A syllable segmentation algorithm for English and Italian

Massimo Petrillo[§*], *Francesco Cutugno*[^*]

[§]Dipartimento di Informatica e Sistemistica
[^]Dipartimento di Scienze Fisiche
[*]CIRASS
Università degli Studi di Napoli "Federico II"
{massimo.petrillo,cutugno}@unina.it

Abstract

In this paper we present a simple algorithm for speech syllabification. It is based on the detection of the most relevant energy maximums, using two different energy calculations: the former from the original signal, the latter from a low-pass filtered version. The system requires setting appropriate values for a number of parameter. The procedure to assign a proper value to each one is reduced to the minimization of a n-variable function, for which we use either a genetic algorithm and simulated annealing. Different estimation of parameters for both Italian and English was carried out. We found the English setting was also suitable for Italian but not the reverse.

1. Introduction

The problem of automatic segmentation of a speech utterance into syllabic portions was first attempted in 1970 by Maerlmelstein [1]. He used a loudness function obtained by giving a weight to each element within a set of spectral bands. An algorithm evaluating the shape of the loudness pattern (convex-hull) was used to find syllable boundaries.

Pfitzinger et al. [2] processed the speech signal using a bandpass filter, then they computed the energy pattern using a short term window and finally they low-pass filtered this energy function. The syllable nuclei were found from local maxima of the energy contour. Another important result of Pfitzinger and colleagues was the comparison of manual syllabic segmentation by several human labelers. They found an agreement of only 96% on nuclei position, so they assumed this value as the upper limit for any automatic segmentation.

Another algorithm for speech syllabification was developed in 1998 by Jittiwarangkul et al. [3]. The core of their method was energy computation and smoothing. They tested various kinds of temporal energy functions for syllable boundary detection. The behavior of their algorithm depends on a number of empirically predefined thresholds. Differently from the work we are going to present here, these authors did not supply any information about the methods used to choose the parameter values in order to obtain the best performance of the segmenter.

Reichl and Ruske [4] were among the first to use neural networks to segment speech into syllables. Their goal was to find syllabic nuclei in German read sentences. The features extracted from the speech signal were Bark-scaled loudness spectra calculated every 10 ms. Two kinds of artificial neural network were compared, a multilayer perceptron and a radial basis function neural network.

Wu et al. [5] computed smoothed speech spectra using two-dimensional filtering techniques, enhancing in this way energy changes of the order of 150ms, they applied further techniques aiming at emphasizing syllable onsets related to positive changes in the energy pattern. An averaging over nine critical frequency bands was also computed every 10ms. The resulting vector was concatenated with log-RASTA features and used as input to a multilayer perceptron.

Greenberg [6] introduced the speech modulation spectrogram, a system for the research of invariant features related to frequency portions of the speech spectrum distributed across critical band-like channels. Invariants, according to Greenberg, mainly lie in slowly varying dynamical features present in the speech signal. Temporal constants involved in the processing and recognition of speech features take into account at the same time two different kind of factors: the former connected with speech rhythm parameters and the latter related with auditory temporal integration of the slowest spectral components.

Starting from the modulation spectrogram Shastri et al. [7] used a different kind of artificial neural network, the temporal flow network introduced by Watrous [8]. With this tool they computed a function having local peaks at syllabic nuclei. The main difference between this net and the multilayer perceptron is that it allows recurrent links and time delay.

The present paper introduces an algorithm for speech segmentation into syllabic units. Its behavior depend on a set of parameters (mainly thresholds and window lengths) which must be tuned in order to achieve the best performance. Due to the large number of parameters, we used a number of automatic methods for parameter tuning.

2. Syllable segmentation of the speech signal

The algorithm is based on the analysis of the temporal pattern of the energy of the speech signal. The energy of the signal was computed as follows for non overlapping frames:

$$E_k = \log \sqrt{\sum_{i=k*w}^{(k+1)w-1} s_i^2}$$

where E_k is the value of the energy for the k-th frame, s_i is the i-th signal sample and the w is the frame length in samples (fixed at 11.6ms). The logarithm is used in order to introduce a rough similarity with the perceptual concept of loudness. As we are mainly interested in capturing any important oscillation we choose a length that is short enough to contain any significant contour variation, however some oscillations not related to syllabic structure can still occur. So our algorithm must distinguish which oscillations reflect the syllabic sequence and which do not. Our algorithm is based on two different energy calculations. The first one is the energy of the signal itself, namely "total energy", the second is the energy of the signal filtered with a low pass filter with a cut-off frequency of 1100Hz, namely "residual energy". As can be clearly seen in Fig. 1, the two contours are almost always the same, except in the fricative portions of the utterances. It is easy to see how each prominence in energy contour is on average associated with a syllable, but prominences are also present in some fricative portions of the signal. The basic idea of our algorithm is to use prominences of total energy contour to detect syllable nuclei and residual energy contour to discard some syllable boundaries.

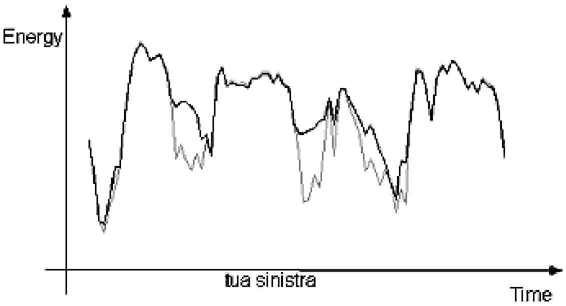

Figure 1: Total and residual energy of the Italian utterance "tua sinistra" - [ˈtuasiˈnistra] - *your left (side)* -

In the following section we will describe the entire method. The algorithm starts with a procedure (described in section 2.1) that produces as output a rough segmentation needing further refinement. As already mentioned, a large number of parameters are needed in order to achieve good segmentation performance. We will give a detailed description of each of them. For clarity we will use italics to indicate parameter names.

2.1. Starting procedure

The first segmentation is made from only those local maxima that are greatest in a window of radius *step*. Fig. 2 shows the selection procedure. The syllable boundaries are, at this stage, at the lowest local minimum of energy between two nuclei.

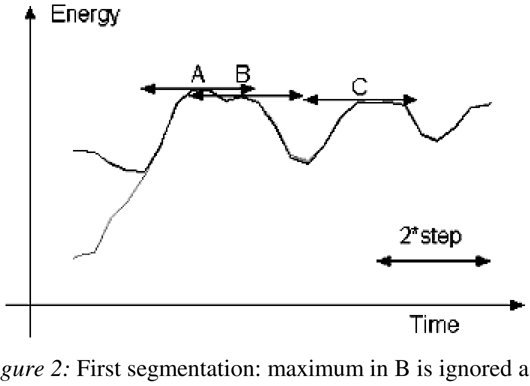

Figure 2: First segmentation: maximum in B is ignored as it is not absolute in the window

At this stage, these errors may occur:

- missed recognition of short syllables
- fricatives segmented as syllables
- long stressed vowels split into two syllables

The procedures described in the following sections try to correct some of these errors.

2.2. Missed recognition of short syllables: splitting

The first recovering step tries to individuate syllable boundaries that were erroneously missed in the previous analysis. The sequence of syllables produced by the first stage is scanned and, for each syllable, all the minima are taken into account in order to find further significant intensity variations.

This step requires four parameters. The maximum in a fixed neighborhood of each minimum is taken into account. The radius *splitStep* of the neighborhood is one of the parameters of the syllabification procedure. The ratio between the maximum and the minimum must be above a threshold in order to split the original segment into two syllables. The threshold is chosen on the basis of the length of the shortest obtained syllable. If it is longer than *shortLengthSplit* the ratio must be above *longRatioSplit* otherwise it must be greater than *shortRatioSplit*. The latter parameter should be greater than *longRatioSplit* in order to require a greater energy variation to split short segments.

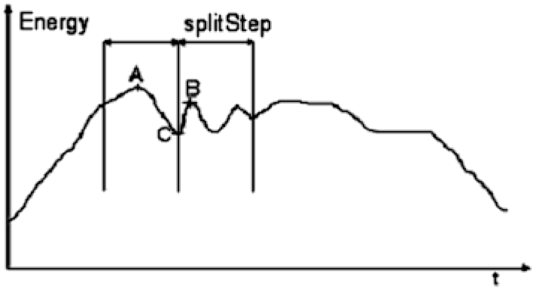

Figure 3: Splitting: if the ratio between B and C is higher than a fixed threshold then the syllable is split.

2.3. Fricative assimilation

The initial segmentation may produce syllables containing only fricative segments; the following step attempts to assign such syllables to one of the adjacent syllables.

A segment is classified as purely fricative if two conditions are simultaneously satisfied:

- the ratio between total and residual energy is greater than *maxFricativeRatio* at the location of the residual energy maximum;
- the average ratio between total and residual energy is greater than *averageFricativeRatio.*

In order to assign these fricatives to the previous or following syllable, the residual energy is used. Fricative sounds have a significant difference between total energy and residual energy. If the speech sound following the fricative is a stop consonant then the slope of the residual energy within the fricative portion will decrease. In all other cases the residual energy will increase or will remain steady. Based on this assumption, we consider the shape of the residual energy in each segment previously labeled as fricative: if it shows a decreasing trend, then the fricative segment is assigned to the left syllable, otherwise it is assigned to the right syllable.

The trend is estimated as decreasing if simultaneously:

- the maximum of the residual energy occurs in the first frame of the fricative and the ratio between the maximum and the last value is greater than *decreasingResidualRatio;*
- the ratio between the maximum and the last value is greater than *decreasingResidualRatio*

2.4. Stressed vowel recompaction

This fourth step is required to detect insertion errors that can occur in long stressed vowels. Some slight energy falls may occur if a vowel is very long, 300ms or more, causing syllable insertion errors. It is possible to recover the error by deleting the syllable boundary if one of the following conditions, regarding the inverse of the ratio between the valley and its lowest adjacent peak, are true:

- the ratio is less than *recompactionRatio;*
- the ratio is less than *mediumRecompactionRatio* and the duration of the obtained segment is less than *mediumRecompactionLength;*
- the ratio is less than *longRecompactionRatio* and the duration of the obtained segment is less than *longRecompactionLength.*

These three conditions apply respectively to: medium, short and long syllables; in order to obtain more “recompaction” of short syllables rather than long syllables.

3. Parameters Tuning

The algorithm presented above requires the determination of the proper values for a set of parameters to work accurately. These parameters prevents their direct computation from the speech signal, so the best configuration of parameters has to be found by testing the analysis procedure using a finite set of configurations. The high number of parameters can easily discourage every effort to set their value in a precise fashion.

Our goal was then to develop a set of tools for optimal parameter tuning. These tools will generate sets of parameters with which to run the procedure over a speech corpus. This corpus is a training set: it has been previously manually annotated by experts who provided a set of temporal markers. The better parameter set will be the one that will minimize a distance function between manual annotation and the output of our algorithm. The process of finding the parameters is then equivalent to a problem of minimization of a function of *n* variables where *n* is the number of parameters used in the algorithm.

Among the various techniques available to solve the problem of function minimization in this field we chose to focus our effort to the following:

- random search
- genetic programming
- simulated annealing

Random search is the simplest method. For each parameter a range of acceptable values must be provided manually, so we only need to generate and test a number of random parameter configurations. Genetic programming [9] starts from a number *n* of random parameter configurations. New configurations can be created in two ways:

- exchanging parameter values among configurations;
- slightly changing some value.

After each iteration new configurations will be created and the worse ones will be discarded.

It can be seen as a metaphor of the Evolution Theory, where our parameters correspond to genes, the exchange of value to DNA crossover and the slight change of a parameter to a genetic mutation.

The last method used, simulated annealing [10], is a metaphor of the annealing of metals. In natural annealing a melted metal is slowly cooled to permit to atoms to get the position of minimum global energy, during the process each atom changes its position due to thermal agitation, if its new position decreases the global energy, the atom holds the new position, otherwise it may or may not hold the new position according to a probability function related to the temperature of the metal and with the energy gain:

$$P_i = e^{-\frac{E_i - E_{i-1}}{T_i}}$$

where P_i is the probability of an energy gain at step *i*, E_i and T_i are the current energy and temperature at step *i*.

The higher the temperature and the lower the energy gain, the higher the probability of holding the new position even if the energy increases. At very high temperatures each atom is free to reach every position (in the melted state), as temperature decreases more and more positions become difficult to reach because jumps to higher energy become less probable.

In our problem the position of an atom corresponds to the value of a parameter, while the energy corresponds to the distance between the corpus annotation and the output of the procedure. The possibility of reaching higher energy configurations permits Simulated Annealing to avoid getting stuck in local minima.

4. Results

The algorithm was trained and tested separately for Italian and English. Training and test data for Italian was a subset of the CLIPS [11] corpus, while for English we used a subset of the Boston Radio News Corpus.

Speech from CLIPS was collect via the map tasks [12], it is manually labeled at the phonetic level. The syllable labels were automatically derived from transcriptions using the algorithm explained in [13]. The subset used is composed of male and female speech uttered by speakers from different regions of Italy. A total of 2923 syllables were used for the process of parameter tuning and 2066 for testing.

The English corpus is composed of read speech from a professional female speaker (ID code: F2B) , 2928 syllables were used for training and 1742 for test.

Each parameter tuning method was repeated five times on the Italian training corpus. In every trial 500 parameter configurations were tested. That was done in order to know which technique is more suitable to our aims. The simulated annealing performed slightly better (15% error rate including deletion and insertions, while genetic 16%, random 17%).

Simulated annealing was then repeated using a larger (2000) number of configurations, for five times again in each training session.

Table 1 shows the results obtained for Italian and Table 2 for English.

	Insertion		Deletion		Total	
Training	188	6.4%	192	6.6%	380	13.0%
Test	167	8.1%	134	6.5%	301	14.6%

Table 1: Results of the original algorithm on Italian

	Insertion		Deletion		Total	
Training	234	8.0%	382	13.0%	616	21.0%
Test	163	9.4%	216	12.4%	379	21.8%

*Table 2:*Results of the original algorithm on English

The poor results obtained for English motivated a slight modification of the algorithm. The only change was to use residual energy in place of total energy in the steps described in sections 2.1, 2.2 and 2.4. We then obtained the results in Table 3 and Table 4.

	Insertion		Deletion		Total	
Training	129	4.4%	339	11.6%	468	16.0%
Test	63	3.6%	200	11.5%	263	15.1%

Table 3: Results of the new algorithm on English

	Insertion		Deletion		Total	
Training	232	7.9%	178	6.1%	418	14.0%
Test	170	8.2%	115	5.6%	285	13.8%

Table 4: Results of new algorithm on Italian

The performance of the new algorithm is almost the same as the previous one for Italian, but better for English.

As a final experiment, we cross-evaluated the algorithms trained for one language using speech of the latter language. We obtained poor results from both the "Italian" algorithms when they processed English speech but, the if training was on the English corpus, the results on Italian speech were not as degraded as expected. Table 5 shows these results.

	Insertion		Deletion		Total	
Original Italian on English	111	6.4 %	488	28.0 %	599	34.4 %
New Italian on English	124	7.1 %	570	32.7	694	39.8 %
Original English on Italian	136	6.6 %	213	10.3 %	349	16.9 %
New English on Italian	111	5.4 %	200	9.7%	311	15.1 %

Table 5: Cross-language results on test sets (1742 English syllables, 2066 Italian syllables)

The asymmetric behavior of "cross language" performance could be due to the more complex syllable inventory of English. For this reason, training on Italian speech can not take into account of the variety of English syllables, while when training on English, most of the Italian speech features will be present, so the results on Italian remain not very different from those obtained with training on Italian speech.

5. Acknowledgements

Part of this work was carried out at Centre for Speech Technology Research – University of Edinburgh. Special thanks to Simon King for his help.

6. References

[1] Maermelstein, P. "Automatic segmentation of speech into syllabic units", *J. Acoust. Soc. Am.*, pp. 880-883, 58 (4), 1975

[2] Pfitzinger, H.R.,Burger, S.,Heid, S. "Syllable detection in read and spontaneous speech". *Proceedings of the Fourth International Conference on Spoken Language (ICSLP),* 1996.

[3] Jittiwarangkul, N., Jitapunkul, S., Luksaneeyanavin, S.,; Ahkuputra, V., Wutiwiwatchai, C. "Thai syllable segmentation for connected speech based on energy" *Circuits and Systems*, 1998. IEEE APCCAS 1998. The 1998 IEEE Asia-Pacific Conference on pp 169 –172, 1998.

[4] Reichl, W. and Ruske, G., "Syllable segmentation of continuous speech with artificial neural networks" in *Proceedings of Eurospeech93*, 3rd European Conference on Speech Communication and Technology, Berlin, pp. 1771-1774, 1993.

[5] Wu, S.L., Shire, M., Greenberg, S., Morgan N. "Integrating syllable boundary information into speech recognition" *IEEE International Conference on Acoustics, Speech and Signal Processing*, Munich, pp. 987-990, 1997.

[6] Greenberg, S. and Kingsbury, B. "The modulation spectrogram: In pursuit of an invariant representation of speech," *ICASSP-97, IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1647-1650, 1997.

[7] Shastri, L., Chang, S., Greenberg S. "Syllable detection and segmentation using temporal flow neural networks" *Proceedings of the Fourteenth International Congress of Phonetic Sciences*, San Francisco, 1999.

[8] Watrous, R. L. "GRADSIM: a connectionist network simulator using gradient optimization techniques," Report, Siemens Corporate Research, Inc., Princeton, New Jersey, 1993

[9] Carnahan, J. and Sinha, R. "Nature's algorithms [genetic algorithms]" *IEEE Potentials*, Volume: 20 Issue: 2, April-May 2001 Page(s): 21 –24.

[10] Kirpatrick S., C.D. Gelatt, M.P. Vecchi "Optimization by simulated annealing". *Science* 220, 671-680. 1983

[11] Albano Leoni, F., Paoloni, A., Refice, M., Sobrero, R.A. "CLIP Corpus della Lingua Italiana Parlata (Corpus of Spoken Italian)" Proceedings of 1st LREC conference, Grenada 1998.

[12] Anderson, A.H., Bader, M., Bard, E.G., Boyle, E., Doherty-Sneddon, G.M., Garrod, S., Isard, S., Kowtko, J.C., McAllister, J.M., Miller, J.E., Sotillo, C.F., Thompson, H.S., Weinert, R.. "The HCRC Map Task Corpus" *Language and Spee*ch, 34(4):351–366, 1991.

[13] Cutugno, F.; Passaro, G.; Petrillo, M "Sillabificazione fonologica e sillabificazione fonetica" in *Atti del XXIII Congresso della SLI (Società di Linguistica Italiana)*, Napoli, 28-30 October 1999, pp. 205-232.