Sylli: Automatic Phonological Syllabification for Italian

Luca Iacoponi[1], *Renata Savy*[2]

[1]Department of Linguistics, University of Pisa, Italy
[2]Department of Humanities, University of Salerno, Italy
jacoponi@gmail.com, rsavy@unisa.it

Abstract

We will present a complete syllabifier for Italian (Sylli), that is based on phonological principles, flexible and easy to adapt for other uses, alphabets and languages. Crucial concepts regarding syllabification principles in modern phonological theory will be discussed (§1.1); specific issues concerning Italian syllabification will then be summarised (§1.2) and an overview of the available automatic syllabification models will be provided (§1.3). We will then move on to describe the program structure, the syllabification algorithm and two particular issues concerning syllabification in Italian (§2). Finally, we will illustrate the results of a manual syllabification test carried out by linguists to verify the accuracy of the algorithm (§3).
Index Terms: automatic syllabification, syllabifier, syllable, Italian

10.21437/Interspeech.2011-265

1. Introduction

The aim of this paper is to present a complete syllabifier for Italian that, unlike other available options, can be founded on phonological principles and architectural constraints and could therefore be flexible and easy to adapt for other uses, alphabets and languages. In order to describe the internal structure and operational rules of our system, we will briefly outline some basic phonological concepts regarding the syllable and syllabification were implemented in the algorithm[1].

1.1. Syllabification Principles

In theoretical linguistics, the syllable has almost always been considered an essential phonological unit with respect to prosody, phonotactics and phonological processing. The role of the syllable in modern linguistic theory has been central especially since the rise of non-linear phonologies in the eighties, and became even more prominent and controversial with the development of later theories based on the autosegmental framework [1] as well as in Optimality Theory (OT) [2].

In speech technologies, the syllable began taking ground a decade after theoretical phonology. Syllabic units have been used in place of individual segments or sub-segments in speech recognition (SR) and text-to-speech systems (TTS) and as feeding units in Artificial Neural Networks (ANNs) [3][4][5].

The first syllabification principle ever recognised was the Sonority Sequencing Principle (SSP) [6]. The SSP is based on the Sonority Hierarchy (SH), which ranks segments by their intrinsic sonority; the SSP states that between any member of a syllable and the syllable peak, only sounds of higher sonority rank are permitted [7]. Sonority Distance principles (SD) are based on the SSP and define a minimum threshold in sonority differences [8]. A consonant cluster violates the sonority principle and is heterosyllabic only if the difference α between two segments is minor than a language-specific threshold. VCCxV will be syllabified as VC.Cx if the sonority of $C1 \geq \alpha$, V.CCxV if $C1 - C2 > \alpha$. The SSP and the SD have changed form, and adapted with respect to the phonological theory they were borrowed into.

Another group of syllabification principles relates to the phonotactics of languages. Possible codas and onsets are defined as possible word-initial or word-final clusters [9]. This principle is based on two assumptions: 1) that only a medial cluster that could be analysed as a word-final followed by word-initial cluster exists in language and 2) that the speaker's intuition tends to identify units that match phonotactic constraints. The SSP is combined with the Maximum Onset Principle (MOP) [9], which regulates the distribution of ambiguous intervocalic clusters. In a sequence VCCV the application of the MOP results in V.CCV if CCV is a possible word-initial cluster or VC.CV.

1.2. Syllabification in Italian

Italian is one of the languages which is claimed to respect the SSP (or its possible variants). The literature on Italian syllabification has focussed on two issues: the sC clusters and the status of vowels. It is claimed that sC clusters in Italian (and universally) are heterosyllabic [10], but some authors argue that they can either be tautosyllabic or that they at least may show some degree of variation [11]. For example, in the word *pasta*, two syllabifications are possible: the heterosyllabic '*pas.ta*' and the tautosyllabic '*pa.sta*'. Accepting the heterosyllabic proposal, forces the first segment to be extra-syllabic (or preceded by an empty nucleus) in words such as *stella* 'star' , which then gives the syllabification '*s.tel.la*'.

The other problem of Italian syllabification is the direct consequence of the status of glides. It is argued that glides are not distinctive in Italian [12], but are instead derived from the lenition of velar and palatal high vowels; such hypothesis can be confuted on the ground of internal evidence. There are in fact cases of surface glides in stressed syllables, where vowel lenition should not occur, as in *chiocciola* ['kjɔttʃola] 'snail' [13]. It will then have to be assumed that glides can be either derived or underlying, but as we will show this distinction has no consequence on the syllabification algorithm.

1.3. Automatic Syllabification

In the literature a distinction is made between rule-based and data-driven syllabifiers [14]. Since most current phonological theories criticise the use of rules in favour of rich representations or constraints, we mantain that it is no longer accurate to refer to these models in such terms. We here propose to dis-

[1]R. Savy bears responsibility for §1, L. Iacoponi for §2, §3 and §4.

Copyright © 2011 ISCA

tinguish between data-driven and algorithmic approaches; algorithmic approaches have developed along the lines of theoretical linguistic research. Recently, many syllabification algorithms have been developed in the OT framework. One of these [15] evaluates the candidate set using a classical set of constraints (PARSE $\gg$ NOONSET $\gg$ ONSET) by assigning four candidates to each segment: onset, coda, nucleus and unsyllabified. Offending candidates are then excluded from the set by a non-linear evaluation procedure. Other proposals include Fisher's algorithm[2] (based on Kahn's hypothesis) and all algorithms that parse the input and apply a rich set of language-specific rules [16][17].

For data-driven models, many ANNs have been used for syllable division tasks, such as generic neural algorithm [18], dynamic systems [19] and recursive networks [4]. Data-driven approaches not using ANNs include look-up procedures, syllabification by analogy [14] and exemplar-based generalisation techniques [3].

An SSP based solution designed specifically for Italian can be considered a hybrid that combines the SSP, the MOP and an exception-handling mechanism [20]. This algorithm parses a string, find the least sonorous segment and for each sonorous segment adds a syllable boundary after the segment (if it is a sonorant, otherwise it adds the boundary before it). Another algorithmic approach is a rule listing algorithm [21]. A binary decision tree parses the input string from left to right and decides whether a string is heterosyllabic or tautosyllabic by means of matching rules. Data-driven models for Italian have been used to explore particular phonotactic phenomena. The basic assumption is that syllabification is governed by the speaker's phonotactic competence. Clusters which occur together more frequently hold a stronger attraction and therefore are tautosyllabic. The syllable is then composed of groups of segments which are strongly attracted to one another [22].

2. Sylli

The Sylli[3] algorithm was designed as a modular system, with one simple, fast, specific and universal module made up of the Syllabification Algorithm (SA henceforth), other computational satellites which provide input-output specific processing (transducers) and a static language-specific vocabulary. Differences in syllabification are managed by changes in both the transducers or the SH. This architecture also has the advantage of an implementation that is flexible and easy to adapt to different alphabets and languages. We will first describe the structure of the system and provide a description of its components (§2.1); then, the syllabification algorithm is presented (§2.2).

2.1. System Structure

The system is composed of two transducers (one for the input and one for the output), the syllabification algorithm and the mapping list (i.e., the vocabulary). The two transducers convert the two-dimensional linear input to a three-dimensional phonological form that is necessary for the processing in the phonological module and then sends the phonological form back into a linear string for output printing. The transducers can be thought of as a modular translator, that allows the phonological module to operate only on a closed set of symbols, a language-specific vocabulary, and the output to get rid of all phonological information necessary for phonological processing. The transducers only define the mapping between two forms, and can therefore perform only two operations: *Translate*, which transforms a symbol into a phonological item, an output form, or to zero; *Send*, which sends the translated content to the syllabification module or to the output.

The transducers use vocabularies, which map input forms A to objects B; for the purpose of syllabification, the vocabulary only contains the phonological form, the sonority of the segment and its natural class. For example, the mapping entry for the segment /a/ is the following: *a = a, 22, V*. Vocabularies are specified in a configuration file and they can be switched between, edited or created using the GUI or a text editor.

2.2. Syllabification Algorithm

The SA is an improvement on another SSP-based algorithm developed for Italian [20]; it implements the SSP without segment-specific exceptions and account for the problem of sC and vowel clusters. The syllabification algorithm is very simple: if there is a minimum sonority peak, put a syllable boundary (Algorithm 1). Segment sonority attributes are compared in a left-to-right fashion. The string to syllabify only consists of an array of phonological objects as specified in the transducer vocabulary.

To summarise this process, first the transducer parses the input, that is converted using the vocabulary until a sequence that triggers *Send* is found. Then the transducer sends the sequence to the SA, where it gets syllabified. Finally, the syllabified sequence sent to the output transducers, which converts the output into the appropriate form for printing.

Algorithm 1 The syllabification algorithm

Require: list of segments
1: **for all** segments **do**
2: $son \leftarrow sonority(segment)$
3: **if** son is minimum peak OR son = sonority(segment-1) **then**
4: put a syllable boundary
5: **end if**
6: **end for**
7: **return** list of syllable boundaries

The particular SH we used (see table 1) differs from those found in the literature in two aspects: the sibilant has a particular sonority; and the sonority classes differ from the traditional natural phonological classes. No further restriction, constraint or repair mechanism are needed, thus accomplishing our goal to keep the syllabification algorithm general and simple. A demonstration of the application of the algorithm is shown in table 2, where a set of clusters representative of Italian phonotactics have been syllabified by the algorithm[4].

In section 1 we discussed two cases when syllabification in Italian might be controversial. Vowel clusters and sC clusters are an example of how Sylli can handle variation in syllabification through changes made solely the SH.

With the sC cluster, the sonority of the sibilant was changed from fricatives (2) to sonorants (3); that was sufficient to predict the heterosyllabic interpretation and the extrasyllabic segment as theorised in phonology. The output is then: *pasta* $\rightarrow$ '*pas.ta*'; *stella* $\rightarrow$ '*s.tel.la*'. On the other hand, tautosyllabic clusters are

[2] ftp://jaguar.ncsl.nist.gov/pub/tsylb2-1.1.tar.Z (last accessed 2 February 2011)

[3] http://sylli.sf.net. The code is released under the Apache 2 licence and it is free to download, use and redistribute.

[4] Transcrips in test data and in table 2 are in SAMPA alphabet [23].

Table 1: *Sonority Hierarchy for Italian.*

Class	Sonority
Vowels	5
Glides	4
Sonorants + /s/	3
Fricatives	2
Stops	1

Table 2: *Syllabification examples*

Sequence	Syllabification	ph-class	CVCV
pane	pa.ne	OV.NV	CV.CV
aglio	aL.Lo	VS.SV	VC.CV
aja	a.ja	V.GV	V.CV
per	per	OVR	CVC
bacio	ba.tSo	OV.OV	CV.CV
pasta	pas.ta	OVO.OV	CVC.CV
strano	s.tra.no	O.OSV.NV	C.CCV.CV
zio	tsi.o	OV.V	CV.V
pazzo	pat.tso	OVO.OV	CVC.CV
gatto	gat.to	OVO.OV	CVC.CV
paura	pa.u.ra	OV.V.SA	CV.V.SV
aiuola	a.jwO.la	V.GGV.SV	V.CCV.CV
nafta	naf.ta	NVO.OV	CVC.CV

predicted by assigning a zero-sonority to the sibilant: *pasta* → '*pa.sta*'; *stella* → '*stel.la*'. A change in the SH is sufficient to handle the distribution of sC clusters.

The hiatus or diphthong dilemma is not a direct problem of the syllabification algorithm itself, because at any point in the processing (both for derived and underlying glides) the segment must be syllabified according to its current status. Given a cluster of two vowels (two nuclei) the syllabification will end up in a hiatus (heterosyllabic cluster). Otherwise, any sequence of glide plus vowel, or vowel plus glide will result in a diphthong (tautosyllabic cluster)[5].

3. Test and Results

Some recent papers [14][25] tried to demonstrate that data-driven models are more accurate than rule-based models by comparing the syllabification programs at hand over syllabifications as present in dictionaries; but we think that this methodology may miss one fundamental aspect concerning syllabification. Two objects of a very different nature are being compared: a natural language process and the output of the application of a normative set of rules. Data-driven methods undoubtedly perform better in such a task, because they guess the syllabification patterns used in dictionaries (which in turn partially refers to phonological as well as orthographic, etymological and morphological normative analyses) while rule-based models do not try at all to simulate this kind of dictionary syllabification. There are of course cases when orthographic syllabification is required, but these two types of syllabifications are to be kept distinguished and are in no case comparable [25].

For these reasons we propose a manual verification test. Three linguists were asked to manually syllabify a portion of a corpus and the resulting syllables were then compared with automatic syllabification obtained from the same corpus. The test aims to verify the accuracy of the syllabifier and not to investigate the linguistic competence of the speakers, so some explicit guidelines were given (§3.2) for sC cluster syllabification and for corpus encoding conventions. We will first discuss the data used (§3.1), the test design (§3.2), and finally present the obtained results (§3.3).

3.1. Materials

All testing material was taken from CLIPS [26], a corpus of spoken Italian. CLIPS contains time-aligned phonetic and phonological transcription of approximately one hundred hours of speech recording, and is divided into five sub-corpora[6]. In this work, we will consider the phonological transcripts (STD layer [27]) of dialogue and ortho-phonic sections of the corpus. These could be taken as representatives of the two ends of a naturalness spectrum: the dialogue corpus is made of semi-spontaneous speech, while the ortho-phonic one is the most artificial, as it consists of sentence lists read by professional speakers, and made up to cover the widest range of Italian phonotactic clusters.

The STD layer consists of time-aligned citation forms, enriched with lexical, non-lexical and segmentation symbols [27]. The symbols in the corpus had to be handled with the two procedures implemented in the transducers (§2.1). In particular, pauses in the speech and major syntactic boundaries triggered *Send*, thus forcing a syllable boundary while other symbols were not translated at all (translation to zero) as they did not affect syllabification. The latter set includes word boundary markers (blank spaces), because in Italian they are almost always context for resyllabification when not strengthened by other prosodic or syntactic boundaries.

3.2. Methods

To carry out the manual syllabification, three undergraduate linguistics students of the University of Pisa were asked to syllabify five random sentences at least ten words long from the dialogue sub-corpus and, for phonotactic covering, five sentences in the orthphonic corpus. The students were first asked to syllabify two example sentences in a training session in order to become familiar with the corpus coding. Arbitrary or ambiguous cases were resolved in the following instructions: re-syllabification always applies across word boundaries, but not across pauses or other major syntactic boundaries; sC clusters are heterosyllabic, and always syllabify according to the given transcription.

3.3. Test Results

In a total of 1008 syllables, the automatic syllabification was faithful to the manual one (see table 3). In the dialogue corpus syllabification, subject A obtains a ration of 1.0, while the other two make minor mistakes, including the missing re-syllabification of a sC cluster and two hiatus syllabified as tauto-syllabic, but always obtaining a ratio > 0.98. The ortho-phonic test mirrors the results of the dialogue. The data shows a high ratio of accuracy and the few differences result again from the missing application of re-syllabification across word boundaries and in the non-native [km] cluster in the word acme.

These results show that the program syllabification is very close to those made by human experts. Almost every minor

[5]No assumption is made on the internal constituency of the diphthongs [24].

[6]Dialogues, Radio and Television, Telephonic, Read speech, Ortho-phonic.

Table 3: *Syllabification test results.*

Subject	Dialogue	Ortho-phonic
A	1	0.98550
B	0.99750	1
C	0.98876	0.98551

difference can be analysed as an inconsistent performance error in manual syllabification. Syllabification in Italian can be mostly predicted algorithmically, even when accounting for minor word boundary segmentation phenomena found in speech.

4. Conclusions

In this paper we present a tool that can be adapted for different tasks, purposes and encodings. The modular structure of the program is inspired by linguistic principles and constraints. The syllabification algorithm was reduced to the simplest and most general mechanism possible, thus demonstrating that it is possible to handle specific cases of Italian syllabification entirely through changes to the sonority scale. – in real-life applications, this is a crucial function, because the algorithm does not depend on data, but on structures. As we did for the testing, the porting to other data sets is straightforward and does not require any tuning of the algorithm itself. Finally, it is possible to use an instrument that implements a phonological theory as a testing platform for the theory itself, as has been done to prove the presented SH. A manual verification of the output of the program confirmed the accuracy of the syllabification in a variety of contexts. Particular clusters arising from re-syllabification as well as segmentation symbols were handled correctly by the automatic syllabification.

Possible uses of the program include corpus syllabification, syllable generation and quantitative syllable analysis. In incoming work we will compare an acoustic syllabification algorithm based on the detection of peaks of intensity in the signal with the time-aligned phonological syllabification obtained with Sylli. The difference in the syllabification will eventually help us understand where the signal-based analysis more often fails, the contexts where the algorithm is less accurate and if these contexts are predictable. In another work the syllabic portions of the acoustic signal obtained by the syllabification of a time-aligned transcription will be extracted automatically and used in place of segmental units as feeding units for TTS and RS systems.

5. References

[1] J. Goldsmith, "An overview of autosegmental phonology," *Phonology: critical concepts in Linguistics*, vol. 3, pp. 382–425, 2001.

[2] A. Prince and P. Smolensky, *Optimality Theory: Constraint interaction in generative grammar*. MA: Wiley-Blackwell, 2004.

[3] W. Daelemans and A. Van Den Bosch, "Generalization performance of backpropagation learning on a syllabification task," in *Proc. of the 3rd Twente Workshop on Language Technology*. Enschede: Twente University, 1992, pp. 27–37.

[4] I. Stoianov, J. Nerbonne, and H. Bouma, "Modelling the phonotactic structure of natural language words with Simple Recurrent Networks," in *Computational Linguistics in Netherlands*, 1997, pp. 77–95.

[5] M. Ostendorf, "Moving beyond the a 'beads-on-a-string' model of speech," in *Proc. IEEE ASRU Workshop*, 1999, pp. 79–84.

[6] T. Vennemann, *Preference laws for syllable structure and the explanation of sound change: with special reference to German, Germanic, Italian, and Latin*. Berlin, New York, Amsterdam: Mouton de Gruyter, 1988.

[7] G. Clements, "The role of the sonority cycle in core syllabification," *Papers in laboratory phonology*, vol. 1, pp. 283–333, 1990.

[8] S. Davis, "Italian Onset Structure and the Distribution of il and lo," *Linguistics*, vol. 28, no. 1, pp. 43–55, 1990.

[9] D. Kahn, "Syllable-based generalizations in English phonology," Ph.D. dissertation, MIT, 1976.

[10] J. Kaye, "Do you believe in magic? The story of s+C sequences," *SOAS Working Papers in Linguistics*, vol. 2, pp. 293–313, 1992.

[11] P. M. Bertinetto, "On the undecidable syllabification of /sC/ clusters in Italian: Converging experimental evidence," *Italian Journal of Linguistics*, vol. 16, no. 2, pp. 349–372, 2004.

[12] N. Vincent, "The Romance Languages," in *Italian*, M. . N. Vincent, Ed. London: Croom-Helm, 1988, pp. 279–313.

[13] M. Krämer, *The phonology of Italian*. Oxford University Press, USA, 2009.

[14] Y. Marchand, C. Adsett, and R. Damper, "Automatic syllabification in English: A comparison of different algorithms," *Language and Speech*, vol. 52, no. 1, pp. 1–27, 2009.

[15] M. Hammond, "Parsing in OT," *Ms., University of Arizona (ROA-222)*, 1997.

[16] R. Weerasinghe, A. Wasala, and K. Gamage, "A rule based syllabification algorithm for Sinhala," *Natural Language Processing IJCNLP 2005*, pp. 438–449, 2005.

[17] J. Beck, D. Braga, J. Nogueira, M. S. Dias, and L. Coelho, "Automatic Syllabification for Danish Text-to-Speech Systems," in *Proc. of INTERSPEECH-2009*, 2009, pp. 1287–1290.

[18] P. Oudeyer, "The origins of syllable systems: an operational model," in *Proc. of the 23rd Annual Conf. of the Cognitive Science Society, COGSCI2001*. London: Laurence Erlbaum Associates, 2001, pp. 744–749.

[19] B. Laks, "A connectionist account of French syllabification," *Lingua*, vol. 95, no. 1-3, pp. 51–76, 1995.

[20] F. Cutugno, G. Passaro, and M. Petrillo, "Sillabificazione fonologica e sillabificazione fonetica," in *Dati Empirici e Teorie Linguistiche, Atti del XXXIII, Congresso della SLI*. Roma: Bulzoni, 2001, pp. 205–232.

[21] L. Cioni, "An algorithm for the syllabification of written Italian," in *V Simposio Internacional de Comunicacion Social*, Santiago de Cuba, 1997, pp. 22–24.

[22] B. Calderone and P. M. Bertinetto, "La sillaba come stabilizzatore di forze fonotattiche. Una modellizzazione," in *Linguistica e modelli tecnologici di ricerca, Atti del XLI Congresso della SLI*. Roma: Bulzoni, 2009, pp. 401–410.

[23] J. C. Wells, "Sampa computer readable phonetic alphabet," in *Handbook of Standards and Resources for Spoken Language Systems*, D. Gibbon, R. Moore, and R. Winski, Eds. Berlin and New York: Mouton de Gruyter, 1997, part IV, section B, [Online], Available: www.phon.ucl.ac.uk/home/sampa/italian.htm.

[24] G. Marotta, "The Italian diphthongs and the autosegmental framework," in *Certamen Phonologicum. Papers from the 1987 Cortona Phonology Meeting*. Torino: Rosenberg & Sellier, 1988, pp. 389–418.

[25] C. Adsett, Y. Marchand *et al.*, "Syllabification rules versus data-driven methods in a language with low syllabic complexity: The case of Italian," *Computer Speech & Language*, vol. 23, no. 4, pp. 444–463, 2009.

[26] R. Savy and F. Cutugno, "CLIPS. Diatopic, diamesic and diaphasic variations in spoken Italian," in *Proc. of the 5th Corpus Linguistics Conference*, Liverpool, 2009, [Online], Available: http://ucrel.lancs.ac.uk/publications/cl2009/213_FullPaper.doc.

[27] R. Savy, "Progetto clips: Specifiche per l'etichettatura dei livelli segmentali," in *Analisi di un dialogo*, F. Albano Leoni and R. Giordano, Eds. Napoli: Liguori, 2007, pp. 1–37.