A Grapheme-to-Phoneme Transcription Algorithm Based on the SAMPA Alphabet Extension for the Polish Language

Mikołaj Wypych†**, Emilia Baranowska**‡ **, Grażyna Demenko**‡

† Poznan University of Technology, Poland
‡ Adam Mickiewicz University, Poland
E-mail: {lin|wypych|emiszal}@amu.edu.pl

ABSTRACT

The paper concerns the automatic generation of broad and narrow transcription of the Polish language in the module of a concatenative TTS system under development at Adam Mickiewicz University and Poznan Technical University, Poland. The existing phonetic notations and transcription rules for the Polish language were verified on the basis of: (1) the literature describing the Polish phonological system, (2) the results of the acoustic segmentation for a few hundred utterances produced by 50 speakers. The modified computer-readable SAMPA alphabet was adopted in its basic and in a broadened allophonic version. A rule-based method aided by an additional dictionary of exceptions is thoroughly described. Finally, the article focuses on a rule compiler, a rule applier and dedicated development environment. The resulting G2P module implementation, called PolPhone, is available free for academic purposes.

1. INTRODUCTION

Automatic grapheme-to-phoneme conversion systems (G2P) are intended to convert texts of a given language into phonologically acceptable strings of phonemic/phonetic transcription symbols. The definition of the relation between the orthographic and transcription layer may be obtained on the basis of rules and/or a dictionary. Since Polish is characterized by a relatively regular relationship between the orthographic text and its phonemic/phonetic transcription, it seems that the rule-based approach will be more favorable in its case. Nevertheless, the dictionary part is still needed to cope with exceptions to the rules. It happens quite frequently that both these methods are used concurrently in one module. If a dictionary search fails G2P rules are activated. The transcription rules and dictionaries may be formulated by human experts ([5], [22]) or derived from a corpus of texts and their transcriptions by machine learning algorithms ([21], [22]). In 1973, Steffen-Batogowa published in [5] a set of expert G2P conversion rules for Polish which were implemented for the first time by Warmus [16]. Since then many different G2P modules based on the Steffen-Batogowa's rules have been presented (cf. [9] or [19] for examples and a survey of solutions). The goal of the present project was to build a practical market-oriented G2P module starting from the rules proposed in [5] and [6].

2. EXPRESSING RELATION

2.1. TRANSCRIPTION TABLES

Let X^* denote a set of all finite strings of elements from the set X, including an empty string – the string of the length 0. Let X^+ denote a set of all finite strings from X, excluding an empty string. Let G be a set of elements acceptable in input strings (e.g., a set of Latin letters). Similarly, let P be a set of elements admissible in output strings (e.g. a set of phoneme symbols). A *transcription table* $T[1..m][1..n]$ is a matrix of m rows and n columns in which the cells meet the following set of requirements: $T[1][1]\in G$, $T[i][1]\subset G^+$ for $i\in[2..m]$, $T[1][j]\subset G^+$ for $j\in[2..n]$, $T[2..n][2..m]\in P^*$. Each cell indexed as $(i,j)\in[2..m]\text{x}[2..n]$ states that if $T[1][1]$ in an input string is preceded by a string that belongs to $T[i][1]$ (*the left context set*) and is followed by a string that belongs to $T[1][j]$ (*the right context set*), then $T[1][1]$ should be transcribed as $T[i][j]$. The transcription tables proposed by Batogowa do not contain overlapping contexts. This solution increases the entropy of the context sets. In present solution if certain contexts can be recognized simultaneously, we give a priority to the rules that are formed by longer contexts. As a consequence zero length context can be used to specify default columns or rows. Similarly to [9], we enable more than one variants of transcription for a grapheme in a given context (what results in $T[2..n][2..m]\subset P^*$). Context sets are defined by means of expressions described in [5]. For any $A,B\in G^*$, we define the sum of sets A and B, denoting it by $A+B$, and a concatenation of sets A and B, denoted as $A*B$ which contains any string in the form of $ab\in G^*$, where $a\in A$, $b\in B$ and ab is a result of concatenation of a and b. We use the digit *1* to denote an empty string. For example, {1,con}*{1,cat} produces {1, con, cat, concat}. The rule tables are stored in a UTF8 encoded file in an XML format or in a plain text format. The plain text format uses a semicolon as a column delimiter and a next-line character as a row delimiter. Elements from G and P are represented by strings of Unicode characters. Examples of transcription tables can be found in 3.3.

2.2. DICTIONARY

An *exception* is a matrix $E[1..n][1..3]$ with n rows and 3 columns, where $E[i][1]\in G$, $E[i][2]\in\{0,1\}$ and $E[i][3]\subset P^*$ for $i\in[1..n]$. The exception in the form of $E[1..n][1..3]$ means that if the string $\mathrm{E}[1..n][1]$ matches an input substring then, for each $i\in[1..n]$, if $\mathrm{E}[i][2]$ is equal to 1, $\mathrm{E}[i][1]$ should be transcribed as $\mathrm{E}[i][3]$. Note that there is no need to specify the transcriptions for all the graphemes in an exception but only for the but only for the locations where errors occurred. The remaining graphemes are transcribed according to the rules (see 4.1).

The dictionary is a set of exceptions. Exceptions are stored in a UTF8 encoded file in a XML format or in a plain text format. The plain text formatted file consists of lines, each containing a single exception. Rows of an exception are separated by spaces and saved in one of the two forms: $E[i][1]{:}E[i][3]$ or $E[i][1]$ depending whether $E[i][2]$ is equal to 1 or 0.

3. TRANSCRIPTION RELATION

3.1 INPUT AND OUTPUT CHARACTER SETS

Two sets of characters were precisely defined for the exact G2P mapping for the Polish language – an input set of characters and an output phonetic/phonemic alphabet.

The input set of symbols for Polish was defined here as a set of the following symbols: X={a, ą, b, c, ć, d, e, ę, f, g, h, i, j, k, l, ł, m, n, ń, o, ó, p, q, r, s, ś, t, u, v, w, x, y, z, ź, ż, #, ##}[1]. One hash denotes interword spaces in a string of orthographic symbols and two hashes substitute sentence final punctuation marks. The modified computer-readable SAMPA alphabet, which codes the IPA symbols into the ASCII characters and seems to be a good solution for technical applications, was adopted as an output phonetic alphabet of the module. An inventory of 39 phonemes was employed for broad transcription and a set of 87 allophones was established for narrow transcription of Polish. Y = { # , i , i~ , y , y~ , e , e] , e~ , e]~ , a , a] , a~ , a]~ , o , o] , o~ , o]~ , u , u] , u~ , u]~ , j , j~ , w , w/ , w, , w~ , l , l/ , l, , r , r/ , r, , m , m/ , m, , n , n/ , n, , n- , n' , n'/ , N , N/ , N, , N,/ , v , v, , f , f, , z , z, , z' , s , s, , s' , Z , Z, , S , S, , x , x, , x/ , x,/ , d^z , d^z, , d^z' , t^s , t^s, , t^s' , d^Z , d^Z, , t^S , t^S, , b , b, , p , p, , d , d, , d- , t , t, , t- , g , J , k , c }.

The authors verified the existing phonetic notations and transcription rules for Polish on the basis of: (1) the literature describing the Polish phonological system (cf. e.g. [8], [10], [12]) and (2) the results of the acoustic segmentation for a few hundred utterances produced by 50 speakers brought up in two Polish cities as in [2].

On the basis of present knowledge of Polish phonetics, our own detailed acoustic analyses and experience derived from annotation of Polish speech databases, it was established that the phoneme counterparts of a grapheme *ę* as well as a grapheme *ą* are two separate phonemes not only in the right context of plosives or affricates, but also before fricatives and (partially) when being in the word final position. Thus, it was resigned from original SAMPA symbols /e~/ and /o~/ before fricative consonants /f v s z S Z x/ and when occurring word finally, in favour of biphonemic notation /ew~/, /ow~/ before these consonants and /e/, /ow~/ when word finally. The monophonemic notation /e~/ and /o~/ before palatal fricatives /s' z'/ was changed into two-phoneme transcription /ej~/ or /oj~/ respectively, although the alternative transcription with 'hard' nasal resonance /w~/ is also acceptable.

The adoption of the above-mentioned solution assumes for practical reasons that /w~/ and /j~/ should be treated as individual phonemes in Polish although their phonological status remains unclear [10]. These two phonemes were not taken into consideration in the original version of Polish SAMPA. /c/ and /J/ were also added to the proposed SAMPA extension as the phonemes essential to the description of acoustic differences between velar consonants /k g/ and, respectively, palatal consonants /c J/, as in word pairs: *kat* /kat/ vs. *kit* /cit/ or *drogę* /drogę/ vs. *drogie* /droJje/.

The chart of SAMPA-based coding of Polish phonemes and allophones used in this work can be found at http://main.amu.edu.pl/~fonetyka.

3.2. TRANSCRIPTION TABLES

The tables of rewriting rules presented in [6] were extended to enable transcription which reflects two types of Polish pronunciation: Warsaw (W) and Krakow-Poznan (K-P) ones. The main difference between these two pronunciation varieties concerns voicing in interword assimilations. Alternative transcription was obtained by supplementary rewriting rules and the modification of some subsets of graphemes.

3.3. EXCEPTIONS

On the basis of the error detection procedure, the authors compiled a preliminary lexicon of exceptions. It contained (a) Polish loanwords that kept their original foreign pronunciation and spelling and (b) the words of Polish spelling featuring exceptional pronounciation of some grapheme (e.g. irregular *marznąć* /marznon't^s'/ vs. regular *marzyć* /maZyt^s'/; *zimitować* /zimitovat^s'/ vs. zima /z'ima/). The criteria for entering a word along with its correct transcription into the preliminary version of the dictionary were as follows:

Words that could not be (efficiently) transcribed using the set of G2P rules. New rules would violate the existing ones. Foreign words that were well-grounded in the Polish language. The majority of the listed words are also the entries of the newest dictionary of loanwords in Polish [15]. Some words that where not covered by the Sobol's dictionary were included into the lexicon only on the basis of their newspaper's usage, e.g., e-mail vs. airmail, copywriter vs. copyright. Common nouns. Proper nouns were put in the list only if they appeared in expressions such as choroba Alzheimera 'Alzheimer's disease', skala

[1] Notice that {q, v, x} are not standard Polish graphemes, they appear only in loanwords. It seemed to be more economical later to operate on them in the rules rather than to place words containing them in a module's dictionary.

Beauforta ‘Beaufort scale’ or eau de Cologne. Most loanwords which retained their original spelling and pronunciation in Polish come from English, particularly those relatively new in Polish, e.g., dubbing, hardware, factoring, etc. Although many borrowings had adopted to Polish orthography & pronunciation rules, their original spelling had to be included into the lexicon because its functions in our language alongside, e.g., diler – dealer, interfejs – interface, recykling – recycling, kolaż – collage. Proper names in Polish showing variant pronunciation due to e.g. the insertion of foreign expressions, as it often is in the case of company names and product names, were beyond the scope of the present work. They can be correctly transcribed only with the aid of a large data corpus by lexicon lookup technique (cf. [3]; [1]). As mentioned above, the program takes only ‘repairs’ from the dictionary; the remaining context is generated from the rules.

4. SOFTWARE

The resulting G2P module implementation for Polish, called *PolPhone*, is a set of portable ISO C++[2] source code files. They can be split into two classes: The G2P module skeleton files and the data files that drive transcription process for Polish. The data files are based on sets of rules and dictionaries. The source codes can be compiled using a C++ compiler. As a result, an executable file or a dynamic library is obtained.

4.1. G2P MODULE SKELETON

The term filter denotes an object that gets data from a source stream and puts data into a target stream. A few useful filter types were defined in the described G2P skeleton. A graph of filters connected by streams yields the G2P module skeleton. We use the following types of filters (fig.1):

identity filter copies the input stream to the output stream without any changes to the data,

transcoding filter converts the codepage encoding of streams,

preformatting filter prepares an input text to the form expected by the transcription rules. It includes: downcasing the letters, replacing sequences of blank spaces and digits with a single hash sign, replacing punctuation marks with a double hash sign,

transcribing filter consists of three *transcribers*. The transcriber works on the basis of rules defined in the transcription tables (cf. 2.1 and 4.2). The transcriber consists of two deterministic finite state transducers (FST) and a 2-dimensional decoding matrix. For a given character in the input stream, it applies one FST to preceding chars (its right context) and the another one to the following chars (its left context). The FST produces a single integer if the context is recognized. Decoding matrix cells contain characters hat are placed in the output stream only if both FTS have not failed. The transcriber fails if at least one of the transducers fail or the determined decoding matrix cell contains null. Such a design of the transcriber is enforced by the trade-off between its size and time efficiency. Although building a single deterministic FST that would take over the transcriber’s role for a set of rules in presented project is possible, it would require enormous amount of memory. The transcribing filter can be defined as a chain of three transcribers: An exception transcriber, a rule transcriber and a default transcriber. They are applied to each character of the input stream in the same order that they were itemized in. The subsequent transcriber is executed only if the preceding transcriber fails.

The presented design allows the resulting G2P module to get no more characters from the input stream than it is necessary to obtain a desired number of phonemes (which may require reading a few characters ahead). On the other hand, it is possible to put a given number of characters as input and read all the resulting phonemes until the first that would be dependent on characters that are not available yet. Since the linguistic knowledge incorporated in the skeleton of the G2P is kept at a minimum level, the module is language universal.

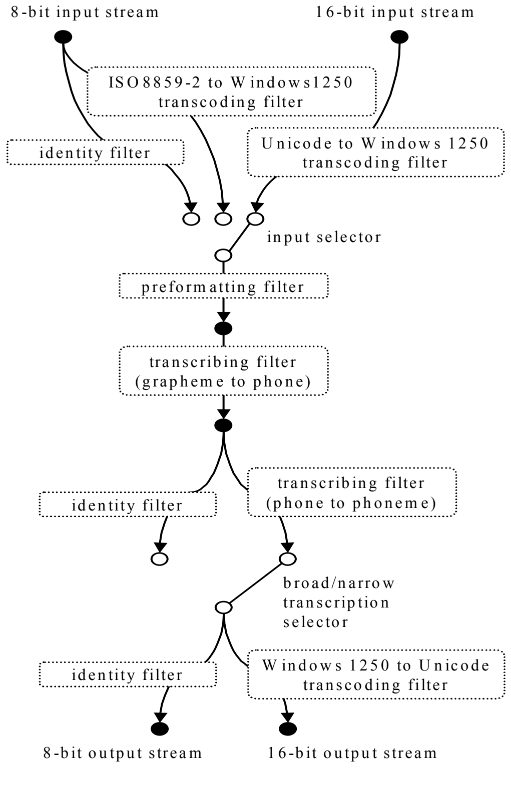
8-bit input stream
16-bit input stream
ISO8859-2 to Windows1250 transcoding filter
identity filter
Unicode to Windows 1250 transcoding filter
input selector
preformatting filter
transcribing filter (grapheme to phone)
identity filter
transcribing filter (phone to phoneme)
broad/narrow transcription selector
identity filter
Windows 1250 to Unicode transcoding filter
8-bit output stream
16-bit output stream

Fig.1. A diagram of G2P skeleton.

4.2. G2P MODULE DATA COMPILATION

The data compilation unit converts a set of files containing the transcription tables and dictionaries into source files supplementing the G2P module skeleton. An editor for rule editing and testing was created. The editor transcribes example texts by means of the defined rules and edits them

[2] Source codes were successfully compiled with Borland C++ Builder 5.0 Update 1 and Microsoft Visual C++ 6.0 under Windows XP system and with GNU C++ 3.2 and Borland Kylix 3 under RedHat Linux 7.3.

ISBN 1-876346-48-5 © 2003 UAB

according to the predefinded rules of transcription and amendments made by experts. The sets of transcription tables are translated into regular expressions and decoding matrices expressed as C++ arrays by means of the program from by Wypych [20]. The FSA6 Finite State Automaton Utilities [20] are used to build, determinize and optimize the transducers from regular expressions. The post-processing stage yields a set of C++ source code files containing arcs of transducers and decoding a matrix suitable for the G2P module skeleton.

5. CONCLUSIONS

The traditional rule-based approach aided by a moderate-sized exception dictionary appears to be a good solution in the G2P module of speech processing systems for the Polish language. The presented implementation aimed at extending the idea of Steffen-Batogowa's transcription rules to achieve maximum effectiveness, i.e. to produce a computationally efficient "knowledge base" giving high quality transcription at low time and memory consumption in a market-ready application.

Acknowledgement

This work was partially supported by research grant 7T11C00921 from the Polish Committee of Scientific Research.

REFERENCES

[1] Demuynck, K., Laureys, T., "Automatic generation of phonetic transcriptions for large speech corpora", www.cnts.uia.ac.be/cnts/pdf/ 20021001.2492.ICSLP2002paper.pdf, 2002.

[2] Demenko, G., Grocholewski, S., "Text independent speaker verification based on segmental and suprasegmental features", in *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics*, SCI: Orlando, 2001.

[3] Łobacz, P., "Problems of automatic phonematic transcription of contemporary Polish proper names", in *Linguam amicabilem facere. Ludvico Zabrocki in memoriam*, Bańczerowski, J. and Zgółka, T. Eds., *Seria Językoznawcza nr 22*, Poznań: UAM, 1999.

[4] Steffen-Batóg, M., "The problem of automatic phonemic transcription of written Polish", *Biuletyn Fonograficzny XIV*, Warszawa – Poznań, 1973.

[5] Steffen-Batogowa, M. *Automatisation of the phonemic transcription of Polish texts*), Warszawa: PWN, 1975.

[6] Steffen-Batóg, M. & Nowakowski, P. "An algorithm for phonetic transcription of orthographic texts in Polish", in *Studia Phonetica Posnaniensia*, vol. 3, Steffen-Batóg, M., Awedyk, W., Eds., Poznań: Wydawnictwo Naukowe UAM, 1993.

[7] Dukiewicz, L., *Polskie głoski nosowe*, (*Polish nasal sounds*), Warszawa: PWN, 1978.

[8] Dukiewicz, L., Sawicka., J., "Fonetyka i fonologia" ("Phonetics and phonology"), in *Gramatyka współczesnego języka polskiego*, Wróbel, H., Ed., Warszawa: PWN, 1995.

[9] Jassem, K., "A phonemic transcription & syllable division rule engine". Onomastica-Copernicus Research Colloquium, Edinburgh, 1996.

[10] Jassem, W., *Podstawy Fonetyki Akustycznej*, (*Fundamentals of Acoustic Phonetics*), Warszawa: PWN, 1973.

[11] Jassem, W., "An Acoustical Linear-Predictive and Statistical Discriminant Analysis of Polish Fricatives and Affricates", *Speech and Language Technology*, vol.2, Poznań: Polish Phonetic Association, 1998.

[12] Madejowa, M., "Zasady współczesnej wymowy polskiej", („Rules of Polish modern pronunciation"), *Biuletyn Audiofonologii*, vol. 2-4, Warszawa: Polski Komitet Audiofonologii, 1989.

[13] Nowak, I., *Automatyczna transkrypcja polszczyzny nieregionalnej (odmiana północno-wschodnia i południowo-zachodnia)*, (*Automatic transcription of the Polish language – north-eastern and south-western varieties*), Warszawa: Prace IPPT PAN, 21/1991.

[14] Ostaszewska, D., Tambor, J., *Fonetyka i fonologia współczesnego języka polskiego*, (*Phonetics and phonology of modern Polish*), Warszawa: Wydawnictwo Naukowe PWN, 2001.

[15] Sobol, E., Ed., *Słownik wyrazów obcych*, Warszawa: Wydawnictwo Naukowe PWN, 2002.

[16] Warmus, M., "Program na maszynę ODRA 1204 dla automatycznej transkrypcji fonematycznej tekstów języka polskiego", in *Zastosowanie maszyn matematycznych do badań nad językiem natyralnym*, Bolc, L. Ed., Warszawa, 1973.

[17] Wells, J., SAMPA homepage http://www.phon.ucl.ac.uk/home/sampa/home.htm.

[18] IPA homepage, ttp://www.arts.gla.ac.uk/IPA/ipa.html.

[19] Wypych, M., "Implementacja algorytmu transkrypcji fonematycznej", in: *Speech and Language Technology*, vol. 3, Poznan, 1999.

[20] van Noord, G., FSA6 homepage http://odur.let.rug.nl/~vannoord/Fsa/

[21] Daelemans, W., van den Bosch, A., "Language-Independent Data-Oriented Grapheme-to-Phoneme Conversion", in: *Progress in Speech Synthesis*, New York: Springer-Verlag, 1997.

[22] Black, A. W., Lenzo, K, Pagel, V., "Issues in Building General Letter to Sound Rules", in: *The Third ESCA Workshop in Speech Synthesis*, 1998.