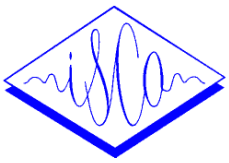

3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

ON THE DEVELOPMENT OF PRONUNCIATION RULES FOR TEXT-TO-SPEECH SYNTHESIS

Bert Van Coile

Lernout & Hauspie Speech Products, Ieper, Belgium
University of Gent, Gent, Belgium

ABSTRACT

This paper describes several aspects of the development of grapheme-to-phoneme systems for 6 different languages. We report on the lexical data bases we are using for these developments. Our method for the automatic alignment of orthographic and phonetic transcriptions is reviewed. We describe how the method can be used to facilitate the development of pronunciation rules. Performances are given for word data and name data. In the last part of the paper we report on some orthographic/phonetic properties of the different languages using the concepts of entropy and mutual information.

keywords: text-to-speech, lexical data bases, grapheme-to-phoneme conversion

10.21437/Eurospeech.1993-331

1. INTRODUCTION

One of the important aspects in text-to-speech synthesis is the grapheme-to-phoneme conversion of the input text. Although there are different approaches to this conversion, this contribution focuses on the use of context dependent pronunciation rules.

Previously, we reported on a rule development environment called *Depes* [8]. This software tool was initially intended to facilitate the manual rule development. It offered the user a powerful knowledge representation language and provided several development tools such as a rule compiler, a linker and a debugger. Later on, in order to facilitate and speed up our multi-lingual text-to-speech developments, the *Depes* environment was extended with tools for the automatic alignment of orthographic and phonetic representations and with tools for automatic rule induction [9,10] . In the past 2 years, the environment has been modified and extended based on our experience with the development for 6 different languages (French, Dutch, German, Spanish, English and Korean). Our rule development tools and methodology have also been used for the development of pronunciation knowledge for Dutch, French and German names.

We will report on the methodology and some of the tools we are using to create pronunciation rule grammars. We will also describe some aspects of the lexical data bases we are using for our text-to-speech developments.

2. THE LEXICAL DATA BASES

We currently have word data bases for each of the six languages we worked on and a name data base with more than 500,000 Dutch, French and German entries.

2.1. THE WORD DATA BASES

Currently, our lexical data bases contain between 20,000 and 60,000 types depending on the language. Each data base contains at least orthographic and phonetic representations. Frequency information is available for most of the entries. In some data bases, part-of-speech information is quite detailed and available for almost all entries while in other data bases (such as our Korean one) the part-of-speech data is limited.

Our word data bases were compiled from other lexical data bases and from electronic as well as printed dictionaries (including phonetic and frequency dictionaries). Also a large amount of computer readable texts (newspapers, letters, etc.) was processed. To create for example our Spanish lexical data base, we merged the content of a preliminary version of a new Spanish dictionary with the content of a published frequency dictionary [3]. The dictionary was obtained in computer readable form as part of a cooperation agreement with the publisher. The information from the frequency dictionary was entered into our lexical data base manually. Phonetic transcriptions were added during our rule development, following a semi-automatic bootstrapping procedure. A rule grammar was used to transcribe a set of data automatically. The output was manually corrected by the native Spanish member of our text-to-speech team and then used for improving the rules. The new rule set was used to create transcriptions for additional data. The last two steps were repeated several times. For most of our lexical data bases, the phonetic transcriptions were added by us, using the same iterative bootstrapping procedure.

When we started our Korean (Hangul) text-to-speech project, we did not succeed in obtaining computer readable dictionaries. Therefore, we started the development of a Korean lexical data base from scratch. A text corpus has been composed. It contains a.o. Korean textbooks, business correspondence, industrial texts, scientific reports, etc. The current version of the corpus contains only 300,000 tokens. Additional data will be added to increase the size of the corpus drastically. Word lists and frequency information have been extracted from the corpus. Phonetic transcriptions have been added using the same procedure as already described.

The Korean character set consists of approximately 2400 characters (excluding the Chinese characters). Each such character can be seen as an orthographic representation of one syllable. This concept and the fact that Korean characters are represented by 2 bytes, make it difficult to process Korean words with the same tools and methodology that were used for European languages. In order to overcome this problem we defined a transliteration system that allows us to convert a Korean orthographic representation into a latin representation. In contrast to the so called Romanisation systems described in

literature (see [5]) our transliteration method is not designed to mimic Korean pronunciation but solely to create an equivalent latin representation. It is based on the fact that each Korean character is composed of 2, 3 or 4 elementary symbols or *letters*. In total 24 different *letters* are used: 14 consonants and 10 vowels. A translation table was defined that maps each Korean letter on a latin one. In order to maintain the Korean character structure, hyphens are inserted in the transliterated representation. Table 1 exemplifies the transliteration principle. In the remaining of this article, when referring to Korean, we will use the term *orthographic* representation as a synonym for the *latin transliterated* representation.

Table 1: Transliteration method

ㅈ	C
ㅏ	a
ㅇ	N
ㄹ	l
ㅣ	i
...	...

장래 → CaN-lai

2.2. THE NAME DATA BASES

The data base we have used is derived from the telephone directory of the Belgian Telecom Company and consists of about 3.5 million subscriber records. Every record contains the subscriber name, the street name and the place name. Accordingly, we constructed a subscriber name data base, a street data base and a place data base. Due to the format of the original data base, it was impossible to distinguish between first names, surnames and business names. Hence, these different categories are all comprised within the subscriber name data base, henceforth referred to as the *name data base*. In the original Telecom data, every record is assigned a language code (Dutch, French or German). This reflects the linguistic situation in Belgium. The official language in the northern half of the country (Flanders) is Dutch, while the official language in the southern half (Wallonia) is French. The very south-eastern corner of the country houses a German speaking minority. German is the official language in that area. Brussels, being the capital, is officially bilingual (French, Dutch). Also, in some municipalities at the language borders, two languages have an official status.

For reasons of efficiency and ease of handling, the name and street data bases were conceived as data bases of *types*. They contain not only orthographic representations but also phonetic transcriptions and etymological tags. Table 2 shows some statistics for the Dutch, French and German part of our name type data base. Other characteristics of our name data bases can be found in [11].

Table 2: Name types

	Dutch	French	German
entries	202531	225314	11110
entries for 50% cum. freq.	1208	2428	468
unique entries (freq=1)	52.8%	47.7%	60.0%

In the remainder of this article we will often refer to training and test data that were extracted from our word and name data bases. For each of the databases (except the German one), we used the 17,000 most frequent words or names. The words and names were selected on the basis of frequency information published in [1,2,3,6,7]. This information is also available in our lexical data bases. For German only 9,000 words were used (The words for which we have frequency information), and all available names.

3. THE AUTOMATIC ALIGNMENT

3.1. WHY?

Our grapheme-to-phoneme systems are developed within the framework of the *Depes* development environment. The use of synchronized data layers within *Depes* [8] enables us to keep track of the correspondence between graphemes and phonemes. The lexical data bases described above can serve as training and testing data for the development of grapheme-to-phoneme conversion rules. For each orthographic word the system can generate a phonetic transcription and an alignment between graphemes and phonemes. Suppose a reference alignment between the orthographic representation and the target phonetic transcription is available for each training and test word. Then, it is of course very easy to create a common alignment between the orthographic writing, the phonetic transcription generated by rule and the reference phonetic transcription. Once one has a common alignment, one can easily trace mistakes on phoneme or letter level. The researcher is given suggestions by the development program how to write (very specific) rules to solve pronunciation problems. Suppose the grapheme to phoneme conversion of the Dutch word *meisje* leads to the following alignment:

orthographic	m	ei	sj	e
phonetic (from data base)	m	J	S	$
phonetic (by rule)	m	J	sj	$

The vertical lines represent the phoneme boundaries for the phonemes of the *correct* phonetic transcription.

The alignment clearly shows an error in the *S* phoneme, which is incorrectly pronounced by the rule system as two phonemes. The program generates the following very specific rule:

```
|sj| ---> |S| /#mei_e#;        (1)
```

Let us have a look at the following example for the English word *computer*:

orthographic	c	o	m	p		u	t	e	r
phonetic	k	$	m	p	j	u	t	$	r
phonetic (by rule)	k	$	m	p		u	t	$	r

Here as well, only one phoneme error occurs, namely: the rules do not generate the *j* phoneme. The program then suggests the following specific insertion rule:

```
|| ---> |j| /#comp_uter#;        (2)
```

Besides performance statistics at the word level, the development program also creates statistics about phonemie and/or grapheme errors. These statistics are interesting to decide on which pronunciation problems we should focus. On the basis of a list with very specific rules such as (1) and (2), the researcher can create more general rules improving the performance of the system.

Reference alignments between orthographic representations and phonetic transcriptions are also needed for the automatic learning facility of *Depes* [9,10]. The learning procedure can only generate context dependent pronunciation rules if the correspondence between the graphemes and the phonemes of the training words is known.

In the following paragraphs we will review and evaluate our automatic alignment method.

3.2. PRINCIPLE

Previously, we already reported on an automatic alignment method based on the technique of Hidden Markov models. A similar approach is also mentioned in [4].

The previous version of our alignment method used the concept of HM phoneme models. The output symbols of the models were letters (graphemes). To align the phonemic transcription of a word with its orthographic representation, a HM word model was constructed by concatenating phoneme models according to the given phonemic transcription. The alignment was then determined using the well-known Viterbi algorithm.

The new version of the alignment program uses the same concept. However the phoneme models are much simpler. When a phoneme model is entered, one grapheme string is emitted. The set of possible output strings contains all orthographic strings up to a specific length including the empty string. The probability that string S is emitted by phoneme model P is represented by P(S|P). The model for phoneme P is completely characterised if the conditional probabilities P(S|P) are known for all strings S. Conceptually, the phoneme models can be seen as 1-state HM models that use orthographic strings as output tokens.

Also in this new version of our algorithm, the alignment between the orthographic and the phonemic transcription is determined by means of the Viterbi algorithm. The same Viterbi algorithm is also used to train the phoneme models. A very simple and effective method can be used to determine good initial probabilities from which the training can be started. For each language an initialisation file is composed. This file contains the obvious *(phoneme,grapheme string)* combinations that occur in the target language. Table 3 gives some examples for different languages. When the initialisation file is read by the alignment program, the probabilities P(S|P) that correspond with the *(phoneme,grapheme string)* combinations are given a value 1. All the remaining probabilities are given a small non-zero value. Next, the output probabilities are normalised so that their sum per phoneme model equals 1.

Table 3: Some initialisation examples for the 6 different languages (Ph: phoneme; Str: orthographic string)

Dutch	
Ph	Str
J	ij
J	ei
H	ch
H	g
y	u
y	uu

French	
Ph	Str
o	au
o	aux
o	eau
s	ç
E	è
E	ê

English	
Ph	Str
k	ck
T	th
f	ph
i	y
J	a
J	ai

German	
Ph	Str
v	w
v	v
S	sch
S	s
e	e
e	ä

Spanish	
Ph	Str
k	qu
k	k
b	b
b	v
a	a
a	á

Korean	
Ph	Str
a	a
a	aN
n	n
n	nN
t	t
t	s

3.3. EVALUATION

The proposed alignment method was evaluated on the 6 different languages we worked on. For each language, an initialisation file was created manually by a member of our text-to-speech team. This took only a few minutes per language.

For each language, the set of the 17,000 most frequent words (9,000 words for German) was subdivided in a training and a test set. The test sets always contained 2,000 randomly selected types. The alignments in the test sets were manually verified.

Some alignment examples are given in table 4. Performances can be found in table 5.

Table 4: Some alignment examples

```
English: |lightning|
         |lY""tnIN"|

French:  |beaucoup| (many)
         |bo""ku""|
```

The phoneme models that were created during the training on the Dutch, French and German words, were used as starting models for a training on the most frequent names (15,000 for Dutch and French; 9,000 for German). Again, the performance was checked on sets of 2,000 name types. Results are also given in table 5.

Table 5: Evaluation of the alignment method: results.

language	performance (%)	
	word	phoneme
Words		
English	95.0	98.2
French	95.1	98.2
Korean	94.0	98.3
German	95.8	98.7
Dutch	95.7	99.0
Spanish	97.3	99.9
Names		
Dutch	98.0	99.1
French	96.1	98.7
German	96.1	98.8

As can be seen in this table, the alignment performance between letters and phonemes is always better than 98 %. At least 94 % of the words or names show no alignment errors at all. It is important to notice that the errors are frequently the result of a very small number of different misalignments. Most errors are found several times throughout the data. They occur in a consistent way. Part of the errors are due to the fact that our alignment method is not very well in handling letter insertions, these are letters that do not correspond to a phoneme. For example, in Spanish, the letter *h* is never pronounced except when it is preceded by *c*. In a word such as *hombre*, the alignment method will align the vowel *o* with the string *ho*. To avoid problems with letter insertions, a special insertion model can be added to the set of phoneme models. The results given in table 5 were obtained without the use of insertion models.

4. SOME ORTHOGRAPHIC/PHONETIC PROPERTIES OF SEVERAL LANGUAGES

The aligned lexical data bases are helpful to estimate and compare the complexity of the relationship between orthographic and phonetic representations for different languages. Also, the

usefulness of context dependent pronunciation rules can be estimated. This is done using the well-known concept of entropy and mutual information. In what follows, all values are calculated on our training sets with the most frequent words of each language, as described at the end of section 2.2. The entropy or mathematical uncertainty of the phonetic transcriptions is represented by $H(P)$. To calculate this entropy value, phonetic transcriptions were used as shown in figure 4. The dummy symbols that are used for alignment purposes were maintained in the transcriptions. For English, the entropy $H(P)$ equals 4.636 bits. This corresponds to a perplexity 2^H of 24.9 equally likely phonemes. It is interesting to calculate the entropy $H(P|L_0)$ of the pronunciations P if for each phoneme in the transcriptions the corresponding letter is known. For English, $H(P|L_0)$ equals 1.492 bits. The perplexity is 2.8 phonemes. From $H(P)$ and $H(P|L_0)$ the mutual information between phonemes and corresponding letters can be calculated. For English, $I(P|L_0)$ equals 3.144 bits of information. Table 6 gives figures for different languages. The different languages were sorted according to $2^{H(P|L_0)}$. The resulting order confirms the known fact that the orthography of English and French are quite complex contrary to the orthography of Spanish that is more phonetic oriented. See also the order in table 5.

Table 6: A comparison between different languages based on some information theoretic quantities.

(1)	language	(5)	perplexity $2^{H(P)}$
(2)	number of letters	(6)	entropy $H(P\|L_0)$
(3)	number of phonemes	(7)	perplexity $2^{H(P\|L_0)}$
(4)	entropy $H(P)$	(8)	information $I(P;L_0)$

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
English	26	40	4.636	24.9	1.492	2.8	3.144
French	41	37	4.197	18.3	1.308	2.5	2.889
Korean	24	42	4.717	26.3	1.200	2.3	3.516
Dutch	28	40	4.710	26.2	1.173	2.3	3.537
German	31	44	4.764	27.2	1.162	2.2	3.602
Spanish	32	28	4.251	19.0	0.427	1.3	3.824

It is also interesting to investigate the mutual information between a phoneme and the letter at a distance of i letters away from that phoneme. This mutual information value is represented by $I(P;L_i)$. Table 7 gives values for the different languages, except for German (There are only 9,000 words in the German word set; see section 2.2.).

Table 7: Mutual information $I(P;L_i)$ between phonemes and letters for different languages

i	Eng	Fre	Kor	Dut	Spa
-3	0.173	0.213	0.262	0.196	0.174
-2	0.303	0.305	0.485	0.347	0.341
-1	0.725	0.720	1.153	0.891	0.966
0	3.144	2.889	3.516	3.537	3.824
1	0.912	1.100	1.227	0.980	0.958
2	0.323	0.545	0.443	0.462	0.367
3	0.179	0.284	0.260	0.242	0.178

The letters to the right of the phoneme ($i > 0$) seem to be somewhat more informative than letters to the left, especially for French. Probably this is due to the fact that letter strings and corresponding phonemes are left aligned (see table 4). The effect is rather important for French, reflecting the fact that French phonemes correspond frequently with quite long orthographic strings.

It also seems that for Korean, the left and right letter contexts are somewhat more informative than for the European languages.

5. CONCLUSION

We reported on the development of lexical data bases for 6 different languages.

We described a method for the automatic alignment of orthographic and phonetic representations. The method was evaluated on different languages. Some preliminary statistics based on the concepts of entropy and mutual information were presented. These statistics were derived from our automatically aligned lexical data bases.

6. ACKNOWLEDGMENTS

The work described in this paper could only be realised thanks to the efforts of José Manuel Conejo, Tiene Depoorter, Sylvia Joos, Steven Leys, Luc Mortier, Jason Nicholson, Eun Young Park, Helen Williams and several other people.

The name data bases were derived from original Telecom data, kindly put at our disposal by the Belgian Telecom company *Belgacom*.

7. REFERENCES

[1] W.N. Francis, H. Kucera (1982), "Frequency Analysis of English Usage: Lexicon and grammar," Boston: Houghton Mifflin

[2] A. Juilland, D. Brodin, C. Davidovitch (1970), "Frequency Dictionary of French Words," The Hague: Mouton

[3] A. Juilland, E.C. Rodriguez (1964), "Frequency Dictionary of Spanish Words," The Hague: Mouton

[4] J.M. Lucassen (1983), "Discovering Phonemic Base Forms Automatically: an Information Theoretic Approach," IBM Research Report RC 9833 (# 43527)

[5] S.O. Lee (1982), "The Second Best Compromise: The National Academy of Sciences' Proposal on Romanization of Korean," Korea Journal, Vol. 22, pp. 5-15.

[6] I. Rosengren (1972), "Frequenzwörterbuch der deutschen Zeitungssprache," Lund.

[7] P.C. Uit den Boogaart (1975), "Woordfrequenties in geschreven en gesproken Nederlands," Utrecht: Oosthoek, Scheltema & Holkema

[8] B.M. Van Coile (1989), "The DEPES Development System for Text-to-Speech Synthesis," Proceedings ICASSP-89, Glasgow, vol. 1, pp. 250-253.

[9] B.M. Van Coile (1990), "Inductive Learning of Grapheme-to-Phoneme Rules," Proc. ICSLP-90, Kobe, vol. 2, pp. 19.1.1.-19.1.4.

[10] B. Van Coile (1991), "Inductive Learning of Pronunciation Rules with the DEPES System," Proc. ICASSP-91, Toronto, Vol. 2, pp. 745-748.

[11] B. Van Coile, Steven Leys and Luc Mortier (1992), "On the Development of a Name Pronunciation System," Proc. ICSLP-92, vol. 1, pp. 487-490.