G2PIL: A GRAPHEME-TO-PHONEME CONVERSION TOOL FOR THE ITALIAN LANGUAGE

Michele Salvatore Biondi[1], Vincenzo Catania[2] Raffaele Di Natale[3] Ylenia Cilano[5]
Antonio Rosario Intilisano[4] Giuseppe Monteleone [6] Daniela Panno[7]

[1]DIEEI, University of Catania, Italy
[2]DIEEI, University of Catania, Italy
[3]DIEEI, University of Catania, Italy
[4]DIEEI, University of Catania, Italy
[5] DIEEI, Giuseppe Monteleone, Italy
[6] A-Tono Corporate, Italy
[7]DIEEI, University of Catania, Italy

ABSTRACT

This paper presents a knowledge-based approach for the grapheme to-phoneme conversion (G2P) of isolated words of the Italian language. With more than 7,000 languages in the world, the biggest challenge today is to rapidly port speech processing systems to new languages with low human effort and at reasonable cost. This includes the creation of qualified pronunciation dictionaries. The dictionaries provide the mapping from the orthographic form of a word to its pronunciation, which is useful in both speech synthesis and automatic speech recognition (ASR) systems. For training the acoustic models we need an automatic routine that maps the spelling of training set to a string of phonetic symbols representing the pronunciation.

KEYWORDS

ASR Acoustic model, Phonetic Dictionary, Spoken Dialog Systems

1.INTRODUCTION

Automatic Speech Recognition (ASR) has evolved significantly over the past few years. Early systems typically discriminated isolated digits, whereas current systems perform well at recognizing spontaneous continuous speech. Huge effort has been spent for improving word recognition rates, but the core acoustic modelling has remained not stable or not available for language like Italian, despite many attempts to develop better alternatives [1]. Handcrafted creation of pronunciation dictionaries for speech processing systems can be time-consuming and expensive [2]. In our previous work an Acoustic Model for Italian language from speech corpora generated by Audiobooks has been created [3]. The new words obtained through speech corpora do not contain a corresponding phonetic transcription. As pronunciation dictionaries are so fundamental to speech processing systems, much care has to be taken to create a dictionary that is as free of errors as possible. Faulty pronunciations in the dictionary may lead to incorrect training of the system and consequently to a system that does not exploit its full potential. Flawed dictionary entries can originate from G2P converters with shortcomings [4]. In this paper we

present a graphemes to phonemes algorithm for Italian language that automatically generates correct phonetic transcription.
The paper is organized as follows. Section 2 presents the related Work. Section 3 presents the proposed grapheme-to-phoneme conversion approach followed by presentation of experimental results in section 4. Finally, we conclude in Section 5.

2. RELATED WORK

Grapheme-to-Phoneme (G2P) conversion is an important problem related to Natural Language Processing (NLP), Speech Recognition and Spoken Dialog Systems (SDS) development. The goal of G2P conversion is to accurately predict the pronunciation of a new input word. For example, to predict the word "SPEECH" the g2p generate the following,

SPEECH → S P IY CH

This problem is straightforward for some languages like Spanish or Italian, where pronunciation rules are consistent. For languages like English and French however, inconsistent conventions make the problem much more challenging [5]. There are many different approaches used for the G2P conversion proposed by different researchers. Statistical models such as decision trees, Joint-Multigram Model (JMM) [6], or Conditional Random Field (CRF) [7] are used to learn pronunciation rules. All these approaches invariantly assume access to "prior" linguistic resources consisting of sequences of graphemes and their corresponding sequences of phonemes [8]. Writing by hand, this rule is hard and very time consuming. The difficulty and appropriateness of using G2P rules are very language dependent. Currently Sphinx-4 [9] uses a predefined dictionary for mapping words to sequence of phonemes. Recently, Sphinx-4 is able to use trained models (based on machine learning algorithm) to map letters to phonemes and thus map words in a sequence of phonemes without the need of a predefined dictionary. A predefined dictionary will be used to train the required models. Our approach does not use a training method or initial dictionary, but grammatical rules only.

3. G2P ALGORITHM

This work proposes a novel grapheme-to-phoneme G2P conversion approach. One of the key issue when developing a G2P converter is how to effectively learn/capture the relation between phonemes and graphemes. Every step of this algorithm is based on a specific set of rules developed via linguistic engineering.

The substring with which the word ends is compared with a list of available desinences.

There is a list of desinences in which every desinence is written according to the following format:

```
vowel_or_consonant+desinence,category,part_of_speech,phonetical_transcription
```

- "vowel_or_consonant" indicates if "desinence" must be preceded by a vowel, a consonant, or both. This parameter is optional and can take 3 values:
    - V (vowel)
    - C (consonant)

  - VC (vowel + consonant)
- "desinence" is the analysed desinence, which is the string compared with the substring with which the word ends. The desinences has been obtained from [10]. For each desinence, inflected versions are obtained by analysing the category.

- "category" is used to get the inflections of each desinence. A letter represents each category. There is a list of categories in which each category is associated with some characteristics. Each characteristic is written according to the following format (in which: s=substring, pt=phonetic transcription, i=inflected):

$$\text{category: } s_{1,1}\text{-}pt_{1,1} > si_{1,1}\text{-}pti_{1,1},\ldots,\ si_{1,n}\text{-}pti_{1,n};\ \ldots\ ;\ s_{k,1}\text{-}pt_{k,1} > si_{k,1}\text{-}pti_{k,1},\ldots,\ si_{k,n}\text{-}pti_{k,n};$$

In general, if a desinence belongs to a certain category and it ends in $s_{k,1}$, to obtain the inflected form, $s_{k,1}$ is removed from the desinence and it is replaced with $si_{k,1}$, $pt_{k,1}$ is removed from its phonetic transcription and it is replaced with $pti_{k,1}$. This proceeding is applied for all eventual n inflections.

For example, consider the "D" category that is written as:

```
D:gia-dZ i! a>gie-dZ i! e;cia-tS i! a>cie-tS i! e;
```

This means that if a desinence belongs to the category D, if it ends in "-gia" and the correspondent phonetic transcription ends in "-dZ i! a ", the inflected form is obtained by removing "-gia" and adding "-gie " in the desinence and by eliminating " dZ-i! a " and adding "-dZ i! e " in the phonetic transcription.

There is also the "I" category for desinences that do not have inflected forms: I:;

- "part_of_speech" indicates the part of speech, is used because some equal desinences endings have different accents emphasis depending on the part of speech. The used parts of speech are:
  - V for verbs
  - N for nouns and adjectives
  - A for other

- "phonetical_transcription" is the phonetic transcription of the desinence.

The desinences that can be recognized can also contain an accent, placed in a particular position (represented by the symbol "!"). The presence of an accent, resulting in the final transcription is optional.

If no desinence has been identified, the word is analysed letter by letter to retrieve the transcription. In this case, if the word does not include accents and require the presence of the accents in the transcript, the algorithm generates a set of possible transcriptions with an emphasis on the different possible positions. The user will choose the correct transcriptions.

Even when the presence of accents is not required, the algorithm in some cases can generate multiple possible transcriptions for a certain word and, in this case, the user chooses the correct transcriptions.
In any case, a word can have multiple correct transcriptions (example with an accent: “Ankara” -> “a ng k o! r a” and “a! ng k o r a”; example without accent: “casale” -> “c a s a l e” and “c a z a l e”).

The creation of phonetic transcriptions analysing the word letter by letter is the main part of the algorithm. The Italian language uses a subset of the phonemes of the International Phonetic Alphabet (IPA). IPA is an alphabetic system used to represent the sounds in phonetic transcriptions. For ours phonetic transcription the Italian phonemes are used [11], except for two, in fact, there are two ways to pronounce the vowel “e” and two to pronounce the vowel “o”. Nevertheless, they generate a very similar sound, so in this algorithm a unique way to pronounce the “e” and a unique way to pronounce the “o” are used. Moreover, these sounds are so similar that different speakers can pronounce them in both different ways according to their geographical origin.

The rules for the algorithm are based on knowledge of the Italian language. In the Italian language, a specific sound corresponds to each combination of characters. Following, some examples of sounds produced by particular combinations of letters are listed:

- c + a, +he, +hi, o, u: a harsh sound is produced (as in words “cane”, “che”, “chilo” “corda”, “culla”) that corresponds to the phoneme “k”.
- c + e, i: a soft sound is produced (as in words “cielo”, “cena”) that corresponds to the phoneme “tS”.
- g + a, +he, +hi, o, u: a harsh sound is produced (as in words “gatto”, “ghetto”, “ghiro”, “gonna”, “gufo”) that corresponds to the phoneme “g”.
- g + e, i: a soft sound is produced (as in words “gelo”, “giro”) that corresponds to the phoneme “dZ”
- gn + a, e, i, o, u: as in the Italian word “gnomo”, this sound does not exist in English and corresponds to the phoneme “J” .
- h: no sound.
- gl + i, +e: as in the Italian word “figlio”, this sound does not exist in English and corresponds to the phoneme “L”.
- n +f, +v: as in the Italian words “invece” and “infatti”; in English it corresponds to the sound produced in the word “sy**mph**ony”.
- n +c, +g: as in the Italian word “anche”, in English it corresponds to the sound produced in the word “si**ng**”.
- sc + e, i: as in the Italian words “scena” and “scimmia”; in English it corresponds to the sound produced in “**she**ep”. The phoneme is “S”.

Anyway, it is possible to consider the accents: for this reason, for the stressed vowel the symbol “!” is added after the phoneme. The algorithm provides both phonetic transcriptions, with and without accent. It’s up to the user to choose the correct one. All rules implemented in the tool are not listed in this paper because they would exceed the page limit. The entire algorithm rules are implemented and listed in a c# VS2010 project and is available at this link [12].

PSEUDOCODE:

Input:

word W = {c_1, c_2, …, c_i, …, c_{n1}} where c is a character
(example: CASALE = {c,a,s,a,l,e})

Output:

list of phonetic transcriptions LP = {P_1, P_2, …, P_i, …, P_{n2}}
(example: LP= {casale,cazale})
where P = {p_1, p_2, …, p_i, …, p_{n3}} is a phonetic transcription and p_i is a phoneme
(example P_1={c,a,s,a,l,e}, P_2={c,a,z,a,l,e})

Algorithm:

```
FOR EACH c_i ∈ W
    I(c_i,r) = { c ∈ W : d(c,c_i) < r }
    IF( I(c_i,r) generates (p_x | p_y| ...) )
        FOR EACH P_i ∈ LP
            P_x  <-  P_i + p_x
            LP_TEMP.add(P_x)
            P_y  <-  P_i + p_y
            LP_TEMP.add(P_y)
            …
        LP = LP_TEMP
```

4. EXPERIMENTAL RESULT

The potential of the proposed approach, we considered the transcriptions contained in the Lexicon of Festival Speech Synthesis System (TTS) [13]. A *Lexicon* in Festival is a subsystem that provides pronunciations for words. It consists of three distinct parts: an addendum, consisting of manually added words; a compiled lexicon and a method for dealing with words not present in any list [14]. The third field contains the actual pronunciation of the word. This is an arbitrary Lisp S expression. In many of the lexicons distributed with Festival this entry has internal format, identifying syllable structure, stress markings and of course the phones themselves. In some of our other lexicons we simply list the phones with stress marking on each vowel. Festival, within its lexicon, includes more than 440,000 words. For each word all the List Expressions are removed and only the phonetic transcription is extracted. The comparison showed that only 24.634 out of 440,000 words were transcribed in a different way. These results confirm the quality of the algorithm with an Error Rate close to 5,59%.

5. CONCLUSION

This paper presents a knowledge-based approach to G2P conversion applied to highly inflected free-stress Italian Language. In the future, lexical stress extraction and filtering methods should improve the g2p models. Furthermore, we may integrate a speech synthesis component into a dictionary building process for accelerated and interactive editing of improper phonemes. The source code and binary of this G2P tool are available at this link [12].

ACKNOWLEDGEMENTS

The authors were supported by the Sicilian Region grant PROGETTO POR 4.1.1.1: "Rammar Sistema Cibernetico programmabile d'interfacce a interazione verbale" (utilizzare la dizione degli altri paper).

REFERENCES

[1] A. Mohamed, G. E. Dahl, and g. E. Hinton, "acoustic modeling usingdeep belief networks," IEEE trans. On audio, speech, and language processing, 2011.
[2] Cheap, Fast and Good Enough: Automatic Speech Recognition with Non-Expert Transcription Scott Novotney and Chris Callison-Burch Center for Language and Speech Processing Johns Hopkins University
[3] V. Catania, S. M. Biondi, Y. Cilano, R. Di Natale, A.R. Intilisano, G. Monteleone. 2014."An Audiobooks-Based Approach for Creating a Speech Corpus for Acoustic Models".(in press)
[4] Tim Schlippe, Wolf Quaschningk, Tanja Schultz, "combining grapheme-to-phoneme converter outputs for enhanced pronunciation generation in low-resource scenarios". Cognitive Systems Lab, Karlsruhe Institute of Technology (KIT), Germany
[5] J. Novak, N. Minematsu, and K. Hirose, "WFSTbased Grapheme-to-Phoneme Conversion: Open Source Tools for Alignment, Model-Building and Decoding," in International Workshop on Finite State Methods and Natural Language Processing, Donostia-San Sebastian, ´ Spain, July 2012.
[6] V. Pagel, K. Lenzo, and A. W. Black, "Letter to sound rules for accented lexicon compression," in Proceedings of ICSLP, 1998.
[7] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," Speech Communication, vol. 50, no. 5, pp. 434–451, May 2008.
[8] D. Wang and King. S, "Letter-to-sound pronunciation prediction using conditional random fields," IEEE Signal Processing Letters, vol. 18, no. 2, pp. 122–125, 2011
[9] CMU SPHINX, G2P Conversion, (last visited [23 July 2014]), http://cmusphinx.sourceforge.net/2012/08/gsoc-1012-grapheme-to-phoneme-conversion-in-sphinx-4-%E2%80%93-project-conclusions/
[10] L. Canepari, "Dizionario di pronuncia italiana", (last visited [23 July 2014]), http://www.dipionline.it/guida/docs/DiPI_Terminazioni_Desinenze.pdf
[11] "Wikipedia", (last visited [23 July 2014]), http://it.wikipedia.org/wiki/Fonologia_della_lingua_italiana
[12] "DIEEI Embedded System Lab", http://ctsds.dieei.unict.it/research/index.php/download
[13] "Festvox project", (last visited [23 July 2014]), http://festvox.org/
[14] "Festival Lexicon tutorial", (last visited [23 July 2014]), http://festvox.org/http://www.cstr.ed.ac.uk/projects/festival/manual/festival_13.html

Authors

Vincenzo Catania received the Laurea degree in electrical engineering from the Università di Catania, Italy, in 1982. Until 1984, he was responsible for testing microprocessor system at STMicroelectronics, Catania, Italy. Since 1985 he has cooperated in research on computer network with the Istituto di Informatica e Telecomunicazioni at the Università di Catania, where he is a Full Professor of Computer Science. His research interests include performance and reliability assessment in parallel and distribuited system, VLSI design, low-power design, and fuzzy logic.

Daniela Panno received a Dr.Ing degree in Electrical Engineering and a Ph.D. in Telecommunications Engineering from the University of Catania, Italy, in 1989 and 1993, respectively. In 1989 she joined the Department of Informatics and Telecommunications at

the University of Catania, where she is now an Associate Professor of Signal Theory. Her research interests are MAN architectures and protocols, traffic management and performance evaluation in broadband networks, fuzzy logic application in the telecommunications field.

Raffaele Di Natale is a Contract Researcher at the DIEEI at the University of Catania. He received his M.Sc. degree in computer science from Catania University, Italy, in 1997 and the Ph.D. in bioinformatics from Catania University in 2012. His research interests include spoken dialog systems, Google Android and data mining.

Antonio Rosario Intilisano is currently a PhD candidate with Telecom Italia Grant, University of Catania. His doctoral work explores the Spoken Language Understanding and Automatic Speech Recognition techniques specifically for Neo-Latin Languages.

Giuseppe Monteleone received his Master Degree in Computer Engineering in 2013 from University of Catania. Currently he is a Contract Researcher at DIEEI, University of Catania. His research interests include Natural Language Processing, Machine Learning and Forecasting Models.

Salvatore Michele Biondi is a Contract researcher at the DIEEI at the University of Catania.
His research interests include spoken dialog systems and datamining and google android.

Ylenia Cilano, is a software engineer at A-Tono. She collaborated with the DIEEI at Univeristy of Catania as contract researcher. Her research interests include spoken dialog systems and morphological engines.