SSW6

Are Rule-based Syllabification Methods Adequate for Languages with Low Syllabic Complexity? The Case of Italian

Connie R. Adsett and Yannick Marchand

Institute for Biodiagnostics (Atlantic), National Research Council Canada,
1796 Summer Street, Suite 3900,
Halifax, Nova Scotia, Canada B3H 3A7

Faculty of Computer Science, Dalhousie University,
Halifax, Nova Scotia, Canada B3H 1W5

{connie.adsett, yannick.marchand}@nrc-cnrc.gc.ca

Abstract

Syllabification information is a valuable component in speech synthesis systems. Linguistic rule-based methods have been assumed to be the best technique for determining the syllabification of unknown words. This has recently been shown to be incorrect for the English language where data-driven algorithms have been shown to outperform rule-based methods. It may be possible, however, that data-driven methods are only better for languages with complex syllable structures. In this paper, three rule-based automatic syllabification systems are compared and two data-driven (Syllabification by Analogy and the Look-Up Procedure) on a language with lower syllabic complexity - Italian. Using a leave-one-out procedure on 44,720 words, the best data-driven algorithm (Syllabification by Analogy) achieved 97.70% word accuracy while the best rule-based method correctly syllabified 89.77% words. These results show that data-driven methods can also outperform rule-based methods on Italian syllabification, indicating that these may be the best approaches to the syllabification component of speech synthesis systems.

1. Introduction

Automatic syllabification is the process of determining the proper placement of syllable boundaries in a given word. Syllables have been used as key features in text-to-speech (TTS) systems for diverse languages. For instance, knowledge of syllable boundaries in written words is an essential component of a speech synthesis system for generating regional accents in English [1]. In the Hindi language, it has been demonstrated that using syllables as the units for speech synthesis gives better performance than smaller sized units [2]. Syllable structure information is also used in Czech [3], [4], European Portuguese [5], German [6], [7], Italian [8], [9] and Romanian [10] TTS systems.

Rule-based approaches have traditionally been the preferred method of determining the syllabification of unknown words (for example, see [2], [5], [7], [8] and [10]). Once defined by linguists, rules are straightforward to implement and apply. However, this approach is often time-consuming and requires expert knowledge. More importantly, several studies have raised the question of whether rule-based methods are actually the best approach to these tasks given the high performance of data-driven methods [11]–[14]. In particular, it has been demonstrated that, for syllabification of English words, data-driven methods perform significantly better than rule-based methods [14]. The success of data-driven methods on this language may be due to the fact that English is a language with a complex and irregular syllable structure [15], [16], which is challenging (and perhaps impossible) to fully capture with traditional linguistic rules.

Finnish, Italian and Spanish are considered to exhibit simple syllabic structure [16]–[20]. In this study, we performed an evaluation of syllabification methods on a language with simpler syllabic structure. The Italian language was selected to compare rule-based and data-driven methods for automatic syllabification in the same manner as these approaches were compared for English. This language was chosen because of the availability of both a large lexicon of syllabified Italian words and several rule-based algorithms for automatic syllabification in Italian.

2. Italian lexicon

The Italian lexicon used in this paper is part of the Italian Festival TTS project [21]. It was created by ITC-irst (Instituto Trentino di Cultura - Instituto per la Ricerca Scientifica e Tecnologio) and ISTC-SPFD CNR (Instituto di Scienza e Tecnologi della Cognizione - Sezione di Padova "Fonetica e Dialettologia" - Consiglio Nationale della Ricerche) and is freely available at `www.pd.istc.cnr.it/Software/It-Festival/2.0/lex_ifd.zip` (last accessed 9 March 2007).

Each entry provides spelling, part-of-speech, pronunciation, stress, and syllabification information. A total of 440,084 entries exist in the original lexicon. Because there is such similarity in Italian between different forms of the same words, we endeavoured to reduce the lexicon to only one form of each word. For this reason, proper nouns, plurals, verb forms (apart from the infinitive), superlatives and comparatives, and homophones and homographs were removed from the lexicon.

Because syllabification information was given in the pronunciation domain and the three Italian rule-based syllabification algorithms operate on the spelling domain (see Section 3), all words which did not have the same number of letters as phonemes were also removed. This allowed syllable boundaries to be transferred to the spelling domain without any need of complex alignment processing. Using this simple alignment approach, only 8,697 words were removed. The resulting lexicon (referred to as the *Full* Italian lexicon below) consisted of

44,720 entries, which formed the basis for all experiments unless otherwise stated.

3. Rule-based algorithms

Three Italian-specific rule-based algorithms for automatic syllabification were tested. The algorithms selected were: Cioni's algorithm for the syllabification of written Italian [22], an implementation of Hall's ordered rules for Italian syllabification [23], and Bergamini's SYL-LABE syllabification algorithm [24]. None of these methods have been tested or compared in order to determine accuracy. All three algorithms operate on the spelling domain.

3.1. Cioni's algorithm for the syllabification of written Italian

Cioni [22] presents a method using what he claims to be a "minimal set of rules" developed with the assistance of Italian linguists. The source code (written in C) for this program is available from `www.di.unipi.it/~lcioni/AltroSoftware/sillabatore.tar.gz` (last accessed 9 February 2007).

A subset of these rules is provided by Cioni [22] and is listed below (where C denotes a consonant and V denotes a vowel):

1. CVCV $\rightarrow$ CV | CV;
2. $VC_1C_2V \rightarrow VC_1 | C_2V$, if $C_1 = C_2$;
3. $VC_1C_2V \rightarrow V|C_1C_2V$, if $C_2 = h$;
4. $VC_1C_2V \rightarrow V|C_1C_2V$ or $VC_1|C_2V$, if $C_1 = s$ and depending on the value of C_2;
5. VCCV $\rightarrow$ VC|CV;
6. $VC_1C_2C_3V \rightarrow VC_1|C_2C_3V$, if $C_1 \neq s$;
7. $VC_1C_2C_3V \rightarrow V|C_1C_2C_3V$ or $VC_1|C_2C_3V$, if $C_1 = s$ and depending on the values of C_2 and C_3;
8. VCCCCV $\rightarrow$ VCC|CCV in most cases;
9. $V_1V_2 \rightarrow V_1|V_2$, if $V_1 \in$ {a,e,o} and $V_2 \in$ {a,e,o};
10. $V_1V_2V_3V_4 \rightarrow V_1V_2V_3|V_4$, if $V_1V_2V_3$ is a triphthong;
11. $V_1V_2V_3 \rightarrow V_1|V_2V_3$ if $V_1 \in$ {a,e,o};
12. $V_1V_2V_3 \rightarrow V_1V_2|V_3$, if $V_1 = i$ and $V_2 \neq u$, or $V_1 = u$ and $V_2 \neq i$.

Rules are also included to specify which pairs of vowels form diphthongs and therefore cannot be separated into different syllables. All rules are applied recursively by searching through the given word from left to right.

3.2. Hall's ordered rules for Italian syllabification

Hall [23] lists six ordered rules for breaking single Italian words into syllables:

1. $C_1C_2 \rightarrow C_1|C_2$, if $C_1 = C_2$;
2. $C_1C_2 \rightarrow C_1|C_2$, if $C_1 = c$ and $C_2 = q$;
3. $C_1C_2 \rightarrow C_1|C_2$, if $C_1 \in$ {m,n,r,l};
4. VCC $\rightarrow$ V|CC;
5. VCV $\rightarrow$ V|CV;
6. never divide a sequence of vowels into multiple syllables.

He provides two additional rules for division across word boundaries. The first states that a syllable boundary should never be placed immediately following an apostrophe which connect two words; for example, "l'albero" (*the tree*). The second concerns placement of syllables in musical scores. In this environment when a final syllable in a word ends in a vowel and the next word begins with a vowel and they must be both sung on the same note or over tied notes, it is necessary to indicate that they form a single syllable. This is done by decreasing the space between the two syllables.

These rules are given with the intent of assisting Italian instructors in teaching students how to divide Italian words in the spelling domain. Because they are fully described, it was possible to implement a rule-based automatic syllabification program[1] using these rules for the purpose of evaluation.

3.3. Bergamini's SYL-LABE program

Bergamini's rule-based syllabification algorithm is called SYL-LABE and was implemented in C (available at `http://www.pierotofy.it/pages/sorgenti/C/Utility/`[2] - last accessed 15 March 2007). The results of this algorithm were used as a gold standard in work on the automatic syllabification of Italian [24]. Two versions (1.0 and 3.3) of the algorithm are available and both were tested but only the results of the best algorithm (version 1.0) are reported. Implementation details are given in an Italian file which accompanies the download of version 3.3 of the program. One sample rule used in this system is VCV $\rightarrow$ V | CV.

The SYL-LABE program, as it was originally built, syllabifies only one word each time it runs. In addition to the word itself, stress information is required by the SYL-LABE algorithm in order to obtain syllabified output. Stress information was provided as given in the Italian lexicon. For example, to syllabify the word "sempre" (*always*), the input required is: `sempre` and `2` at the prompts given, where 2 is the location in the input string of the vowel in the stressed syllable. A simple loop program was written in order to obtain the syllabification of a list of words using the SYL-LABE program.

4. Data-driven algorithms

The data-driven algorithms used in this comparison were the same two that performed best on the syllabification of English words [14]: Syllabification by Analogy (SbA) and the Look-Up Procedure.

4.1. Syllabification by Analogy

Syllabification by Analogy is adapted from Pronunciation by Analogy, a method for automatic grapheme-to-phoneme transcription [25]–[28].

To compute the syllabification of an unknown word, it is first broken into substrings. Comparing these to substrings of syllabified words in the lexicon determines each segment's syllabic structure. This information is then used to construct the syllabification of the entire word.

Matching substrings are found by comparing the input word to all words in the dictionary. For each entry, the initial character in the input string is aligned with the final character in the syllabified word. The input string is then shifted left un-

[1] A Python implementation of Hall's rules is available from the authors upon request.

[2] The program may not be listed on the first page of the site.

til its final character is aligned with the initial character of the syllabified word. Before each shift all aligned characters are checked to determine whether there are any matching substrings at this point. If a match is found, syllabification information for this substring (obtained from the syllabified dictionary entry) is stored in a syllabification lattice. Using this procedure, the input word is compared to all words in the dictionary.

For example, the word "able" is represented as the input string `a?b?l?e` where `?` represents each position between letters (juncture) at which a syllable boundary may occur. The syllabified word "syl | la | ble" is represented as `s*y*l|l*a|b*l*e` where `*` and `|` represent non-syllable and syllable boundaries, respectively. When compared to find matching substrings, `?` may be matched with either `|` or `*`. Using "able" as the unknown word and "syllable" as the word from the lexicon, the matching process for them is shown in Table 1.

Step	Matching Procedure
1	`a?b?l?e`
	`s*y*l|l*a|b*l*e`
7	`a?b?l?e`
	`s*y*l|l*a|b*l*e`
Final	`a?b?l?e`
	`s*y*l|l*a|b*l*e`

Table 1: *Example matching process using "syllable" (an entry in the lexicon) to syllabify "able" (an unknown word).*

The resulting syllabification lattice is a graph for which information from matching substrings form the nodes and arcs. Nodes represent the beginning and ending substring characters. Arcs are labeled with any intermediate substring characters along with the number of occurrences of this substring within the matches found in the dictionary. In the case of the above example, the nodes and arc inserted into the lattice from the substring `a|b*l*e` (found in step seven) would be $\bullet_a \xrightarrow{|b*l*:1} \bullet_e$, along with all other subelements of this substring (for example, $\bullet_a \xrightarrow{|:1} \bullet_b$ for `a|b` and $\bullet_* \xrightarrow{l*:1} \bullet_e$ for `*l*e`).

A decision function is used to find the all possible shortest paths through the lattice from the first to the last character of the input word. Syllabification is obtained from a given path by concatenating the node and arc labels (aside from the frequencies). If only one shortest path is found it is used to infer the syllabification of the unknown word.

When two or more shortest paths exist, a set of scoring strategies are used to determine the best syllabification. The three scoring strategies that gave the highest performance on the English language [29] were:

1. the maximum product of the arc frequencies along the shortest path;
2. the maximum frequency of the same syllabification within the shortest paths;
3. the maximum weak link value where 'weak link' is the minimum of the arc frequencies.

For the sake of consistency, these same scoring strategies were used to determine the syllabification of Italian words.

4.2. Look-Up Procedure

The Look-Up Procedure was also originally used for grapheme-to-phoneme transcription [30]. It has since been modified to perform automatic syllabification [11], [14]. This method uses N-grams (each consisting of a left context, right context and central letter) to learn and determine syllable boundaries.

During training, an N-gram is generated for each possible syllable boundary location in a word. Each N-gram is stored in a table along with how often a syllable occurs and does not occur following the central letter. Table 2 shows the table entries for the word `s*y*l|l*a|b*l*e`, using a left and right context of three letters (N = 7).

	Frequencies	
N-grams	|	*
`---syll`	0	1
`--sylla`	0	1
`-syllab`	1	0
`syllabl`	0	1
`yllable`	1	0
`llable-`	0	1
`lable--`	0	1

Table 2: *Table entries for the word "s*y*l | l*a | b*l*e" used by the Look-Up Procedure.*

During testing, the closest matches to the N-grams from the test words are found in the table. Similarity between N-grams is determined using an N-element weight vector. For a given N-gram, if the frequency of a syllable boundary occurring after the central letter is higher than the frequency of no syllable boundary, a syllable boundary is placed in the test word.

For example, using the 7-grams stored in Table 2 to syllabify the word "able" requires finding the closest match to each of four 7-grams (`---able`, `--able-` and `-able--`) within the table. Using $[1, 4, 16, 64, 16, 4, 1]$ as the weight vector, the closest match to the first 7-gram in given above (`---able`) is the entry `yllable` with a similarity value of 85 $(64 + 16 + 4 + 1)$. The frequency of a syllable boundary occurring after the central letter in this pattern is greater than the frequency of no syllable boundary and therefore a syllable boundary is placed following the `a` in "able".

The Look-Up Procedure was tested with all 15 weight vectors (given in Table 3) that were used in the comparison of automatic syllabification methods for English [14] and the study in which this technique was originally described [30].

5. Results

To compare the syllabification algorithms described above, a leave-one-out procedure was used whereby each word was removed from the lexicon in turn and its syllabification was inferred from all other words.

Results were computed using word and juncture accuracy. Word accuracy is simply the number of words syllabified by the method in exactly the same way as is given by the standard used (in this case, the Italian lexicon). Juncture accuracy compares syllabification at the sub-word level. Each position between letters is assessed to determine whether it was classified correctly. For example, the Italian word "sempre" has five junctures (denoted by a '`?`') and can be shown as "`s?e?m?p?r?e`". The accepted syllabification, according to the Italian lexicon, is "sem | pre". If an algorithm syllabifies the word as "semp | re", this is considered entirely wrong in terms of word accuracy, however it is 60% (3/5) correct in terms of juncture accuracy, as shown in Table 4 in which C and I correspond to correctly and incorrectly syllabified junctures.

Version	Left Context				Central Letter	Right Context				
	-4	-3	-2	-1	0	+1	+2	+3	+4	+5
1					1					
2				1	4					
3					4	1				
4				1	4	1				
5			1	4	16	4				
6				4	16	4	1			
7			1	4	16	4	1			
8			1	4	16	4	2			
9		1	4	16	64	16	4	1		
10		1	4	16	64	16	5	1		
11	1	4	16	64	256	64	17	4		
12		4	16	64	256	64	16	4	1	
13		4	16	64	256	64	17	4	1	
14		16	64	256	1024	256	64	16	4	1
15		16	64	256	1024	256	65	16	4	1

Table 3: *Weight vectors used in the Look-Up Procedure.*

	Syllabification of "sempre"
Output	s * e * m | p * r * e
Lexicon	s * e * m * p | r * e
Junctures	C C I I C

Table 4: *Juncture accuracy example.*

The results for all automatic syllabification algorithms are presented in Table 5. Although all 15 weight sets were used for the Look-Up Procedure, only the top five are reported. The difference in performance between the best rule-based method (SYL-LABE) and the best data-driven method (SbA) is approximately 10%. A Chi-squared test (χ^2) reveals $\chi^2_{obt} = 2977.0$ for words and $\chi^2_{obt} = 5030.3$ for junctures. These differences between SbA and SYL-LABE are highly statistically significant ($p < 0.01$ in both word and juncture accuracy).

Discrepancies in performance amongst the rule-based methods are attributed to differences in the rule sets used by each. Although some rules are consistent between methods, others are vastly different. For example, Hall's rules [23] state that no vowel cluster should ever be separated by a syllable boundary while Cioni [22] states that when the vowels 'a', 'e', and 'o' are adjacent within a word (e.g. 'ae' or 'eo'), they are not in the same syllable. Simple analysis of the lexicon reveals that, for bigrams involving the vowels 'a', 'e', and 'o', Hall's rule is nearly always wrong while Cioni's rule is often correct.

	Percentage Correct	
Algorithm	Word	Juncture
Cioni	86.59	97.78
Hall	81.59	97.24
SYL-LABE	89.77	97.89
Syllabification by Analogy	97.70	99.67
Look-up Procedure		
version 10	96.43	99.54
version 11	96.04	99.49
version 8	96.02	99.49
version 13	95.93	99.48
version 15	95.82	99.46

Table 5: *Syllabification results on the* Full *Lexicon.*

In addition, the overall results of all methods appear to be better in Italian than syllabification results in English, previously reported in [14]. This could be due to the fact that the *Full* Italian lexicon contained many more words than any of the lexicons used when comparing syllabification methods in English.

Because the correct syllable boundaries in a word are sometimes disputed, the English comparison used three lexicons: one from *Webster's Pocket Dictionary* (19,596 entries), another from the *Wordsmyth English Dictionary-Thesaurus* (18,016 entries), and a third (called the *Overlap* database) which consisted of the 13,594 words with the same syllabification in both of the other lexicons [14]. To determine whether Italian is indeed easier to syllabify automatically than English, a randomized reduced set of the *Full* Italian entries, which matched *Overlap* lexicon size and, as closely as possible, the word length distribution was selected. The resulting *Reduced* Italian database also consisted of 13,594 entries.

	Words Correct (%)		
Algorithm	Italian	English	χ^2_{obt}
SbA	95.33	85.43	764.0
Look-up Procedure	91.60	73.53	1541.6
Rule-based	89.77	36.88	8186.0

Table 6: *Comparison of syllabification results for* Overlap *English [14] and* Reduced *Italian lexicons.*

The performance of the data-driven methods on the *Reduced* Lexicon was slightly less than on the *Full* Lexicon, as would be expected given that significantly fewer words were provided for training. However, the difference between the best data-driven algorithm (SbA - 95.33% for words) and best rule-based method (SYL-LABE - 89.77% for words) is still significant ($\chi^2_{obt} = 563.3, p < 0.01$).

Furthermore, from a computational perspective, these results quantitatively confirm linguistic and psychological findings, which indicate that Italian is simpler and more consistent in syllable structure than English, as stated by [17], [20], [31] and [32]. Although the CV syllable has been found to be most common in both English and Italian, this constituted only 34% of the syllables in English [17] as opposed to a full 60% in

Italian [31]. Such a marked difference should result in Italian being easier to syllabify than English. Table 6 compares the results for the Italian and English [14] languages, showing that, for SbA, the best Look-Up Procedure weights (version 10 for both languages) and the best rule-based methods in each of the two languages, the results obtained for Italian are significantly higher ($p < 0.01$). Indeed, although data-driven methods provide the most accurate results for Italian, rule-based methods still perform well with the best word accuracy at a full 89.77% in comparison to the poor performance of rules on English syllabification.

6. Conclusion

Previous studies show that data-driven methods outperform rule-driven methods in English syllabification tasks [14]. The purpose of this study was to extend the comparison of these two approaches to the syllabification of a language known to have lower syllabic complexity, namely Italian.

When the results from Italian syllabification methods are compared to those from English, it is evident that, regardless of the method used, performance on the Italian lexicon is significantly better. This indicates that syllabification must be a more straightforward task in Italian which is not surprising due to the fact that Italian exhibits lower syllabic complexity.

This comparison on a set of about 44,000 Italian words also confirms the superiority of the data-driven algorithms in terms of both word and juncture accuracy. Overall, all the algorithms presented attain at least 80% word accuracy. The best data-driven method (SbA) reaches a word accuracy of 97.70%, whereas the best rule-based method (SYL-LABE) achieves 89.77%.

In conclusion, these results suggest that, when a syllabification procedure is included as a component of a TTS system, a data-driven method is a more appropriate choice than a rule-based approach, even for languages with low syllabic complexity.

7. Acknowledgements

The authors wish to thank Piero Cosi for providing help with the Italian lexicon. This work was supported in part by funding from the Natural Sciences and Engineering Research Council of Canada (NSERC). In addition, the first author was funded by the National Research Council (NRC) Graduate Student Scholarship Supplement Program (GSSSP), and an Izaak Walton Killam Predoctoral Scholarship.

8. References

[1] S. Fitt, "The generation of regional pronunciations of English for speech synthesis," in *Proceedings of Eurospeech 1997*, 1997, pp. 2447–2450.

[2] S. P. Kishore and A. W. Black, "Unit size in unit selection speech synthesis," in *Proceedings of Eurospeech 2003*, 2003, pp. 1317–1320.

[3] I. Kopecek, "Syllable based speech synthesis," in *Proceedings of the 2nd International Workshop on Speech and Computer (SPECOM '97)*, 1997, pp. 161–165.

[4] I. Kopecek, "Automatic segmentation into syllable segments," in *Proceedings of the First International Conference on Language Resources and Evaluation*, 1998, pp. 1275–1279.

[5] C. Oliveira, L. C. Moutinho, and A. Teixeira, "On European Portuguese automatic syllabification," in *Proceedings of Interspeech 2005*, 2005, pp. 2933–2936.

[6] B. Möbius, "Word and syllable models for German text-to-speech synthesis," in *Proceedings of the Third ESCA/COCOSDA Workshop on Speech Synthesis (SSW3-1998)*, 1998, pp. 59–64.

[7] M. Libossek and F. Schiel, "Syllable-based text-to-phoneme conversion for German," in *Proceedings of the Sixth International Conference on Spoken Language Processing (ICSLP-2000)*, 2000, pp. 283–286.

[8] P. B. de Mareüil, "Linguistic-prosodic processing for text-to-speech synthesis in Italian," in *Proceedings of the Sixth International Conference on Spoken Language Processing (ICSLP-2000)*, volume 1, 2000 pp. 697–700.

[9] M. Refice and M. Savino, "Automatic grapheme-to-phoneme conversion for Italian," *Archives of Control Sciences*, vol. 15, pp. 415–428, 2005.

[10] D. Burileanu and C. Negrescu, "Prosody modelling for an embedded TTS system implementation," presented at the 14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, 2006.

[11] W. Daelemans and A. van den Bosch, "Generalisation performance of backpropagation learning on a syllabification task," in *Proceedings of Twlt3: Connectionism and Natural Language Processing*, 1992, pp. 27–37.

[12] A. van den Bosch and W. Daelemans, "Data-oriented methods for grapheme-to-phoneme conversion," in *Proceedings of EACL '93*, 1993, pp. 45–53.

[13] R. I. Damper, Y. Marchand, M. J. Adamson and K. Gustafson, "Evaluating the pronunciation component of text-to-speech systems for English: A performance comparison of different approaches," *Computer Speech and Language*, vol. 13 pp. 155–176, April 1999.

[14] Y. Marchand, C. R. Adsett, and R. I. Damper, "Automatic syllabification in English: A comparison of different algorithms," *Language and Speech*, to be published.

[15] M. Bruck, R. Treiman, and M. Caravolas, 1995. "Role of the syllable in the processing of spoken English: Evidence from a nonword comparison task," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 21, pp. 469–479, June 1995.

[16] M. Conrad and A. M. Jacobs, "Replicating syllable frequency effects in Spanish in German: One more challenge to computational models of visual word recognition," *Language and Cognitive Processes*, vol. 19, pp. 369–390, March 2004.

[17] R. M. Dauer, "Stress-timing and syllable-timing reanalyzed," *Journal of Phonetics*, vol. 11, pp. 51–62, 1983.

[18] P. M. Bertinetto, L. Cioni, and M. Agonigi, 1994. "The hierarchical structure of the syllable in Italian (Or: Does one really start from the beginning?)," *Quaderni del Laboratorio di Linguistica*, vol. 8, pp. 1–21, 1994.

[19] C. J. Alvarez, M. Carreiras, and Marcus Taft, "Syllables and morphemes: Contrasting frequency effects in Spanish," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 27 pp. 545–555, March 2001.

[20] P. H. K. Seymour, M. Aro, and J. M. Erskine, "Foundation literacy acquisition in European orthographies," *British Journal of Psychology*, vol. 94, pp. 143–174, May 2003.

[21] P. Cosi, F. Tesser, R. Gretter, and C. Avesani, "Festival speaks Italian!" in *Proceedings of Eurospeech 2001*, 2001, pp. 509–512.

[22] L. Cioni, "An algorithm for the syllabification of written Italian," in *Proceedings of the 5th International Symposium on Social Communication*, 1997, pp. 22–24.

[23] R. A. Hall Jr., "Ordered rules for Italian syllabification," *Italica*, vol. 51, pp. 305–307, Autumn 1974.

[24] R. MacKinney-Romero and J. Goddard, 2006. "Syllabification using decision trees, early results on three languages," in *3er Taller de Tecnologías del Lenguaje Humano*, 2006, pp. 282–287.

[25] M. J. Dedina and H. C. Nusbaum, 1991. "PRONOUNCE: A program for pronunciation by analogy," *Computer Speech and Language*, vol. 5, pp. 55–64, January 1991.

[26] F. Yvon, "Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks," in *Proceedings of Conference on New Methods in Natural Language Processing (NeMLaP-2'96)*, 1996, pp. 218–228.

[27] R. I. Damper and J. F. G. Eastmond, "Pronunciation by Analogy: Impact of implementation choices on performance," *Language and Speech*, vol. 40, pp. 1–23, March 1997.

[28] Y. Marchand and R. I. Damper, "A multistrategy approach to improving pronunciation by analogy," *Computational Linguistics*, vol. 26, pp. 195–219, June 2000.

[29] Y. Marchand and R. I. Damper, "Can syllabification improve pronunciation by analogy of English?" *Natural Language Engineering*, vol. 13, pp. 1–24, March 2007.

[30] A. J. M. M. Weijters, "A simple look-up procedure superior to NETtalk," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN-91), volume 2*, 1991, pp. 1645–1648.

[31] S. Frota and M. Vigário, "On the correlates of rhythmic distinctions: The European/Brazilian Portuguese case," *Probus*, vol. 13, pp. 247–275, 2001.

[32] C. Burani, L. Barca, and A. W. Ellis, "Orthographic complexity and word naming in Italian: Some words are more transparent than others," *Psychonomic Bulletin & Review*, vol. 13, pp. 346–352, April 2006.