Automatic phonetisation for Icelandic.

Björn Kristinsson
University of Iceland

1. Introduction

As a part of my final thesis in language technology, I created a speech synthesiser using the free MBROLA[1] system. MBROLA is a project designed to make speech synthesisers for as many languages as possible available for free. It does not require a lot of technological prowess for the general user to create such a synthesiser: all that is required is segmented speech data, and the rest is handled by the experts at the Faculté Politechnique de Mons in Belgium.

The resulting speech synthesisers are usually very good, especially considering that most are made by amateurs in their spare time, but they are also quite primitive in that they are only phoneme-to-speech synthesisers, as opposed to text-to-speech, which is what most people think of when they hear of a speech synthesiser.

The step from phoneme-to-speech to text-to-speech is a very large one. It includes generating a phonetic transcription of what is to be said, as well as prosody. Depending on the complexity of the system, this may include complex grammatical analysis and tagging. I will be focussing on how best to perform phonetisation where data about the text being phonetised are limited. I will begin by looking at different ways of phonetisation, then describe some results of ongoing experimentation with one of those methods, before exploring the problems that were encountered.

[1] http://tcts.fpms.ac.be/synthesis/mbrola.html

2. Automatic phonetisation

There are numerous ways of translating written text to phonetic transcription. Dutoit mentions two basic strategies: dictionary-based and rule-based (Dutoit 1997:111).

Dictionary-based systems are, as the name implies, based on vast phonetic dictionaries where the phonetic transcription of a word can be looked up. For out-of-vocabulary words, rules are applied.

Rule-based systems on the other hand "transfer most of the phonological competence of dictionaries into a set of letter-to-sound […] rules" (Dutoit 1997:111). Only irregular or otherwise exceptional words are stored in a so-called exception dictionary. As the only phonetic dictionary available in Icelandic contains only about 50,000 word forms, rule-based systems were quickly favoured over dictionary-based ones.

Perhaps the most obvious way to create a rule-based system would be writing a few rules by hand that dictate how a grapheme is pronounced in a certain context, but it soon becomes evident that this will become a very laborious task. Generally, there is no one-to-one relationship between a grapheme and phoneme, but every conceivable context has to be found and described. And while there are rules that are mostly adhered to, there are always exceptions. The fact that these exceptions generally occur in the most common words hardly helps matters. As a drastic example of this, consider the fact that "*f* is always pronounced /f/ in the 20,000 most frequent words of English, except for the single case of *of*" (Dutoit 1997:106).

2.1 Hand-written rules

The initial experiments with phonetisation for the Icelandic MBROLA voice were based on hand-written rules. A perl script was written that looked at a five-letter string at a time and looked for a match in a database. If no match was found, it tried again after reducing

the string to four letters and so on, until a match was found. Initially it seemed that this could give some decent results, and indeed quite soon it reached an accuracy rate of about 80%. Any improvement on this figure was hard to achieve however: soon the database spiralled out of control with numerous conflicts appearing with nearly every addition, and finding those conflicts was not always easy. Combining this method with the dictionary-based approach might work as a quick-and-dirty phonetisation system.

For hand-written rules to work properly however, a better approach should be tried. Using more generalised rules, where features rather than individual phonemes are examined in context, could improve things. A method for detecting compound words and grammatical categories would improve matters even more. With hand-written rules, the more the computer knows about the text, the better it is at phonetising it.

2.2 Corpus based rules

Rules made automatically by the computer will be quite far from any phonetic theory. Rather, they are created to give the right result while the means to arrive at that result are less important. While this can be considered a disadvantage, it is more like how a computer works, and perhaps a more appropriate method for that reason. Hand written rules tend to be influenced by our own understanding of the language, while the computer does not have this advantage. Automatic rules will be based on exactly the information the computer can use, and nothing more.

There are numerous ways of training a computer on phonetic data. Simple hidden Markov models have been shown to be effective (Dutoit 1997:120-121), and more advanced neural networks have also been suggested, although these still have some way to go (Dutoit 1997:123, Burileanu 217-218).

For this project a small program called *t2p*[2], written by Kevin Lenzo, was used. It is a program for building rules from a pronunciation dictionary, where each letter in the written form is matched with either a phoneme or silence. After the initial alignment is done, feature vectors can be created. These show each letter in context of three preceding and three following graphemes, where L stands for the letter itself, L1 the next letter to the left, R1 the next to the right and so on (Lenzo 1998).

L3	**L2**	**L1**	**L**	**R1**	**R2**	**R3**	**Phoneme**
_	_	_	e	l	d	h	**E**
_	_	e	l	d	h	6	**l**
_	e	l	d	h	6	s	**t0**
e	l	d	h	6	s	b	_
l	d	h	6	s	b	E	**U:**
d	h	6	s	b	E	k	**s**
h	6	s	b	E	k	k	**p**
6	s	b	E	k	k	u	**E**
s	b	E	k	k	u	r	**h+k**
b	E	k	k	u	r	_	_
E	k	k	u	r	_	_	**Y**
k	k	u	r	_	_	_	**r0**

Example of t2p's grapheme-to-phoneme alignment. The word is *eldhúsbekkur* (kitchen counter).

In the summer of 2003 the first Icelandic pronunciation dictionary was created during the making of an Icelandic ASR. This consisted of 50.000 words, transcribed by hand, and would make an ideal corpus for training. However, it needed to be adapted to the program's needs.

First of all, the phonemes needed to be separated by spaces, which was not how the corpus was originally transcribed. This was a fairly simple task with the help of a text editor and some regular expressions.

[2] t2p can be downloaded from http://www-2.cs.cmu.edu/~lenzo/t2p/

The big problem was when it came to the actual aligning. The program can assign only one sound or no sound to a grapheme, nothing else. Any word that had more phonemes than graphemes was ignored.

s n j ó s l e ð i

s t n j ou: s t l ɛ: ð ɪ

Snjósleði (sleigh). More phonemes than graphemes, alignent failed, word ignored.

Worse, in some cases a word included both a grapheme that represented two phonemes, and another grapheme that was silent. This meant that there was the same number of graphemes and phonemes, and so the program would attempt to align them. All the graphemes between the "diphone grapheme" and the silent grapheme would then be misaligned, skewering the results.

e p l a p r e s s a

ɛ h p l a p r̥ ɛ s a

Eplapressa (apple press). Graphemes and phonemes can be aligned, and will be aligned, but wrongly.

To avoid this problem, some of the phone pairs would have to be treated as a single unit. Most of the misalignments were caused by a fairly small set of graphemes: preaspiration (usually represented by an /h/ and the following stop) and the combinations *sl* and *sn* (transcribed with an intrusive stop, /stl/ and /stn/ respectively). These were tied together by inserting a plus (+) sign between them.

s n j ó s l e ð i

s+t n j ou: s t+l ɛ: ð ɪ

Snjósleði. This time the combined phoneme pairs can be aligned with a single grapheme.

e p l a p r e s s a

ɛ h+p l a p r̥ ɛ s _ a

Eplapressa. Again the combined pair can be aligned with a single phoneme, and now the silence can be inserted in the appropriate place.

3. Results

Right from the start the results looked promising. *Velkomin*, (welcome), was a notorious case with an old Icelandic synthesiser, that pronounced it as [vɛl̥kɔmɪn], devoicing the /l/. This is a rather obvious mistake, as the rules of Icelandic pronunciation state that usually an /l/ followed by an aspirated stop should be unvoiced. However, velkomin is a compound word (made up of *vel* (well) and the verb *koma* (to come)) and the rule does not apply over word boundaries. The trained phonetiser passed this litmus test by pronouncing it as [vɛlkʰɔmɪn]. Furthermore, most words where the /l/ should be unvoiced were correctly predicted.

For a speaker to be able to predict the correct pronounciation of *velkomin*, some understanding of the language is required. One would expect that some method of automatically detecting that *velkomin* is a compound word would be needed for a computer to do the same. But all the data the phonetiser is given is a sequence of graphemes. That the automatic corpus based phonetiser can predict the pronunciation fairly accurately based on this very limited data is encouraging.

A short experiment, using a random 200 word article from a news site, shows that counting every error the program makes, 94.7% of the phonemes are correctly predicted. This means that 69.7% of the words are correct. This error rate is hardly acceptable, but many of the errors can be explained away by two factors. The first is that the pronunciation dictionary only has words standing on their own, not in context. This explains a great deal of the errors occurring at word boundaries. The other factor is function words

that are usually unstressed in regular speech, but not in the dictionary. A large number of these errors could be fixed afterwards by a separate program. If we remove these errors and assume for the moment they can be fixed, the accuracy rates improve to 98.9% and 93.6% respectively. A figure somewhere between those numbers should be attainable.

4. Problems

The faults of the system are numerous. Just as the abstract nature of automatic phonetisation may be helpful to a system that works on an abstract basis as was mentioned above, it also makes it extremely hard or even impossible to correct the system when it makes an error. The rules are mostly unreadable to humans, and usually the system must be retrained to try and rectify the problems encountered. Finding the source of the error can be a daunting task.

It is perhaps encouraging that most errors did seem to stem from incorrect transcription rather than the system misinterpreting the data. However, there are a few errors that is hard to rectify or make sense of. Strings like 'h+p' for words such as *epli*, [ehplɪ] (apple), would also be in words like 'keppni', [cʰehpnɪ] (competition). While a compound is required in the first case so the alignment can be made, it is not in the second case, and so either one of the 'p's could have silence assigned to it. And this is indeed what happened: some of the 'pp' cases were aligned as 'h+p _', while others were '_ h+p', which in turn means that an individual 'p' can either be realised as 'h+p' or '_'. In some cases, then, both 'p's were assigned a silence during the phonetisation, and so 'pp' was realised as '_ _'. A silence.

It is not always obvious which sounds to join together. I mentioned above that 'sl' is usually pronounced /stl/, and that these phonemes

could be joined up as /s t+l/. It is however far from obvious whether the /t/ is a part of the /s/ or the /l/. In fact, the main reason I decided to join /t/ and /l/ was to avoid problems like

v í x l a
v i k+s+t l a
v i k+s t+l a

I decided to try and limit these compounds to pairs, but what method gives the best results remains to be seen.

Despite these problems, for the most part the transcriptions are correct. Faults caused by mistakes in the data are hard to find and correct, but by using a perl script for correcting known problems, the accuracy of the phonetisation goes up dramatically. So both systems require a pronunciation dictionary of some sort, but the base is more solid with automatic rule generation when the data is limited.

5. Conclusion

Phonetisation is an important part of a successful synthesis system. There are many problems that have not been covered in this short essay, for instance how to treat abbreviations and numbers, foreign words and so forth. I have merely looked at the very basic aspects of phonetisation.

The methods for generating this basic phonetic transcription of text are varied with different strengths and weaknesses. With limited methods of acquiring data automatically from the text, but with a decent phonetic corpus for training, automatic corpus-based rule generation will give the best results.

Sources

Burileanu, Dragos. 2002. Basic Research an Implementation Decisions for a Text-to-Speech Synthesis System in Romanian. *International Journal of Speech Technology 5*, pp. 211-225. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Dutoit, Thierry. 1997. *An Introduction to Text-to-Speech Synthesis*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Lenzo, Kevin A. 1998. *s/($text)/speech $1/eg;* <http://www-2.cs.cmu.edu/~lenzo/tpj/tpj12_synthesis.html>.