Is it possible to train a speech recognition system on text only?

Enrico Rubagotti
UCD School of Computer Science and Informatics
UCD, Dublin, Ireland
enrico.rubagotti@ucd.ie

Abstract

According to speech recognition literature, one cause of recognition error is the difference in training and testing conditions. One cause of this is the use of speakers with different accents in training and testing. This is because, in the stochastic and deterministic approaches, the system is trained on pairs of acoustic signal- linguistic units. This paper describes the development of a training system that employs only graphemes and studies the feasibility of a model that employs the speech signal, a bigram model, frequencies of four grams and a distance measure of a text from a specific language to recognize speech. This system should be independent of variations in pronunciation and employable in languages for which a corpus has not yet been developed. A model was specified in the class of shallow languages and an experiment was carried out using a phonotypical transcription in Italian with a 22% WER. The input of the system was not the acoustic signal but phonemes to reduce the computational complexity in this preliminary phase. The algorithm employed in the test maps from phonemes to graphemes using a map that dynamically changes to minimise the distance of the output from the expected language. The difference between conventional phoneme parsing and our method is that in the conventional method the mapping phoneme grapheme is given before the recognition procedure, whereas in our method the map that is chosen is the one that minimises the distance between the output and the expected language.

1. Introduction

In speech recognition, the different accents used in training and testing constitute a significant cause of error. The WER of actual speech recognition system is influenced by the difference between training and testing conditions (e.g. [1] [*2*:42] [*3*:28]). The standard speech recognition model [4:109-126][5:9] [3] deals with the following input:

- a speech signal;
- a language model;
- parameters estimated on a pairs acoustic signal-linguistic unit from a training corpus.

The corpus contains pairs acoustic signal linguistic unit that will match some speakers and not others. It is the main hypothesis of this paper that avoiding the use of the corpus to estimate the parameters will equalize training and testing conditions. This paper tests the feasibility of a model that deals with the following input:

- a string of phonemes;
- statistical information on the graphemes output.

In this case the system is trained on text and tested on phonemes. The text used for the training is not the text corresponding to the string of phonemes. To reduce the computational complexity we substituted the acoustic signal by a string of phonemes and, as a consequence, only one small part of the speech recognition task is tackled. The experiment is done under the assumption that if simplifying the conditions does not work (using phonemes instead of the acoustic signal) it will not work in more complex conditions (using the acoustic signal). If it works using phonemes we will develop a system that use the acoustic signal as the input, otherwise this model will be rejected. From this paper it should be possible to draw the following conclusions:

- The WER in this model is constant because training and testing conditions do not differ;
- The WER in this model is not constant, so the model does not equalise training and testing conditions.

Speech recognition is defined as mapping between a digital description of the acoustic signal and a sequence of words. One definition is:
"The ASR problem consists of finding the sequence of words **W** associated to a given acoustic sequence**.** In mathematical terms, recognition is a function" M "that maps a given **X** belonging to the set of the whole acoustic sequences X to a W that is included in the set w ". [5:8] In the class of shallow languages there is a one-to-one mapping of phonemes to graphemes and for this reason Italian was employed in the experiment [6]. In the next section the concept of shallowness of a language and its importance in phoneme to grapheme conversion will be introduced. The paper is organized as follows: section 2 contains a description of the cryptospeech model, section 3 contains a description of the spell-checker, section 4 contains a description of the experiment, section 5 reports on the model's performance, section 6 draws the conclusion.

1.1 Shallowness of a language

Languages in which there is a nearly one to one correspondence between phonemes and graphemes (Spanish, Italian, Serbo-Croatian) are described [6] as shallow. It follows from this definition that in the class of shallow languages it is possible to map from a single phoneme to a single grapheme. This is very significant in phoneme to grapheme conversion. In Cryptography this is called monoalphabetical substitution [7] and algorithms exist which, when applied to the phoneme conversion context, allow the system to learn the phoneme to grapheme mapping given a string of phonemes and 4-grapheme statistics. A

number of authors studied machine learning techniques to map from phonemes to graphemes [8][9] but all have in common a training phase based on phoeneme-grapheme pairs. The training of this system is on graphemes only and this is the innovative part.

2. The cryptospeech model

I named the model due to the contribution made by cryptanalysis algorithms [7] to this paper. Speech recognition consists in mapping a digital representation of the signal to a series of linguistic units. As shown in figure 1 the inputs for this model are:

- a string of phonemes;
- 4-grams' percentages in the language.

The output is a graphemes' string. It is defined as $\hat{W}$, function of the input signal X and the mapping $\underline{M}$ i.e $\hat{W}(X,\underline{M})$. To reduce the computational complexity this implementation of the model does not deal with the acoustic signal but with phonemes. An implementation of the system that uses the signal will be developed in future. One example of mapping phonemes➔graphemes is given in table 1.

/p/➔p
/b/➔b
/t/➔t
/d/➔d
…
/k/➔ c

Table 1: *Example of phoneme➔ grapheme mapping*

It is possible to learn phoneme ➔ grapheme mapping by minimizing a metric that measures the distance of a string from a subset of a language. The metric will be explained in section 2.1. The output is spell-checked and the out of vocabulary words are substituted with the ones in the vocabulary with a smaller distance. The metric used was the Levenshtein distance [10]. The estimated sequence of words is the one that employs a phoneme to grapheme map that minimizes the distance of the output from a subset of the language (e.g. an Italian text). This is denoted by the formula:

$$\hat{W}= \arg\min_{\underline{M}} d(\hat{W}(X, \underline{M}),L) \quad (1)$$

The difference between this approach and template matching is the specification of the distance measure and the minimization function. In the former, the distance measures the similarity between a stored word form and an input. In the latter the distance measures the similarity between the output and a language. In the former, the template that minimizes the distance from the acoustic signal is chosen. In the latter a map that minimizes the distance of the output from the output language (e.g. a text in Italian) is chosen. The minimization algorithm is explained in section 2.2.

2.1 The distance measure between a string and a subset of a language

The frequency of N-graphemes in a language is approximately constant [11:7]. As a consequence it is possible to describe a subset of a language using a vector of N-grapheme frequencies. For this reason the literature (cryptanalysis [7] and language identification[12]) uses as a distance of the similarity of a string from a subset of a language the sum of the absolute values of the differences in frequencies **(D_{ijlm})** of 4-graphemes:

$$\mathbf{d(\hat{W},L)}= \sum_{i,j,l,m} \left| D_{ijlm}(\hat{W}(X,M)) - D_{ijlm}(L) \right| \quad (2)$$

Where i, j, l, m are letters of the alphabet

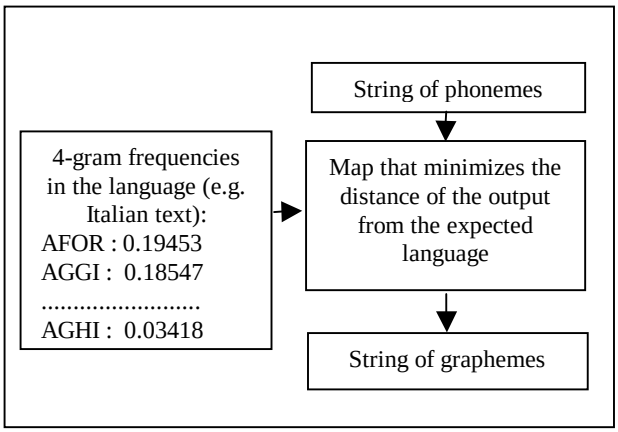

Figure 1:Cryptospeech is described by this flow chart

2.2 The algorithm employed

The software's documentation does not exactly explain the algorithm used but, compared with the literature reviewed, suggests the flow chart in figure 2. The algorithm generates a random phonemes-graphemes matching, generates all the possible permutations obtainable swapping 2 characters, and considers the one corresponding to the least distance to be as best match. Once all the letter's pairs are swapped another random key is generated. The algorithm stops when it reaches a plateau evaluated by a user. The flow chart for this algorithm is presented in figure 2. The output is run through the spell checker described in section 3.

2.3 Example

In this example the distance between a sample of Italian text and a string will be measured. In table 2 the distance between the two strings is calculated using, as an example, only the first eight 4-graphemes. The first column (4-grapheme) contains the 4-graphemes. The second one (% String) contains the percentage of those 4-graphemes in the string. The third one contains the percentage of those 4-grapheme in the sample of Italian text. The last one contains the absolute value of the difference between the two percentages. The absolute values are added to obtain the distance in the last cell on the right.

4-grapheme	**% String**	**%Sample**	**|Difference|**
BERG	1.8405	1.8405	0
DRFZ	1.8405	0	1.8405
ENBE	1.8405	1.8405	0
FZEC	1.8405	0	1.8405
GIOE	1.8405	0	1.8405
GUTE	0	1.8405	1.8405
IOEN	1.8405	0	1.8405
Tot			9.2025

Table 2: Example of the calculations necessary to compute the distance. The probabilities are in per cent.

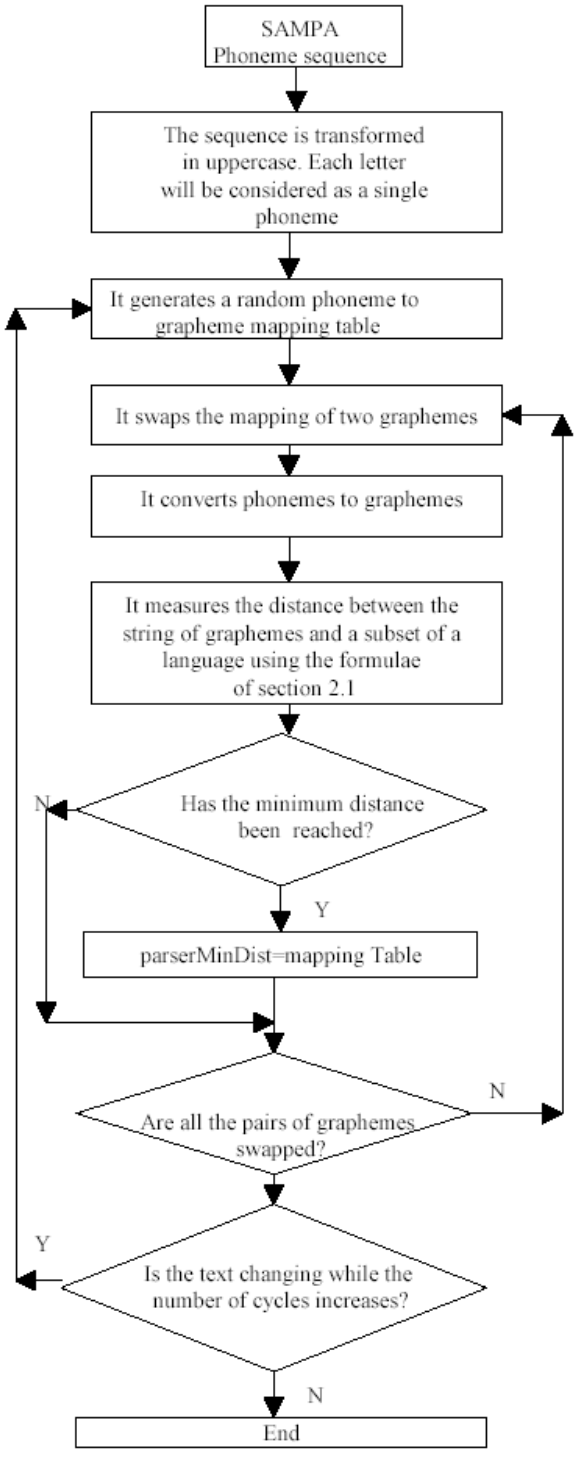

Figure 2: The algorithm employed

3. The spell checker

It relies on a language model used as a grammar: it does not contain probabilities but only permitted sequences. The following table is an example of the language model employed.

1st word	**2nd word**
MIO (my)	CANE (dog) GATTO (cat) CANCELLO (gate) TELEFONO (telephone)
IL (the)	CIOCCOLATO (chocolate) PRESIDENTE (president) GELATO (ice cream) QUADERNO (notebook)

When an out of vocabulary word is found a series of possible candidates is selected. The candidates are the ones that, following the language model, should follow the preceding word. Between them the ones with minor Levenshtein [10] distances are chosen.

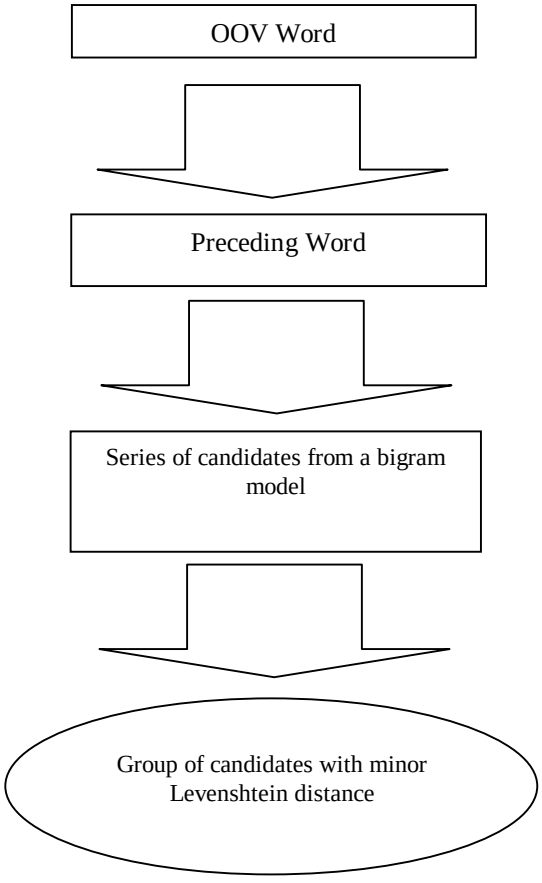

Figure 3 The spell checking algorithm

Example

The Italian word “PIU” was misspelled as “PJU”.
The preceding word was E, the bi-gram model is:

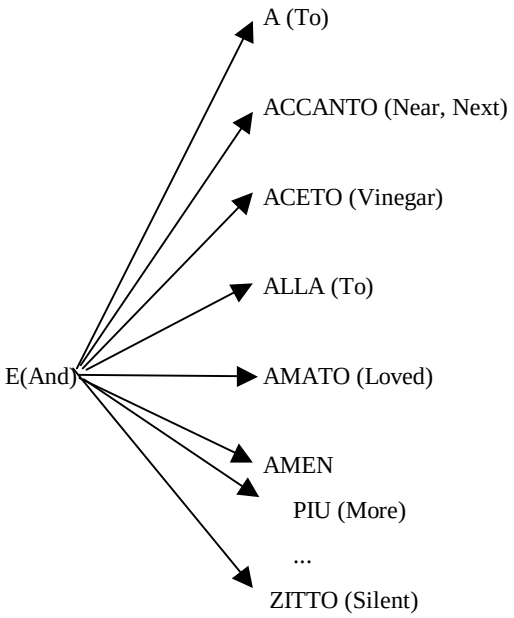

Figure 4

The Levenshtein distance for each candidate word was measured, and the ones that minimized the distance from “PJU” were selected. In this case the number of candidates was 633 and the software found the right one.

4. The experiment

The experiment has the following objectives:

1 To check that the WER in this model is constant, so the aim of equalizing training and testing conditions are achieved;

2 To compare SCBSolvr’s performance with a phoneme-grapheme parser;

3 To check the performance of Cryptospeech after spell checking.

4 to prove that is possible to learn the phoneme to grapheme mapping having as input data a string of phonemes and statistics of 4-graphemes.

The software employed to convert phonememes to graphemes was SCBSolvr [13]. I have taken Italian data from the EUROM_1 corpus which consists of phonotypical Italian transcription along with corresponding orthographic transcription. The phonotypical transcription is going to be used as input and the orthographic transcription will be used purely for the purpose of evaluation. A corpus of 4-graphemes for Italian is being constructed from the Guthemberg project.

This is an example of the phonotypical transcription:

"il bra"zile "E "il "reJJo "del ka"kao
ultima"mente "E "pju zgar"bato "del "sOlito

This is the corresponding orthographical transcription:
Il Brasile e’ il regno del cacao
Ultimamente e’ piu’ sgarbato del solito

This is Cryptospeech’s output after 6.000 cycles:
pe wlrhpei i pe lizzt die arart
ceupfrfioui i bzc hsrlwrut die nteput

This is Cryptospeech’s output after 190.000 cycles (10 seconds):
il brazile e il rejjo del cacao ultimamente e pju zgarbato del solito

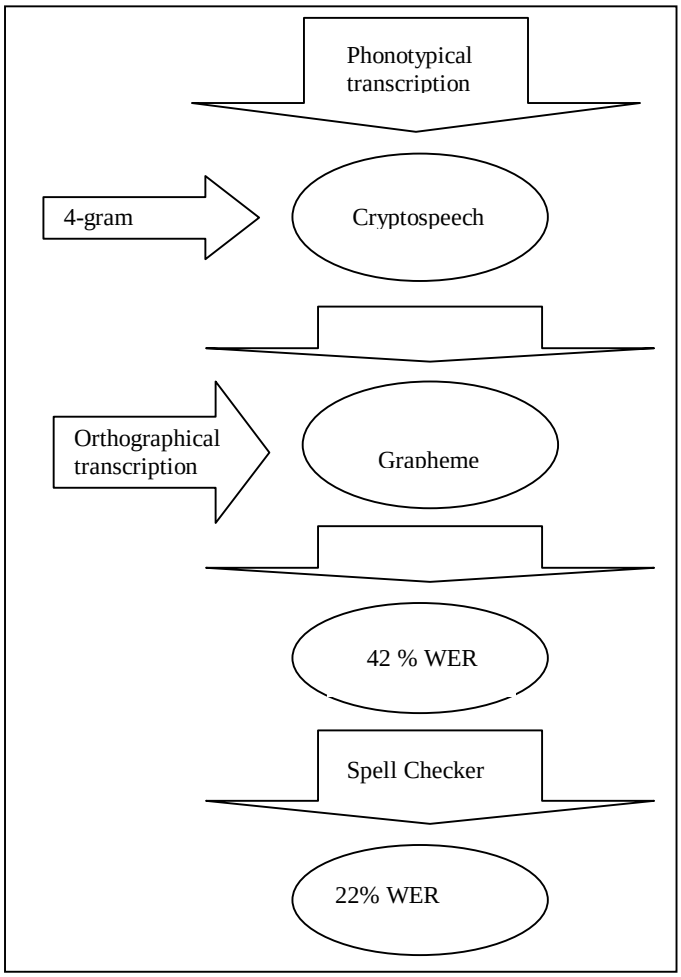

Figure 5: The experiment

The software employed (SCBSolvr [13]) deals with 27 symbols and for this reason the number of input phonemes (Italian SAMPA) was reduced from 43 to 27 representing phonemes indicated by SAMPA by more than one letter as multiple phonemes. The following is a list of some SAMPA symbols that were not represented properly:

SAMPA Symbol	**Input to the system**
/tts/	T,T,S
/ts/	T,S
/dz/	D,Z
/tS/	T,S
/dZ/	D,Z
/ddz/	D,D,Z
/ttS/	T,T,S
/ll/	L,L

This reduction could have influenced the experiment results and for this reason a SAMPA consistent system will be

developed. The experiment consists of three tests reported below. The first two were done on two passages of 1121 words (passage 1) and 1174 words (passage 2). It compares two systems (Cryptospeech and hand defined map) and two different passages of the corpus. The third one was done on a reduced sample (93 words) to be used for the spell checker.

Test 1

The phonotypical transcription of passage 1 was used as an Input to SCBSolvr. The output was compared with the orthographical transcription resulting in a 40.7 %WER.

The phonotypical transcription of passage 2 was mapped to graphemes. The output was compared with the orthographical transcription resulting in a 34.5 %WER.

The two WER are statistically different [14:401]

Test 2

The phonotypical transcription of passage 2 was used as an Input to SCBSolvr. The output was compared with the orthographical transcription resulting in a 41.1 % WER. The phonotypical transcription of passage 1 was mapped to graphemes. The output was compared with the orthographical transcription resulting in a 40.7 % WER.

The two WER are not statistically different.

Test 3

93 phonotypical transcribed word from "EUROM_1 Italian" were used as an Input to Scbsolvr. The resulting WER was 42%. The output of Cryptospeech test 3 was spell checked using the algorithm described in section 3 and the WER drops to 22%.

There follows an evaluation of the hypothesis after the results of the experiment .

Hypothesis 1

The difference between the SCBSolvr's WER in tests 1, 2 and 3 results are not statistically significant . So the first hypothesis that in this model training and testing conditions are the same and will have consistent WER is accepted.

Hypothesis 2

The first two tests give contradictory indications on the performance of Cryptospeech against a phoneme-grapheme parser and for this reason successive studies are necessary.

Hypothesis 3

The third test shows that Cryptospeech's WER decreases notably when using a spell checker.

Hypothesis 4

In the experiment phonemes were converted to graphemes without a previous mapping with a WER of nearly 40% before spell checking and 22% after spell checking.

The performance

In the tests the WERs of Cryptospeech were between 34% and 41%. The two tests give a contradictory result on the equivalence of Cryptospeech and of a phoneme parser and for this reason more investigation in this area is needed. In the third test Cryptospeech's output was spell checked and the WER drops to 22%.

5. Conclusion

The objective of this paper was not to show that the system implemented has a low WER but that in this system training and testing conditions were the same and as a consequence it has a constant WER. This aim was achieved. The experiments were performed by reducing the number of phonemes from 50 to 27 - due to the limitations of the software employed. To understand it's influence on the WER a new software will be developed that deals with 50 symbols. In future it will be necessary to study algorithms able to deal with languages using deep orthographies. This will be done following the path paved by W. Daelemans and A.van den Bosch [15] for grapheme to phoneme conversion.

References

[1]F.Schiel-A.Kipp-H.G.Tillmann-Statistical modelling pronunciation: it's not the model, it's the data-Modelling pronunciation variation for Automatic Speech Recognition Rolduc, 4-6 May 1998

[2]S.Young-G.Evermann-D.Kershaw-G.Moore-J.Odell-D.Ollason-V.Vatchev-P.Woodland-The HTK Book-Cambridge University Engeneering Department-2002

[3] F.Jelinek-Statistical methods for speech recognition-The MIT Press-1997

[4]J.Holmes, W. Holmes-2001-Speech synthesis and recognition- Taylor and Francis

[5] C.Becchetti and L.P.Ricotti-Speech Recognition Theory and C++ implementation-John Wiley & Sons 1999

[6] A. van den Bosch-A.Content, W.Daelemans, B.de Gelder-Measuring the complexity of writing systems- Journal of Quantitative Linguistics, 1994

[7]T.Jakobsen-A fast method for the cryptanalysis of substitution ciphers- *Cryptologia*, 19(3), 1995.

[8] B. Decadt, J.Duchateau, W.Daelemans, Patrick Wambacq
Memory-based phoneme-to-grapheme conversion Reference: In M. Theune, A. Nijholt, and H. Hondrop (Eds.), Computational Linguistics in the Netherlands 2001. Selected Papers from the Twelfth CLIN Meeting, Amsterdam - New York: Rodopi, pp. 47-61

[9] Rentzepopoulos, P. A., & Kokkinakis, G. K. (1996). Efficient multi-lingual phoneme-to-grapheme conversion based on HMM. Computational Linguistics, 22(3), 351–376.

[10] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Doklady Akademii Nauk SSSR, 163(4):845-848, 1965 (Russian). English translation in Soviet Physics Doklady, 10(8):707-710, 1966

[11]Ted Dunning Statistical Identification of Language Computing Research Laboratory New Mexico State University March 10, 1994

[12]W. B. Cavnar, John M. Trenkle N-Gram-Based Text Categorization (1994) Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval

[13]Monoalphabetic Substitution Cipher Solver Program http://secretcodebreaker.com/download.html

[14]G.Cicchitelli M.A.Pannone Complementi ed esercizi di statistica descrittiva ed inferenziale-1991-Maggioli editore - Rimini

[15] W. Daelemans-A. van den Bosch- A language independent, data oriented Architecture for grapheme to phoneme conversion-Proceedings ESCA-IEEE speech synthesis conference, New York, September 1994