ldots q_i)$ and $\mathbf{m}=p(q/q_{i-m+1},\ldots q_i)$ have a small enough difference in the information. The measure of the difference is the divergence *D* defined as

$$D(p \,//\, m)=\sum_{j=1}^{J} p(i)\log\left(\frac{p(j)}{m(j)}\right) \quad (4)$$

Choosing the proper values of the thresholds k_{min} and D , one can significantly reduce the number of states without decreasing of model goodness.

3 Experimental results

LC-STAR (www.lc-star.com) has created dictionaries in 13 languages. In this work we have used the LC-STAR dictionary of American English, kindly provided by NSC (natural Speech Communications) for the development of speech-to-speech translation demonstrator within LC-STAR project.

We have performed experiments using the epsilon scattering method to align the lexicon combined with CART and FST to predict the correct pronunciation

First, the system was trained and tested using the CART tree-based phoneme prediction. The LC-STAR dictionary contains 50 thousand words; it was randomly split into ten equal parts. Ninety per cent of the dictionary was used to train the system, ten percent of which was used to perform the evaluation; other ten per cent were used to test the system.

3.1. Conversion using CART and influence of the decision tree parameters on the error rate

For different parameters of CART decision tree the phoneme and word error rate have been plotted in Fig. 3a and Fig. 3b. Figure 3a shows the error rate as the function of the value of minimum amount of the entropy gain, necessary to justify the further tree expansion. The Fig. 3b shows these error rates for different values of maximum tree depth. If we set this parameter to be very small, the word error rate tends to be very high.

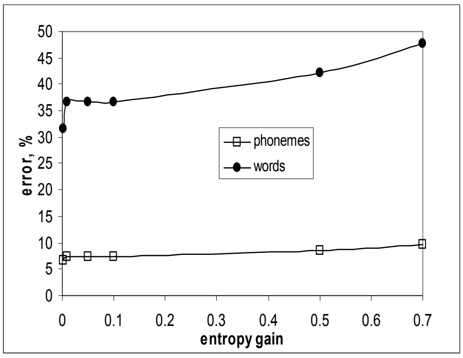

Figure 3a: Phoneme and word error rates (%) as a function of minimum entropy gain needed for further expansion of the tree.

The parameters that give us the best results were found to be 0.001 for the entropy gain and 7 for the tree depth. These values coincide with those obtained by Black, Lenzo and Pagel (1998).

Below we present the results obtained for the stressed and unstressed dictionary (see Table 3). The data presented in the first row were obtained for the phonetic transcription including the stress marks; a misplaced stress mark was counted as an error. The second row shows the percentage of correct phonemes, and words in comparing with the correct phonetic transcription for the unstressed lexicon.

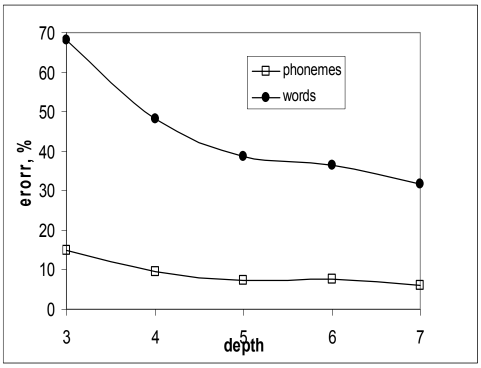

Figure 3b: Phoneme and word error rates (%) as a function of the maximum tree depth.

	Phon.	Words
with stress	90.13	51.04
w/o stress	93.02	65.48

Table 3: Percentage of correct phonemes, and words for stressed and unstressed dictionary using CART.

The results can be improved by removing the ambiguous abbreviations and non-standard words, see Table 4.

	Phon.	Words
with stress	91.29	57.8
w/o stress	93.93	68.32

Table 4: Percentage of correct phonemes, and words for stressed and unstressed dictionary using CART, after the removal of long non-standard words and abbreviations from the corpus.

As we can see from the Tables 3 and 4, the system performs significantly better on the unstressed lexicon, especially if we take into the consideration the percentage of the words correct obtained from the experiment.

3.2 Grapheme-to-phoneme conversion using FST

It is important to analyze how the parameters of the *x*-gram models may affect on the error rates. In order to clarify this, we performed experiments for different values of the parameters *n* and *D* in *x*-gram model, *n* is the maximum possible length conditional probability history.

The Fig. 4a shows error rate percentage as the function of the *n*.

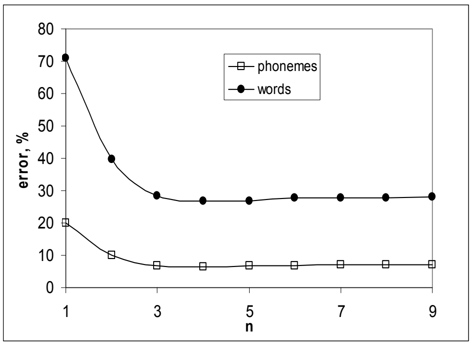

Figure 4a: Phoneme and word error rates (%) as a function of *n*.

Both error rates monotonically decrease as the *n* increases.

This implies that *n*=5 is the optimal parameter of the *n*-gram model for the given corpus. In Fig. 4b we plotted the error rate as a function of the divergence threshold in the *x*-gram model.

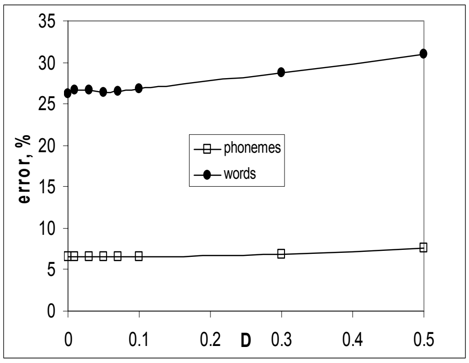

Figure 4b: Phoneme and word error rates (%) as a function of the divergence threshold *D*.

The results presented in Fig. 4 imply that allowing a significant decrease of the states number the *x*-gram model gives good results even for high enough value of the divergence threshold. The results presented in Fig. 4a and 4b give the total percentage of errors in the grapheme-to-phoneme conversions. The error rates practically do not change in the range of the parameters: *n*≥5 and 0≤*D*<0.2. Thus, in the model frameworks there is no possibility to decrease the error rate by increasing *n* or decreasing *D*.

The best parameters for this dictionary were chosen to be n=5 and D=0.01.

The results obtained by FST are given in Table 5. From Table 5 one can see that the results obtained for the unstressed lexicon are significantly higher as we have seen before for CART. Our goal is to predict phonemes and stress together as it is very important for speech synthesis, while for recognition it may be sufficient to predict only phonetic transcription.

	Phon.	Subst.	Ins.	Del.	Word
with stress	91.07	5.21	0.94	2.79	67.91
w/o stress	93.63	3.30	0.51	2.15	75.66

Table 5: Percentage of correct phonemes, insertions, deletions and substitutions for stressed and unstressed dictionary using FST, after the removal of long non-standard words and abbreviations from the corpus.

4 Error rate versus word length

It is interesting to know the error distribution versus the word length. For experimental data for the dictionary "without stress" (the second row of Table 2) we plotted the probability distribution function of errors, $f(l)=N_{er}(l)/N_0$ as a function of the grapheme number per word, (where $N_{er}(l)$ is the number of errors in *l*-lettered words, and $N_0=\sum N_{er}(l)$ is the total number of errors in the given experiment). This plot is shown in Fig. 5 (open squares).

As it is seen from Fig. 5 the erroneous pronunciation is most likely generated for words consisting of 9 letters. However, one should keep in mind that in English the word frequency is a non-monotonic function of the word length. It was recently shown that in English the word frequency obeys the distribution e.g. see (Sigurd, Eeg-Olofsson, and van de Weijer, 2004) with the maximum at *l*=3. In Fig. 5 this distribution is shown by the circle-marked line. Another word distribution by the length for the British English Pronunciation dictionary was taken from (Damper and Marchand, 2005). There, the average word length is 8.87 letters with a standard deviation of 2.58 letters. In Fig. 5 these distributions are shown by the lines with filled symbols. The difference between two distributions is that in Sigurd et al, the distribution is given for running words, while Damper and Marchand, (2005) have counted the distribution for the BEEP dictionary, after deleting the words with length three and under. In the light of these distributions one can expect that the probability of conversion errors in practice should be lower, as the long words in spoken language are rarer than in the dictionary.

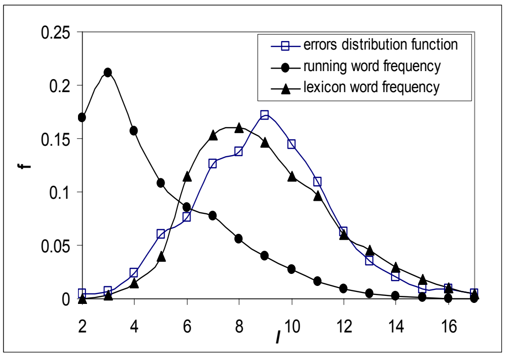

Figure 5: Probability distribution function of errors versus the grapheme number per word (open squares). The line with filled circles represents the distribution for running words; and the line with filled triangles is the distribution for words in a lexicon in English.

5 Conclusions

The performed experiments show that grapheme-to-phoneme results depend on many parameters as well as on the choice of machine-learning technique used to perform conversion. The results were obtained for both accented and unaccented lexicon, using epsilon-scattering method to perform alignment and combining it with CART and FST classifiers.

The FST performed better on both stressed and unstressed dictionary. We have analyzed different factors that may influence errors rate in grapheme-to-phoneme conversion. The obtained error distribution (see Fig. 5) indicates that 9-letter long words mainly contribute to the total error rate if the optimal model parameters are chosen for training of the system. An optimal grapheme-to-phoneme alignment method has the same importance as the classification method selection and it could give a significant performance improvement in the tasks of speech synthesis and speech recognition.

The problem of the currently used one-to-one alignment and inserting of the “empty” symbols is that some alignments produced are completely artificial and do not specify the pronunciation, moreover the reasonable alignment may not be unique (Damper and Marchand, (2005). Black, Lenzo and Pagel (1998) have concluded that the choice of the decision-tree learning technique does not influence on the results as much as the alignment algorithm, used to align the lexicon.

In the future we are planning to work to compare different alignment methods

6 Acknowledgments

This work has been funded by the European Union under the integrated project TC-STAR - Technology and Corpora for Speech to Speech Translation (IST-2002-FP6-506738, http://www.tc-star.org).

7 References

Bisani M. and H. Ney. 2002. Investigations on joint-multigram model for grapheme-to-phoneme conversion. In *Proc. of the 7^th^ Int. Conf. on Spoken Language Processing*, Denver, CO, vol.1, 105-108.

Black A., K. Lenzo K. and V. Pagel. 1998. Issues in building general letter to sound rules. *In Proc. of the 3rd ESCA workshop on speech synthesis.,* 77-80, Jenolah Caves, Australia

Bonafonte A. and J. B. Mariño, 1996 “Language modeling using X-grams” *Proc of ICSLP-96*,Vol.1, 394-397, Philadelphia, October 1996.

Damper R. I., Y. Marchand , J-D. Marsterns. and A. Bazin. 2004. “Aligning letters and phonemes for speech synthesis” in *Proc. of the 5^th^ ISCA Speech Syntesis Workshop*, Pittsburgh, 209-214.

Damper R. I., Y. Marchand. 2005. Information fusion approaches to the automatic pronunciation of print by analogy. *Information Fusion.* To appear.

Dempster A. P., N. M. Laird and D.B. Rubin . 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Ser. B, 39, 1-38.

Galescu L., J. Allen., 2001 "Bi-directional Conversion Between Graphemes and Phonemes Using a Joint N-gram Model", in *Proc. of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis,* Perthshire, Scotland, 2001

Pagel V., K. Lenzo, A.Black. 1998. Letter-to-sound rules for accented lexicon compression. In *Proc. of ICSLP98*, vol 5, 2015-2020, Sydney, Australia

Sigurd B., M. Eeg-Olofsson., J. van de Weijer. 2004. Word length, sentence length and frequency zipf-revisited. *Studia Linguistica* 58(1), 37-52.