Proceedings of the Federated Conference on Computer Science and Information Systems pp. 265–269
DOI: 10.15439/2017F326
ISSN 2300-5963 ACSIS, Vol. 11

PitchKeywordExtractor: Prosody-based Automatic Keyword Extraction for Speech Content

Iurii Lezhenin*, Artyom Zhuikov*,
Natalia Bogach*, Elena Boitsova†, Evgeny Pyshkin‡
*Institute of Computer Science and Technology Peter the Great St. Petersburg Polytechnic University
194021 St. Petersburg Polytechnicheskaya, 21 Email: bogach@kspt.icc.spbstu.ru
†Institute of Humanities Peter the Great St. Petersburg Polytechnic University
194021 St. Petersburg Polytechnicheskaya, 19 Email: el-boitsova@yandex.ru
‡Software Engineering Lab. University of Aizu
Aizu-Wakamatsu, 965-8580, Japan Email: pyshe@u-aizu.ac.jp

***Abstract*—Keyword extraction is widely used for information indexing, compressing, summarizing, etc. Existing keyword extraction techniques apply various text-based algorithms and metrics to locate the keywords. At the same time, some types of audio and audiovisual content, e. g. lectures, talks, interviews and other speech-oriented information, allow to perform keyword search by prosodic accents made by a speaker. This paper presents *PitchKeywordExtractor* - an algorithm with its software prototype for prosody-based automatic keyword extraction in speech content. It operates together with a third-party automatic speech recognition system, handles speech prosody by a pitch detection algorithm and locates the keywords using pitch contour cross-correlation with four tone units taken from D. Brazil discourse intonation model.**

I. Introduction

KEYWORDS make the semantic backbone of a text. As keywords reflect the text ideas and convey text meaning they are used for text indexing, analysis, summarizing compression, etc. [1]. In modern world of on-line information abundance automatic keyword extraction techniques are extremely in-demand ([2], [3]).

There is a great number of research in the area of automatic keyword extraction either for individual documents e. g. [4], [5], or large document corpora [6], as well as for specific types of on-line content like e-newspapers [7] or micro-blogs on Twitter [8]. Content-based retrieval research [9] is also highly relied upon the keywords [10].

Some of these techniques use document corpora, while others do not. When a document corpus is used, a function which balances a measure of a keyword within a document (frequency, location or co-occurrence) with a similar measure from the corpus is applied. When corpus is unavailable, keyword extraction techniques use lexical or semantic analysis or keywords co-occurrences over an individual document. An excellent literature review on automatic keyword extraction techniques is presented in [11]. Automatic keyword extraction techniques for text compression and summarizing can be found in [3].

In comparison with text processing techniques specific audio and audio-visual speech content keyword extraction algorithms are less developed. Meladianos et al. [12] report on a high demand for speech processing from the point of view of information mining. The actual research in this area is usually based on a preliminary audio-to-text conversion by means of automatic speech recognition system (ASR) and further application of content-sensitive text-based techniques (e. g. see Elakiya K. et al. [13] or G. Alharbi [14]).

At the same time, speech content has an inherent powerful feature, namely, speech prosody (i. e. intonation, rhythm, tempo, pausing, etc.) that can help to locate and extract keywords. We use the term "prosody" exactly in the sense of D. Brazil system of discourse intonation (DI) [15], [16], [17] and refer to his tone units to define the prosodic patterns for *PitchKeywordExtractor*. The working hypothesis of the present research is based on concept that keywords being the most informative parts of speech are prosodically highlighted by a speaker, and, therefore, they must have specific discernible prosodic characteristics.

Speech prosody is observable by measuring the fundamental frequency (pitch) and there exist a variety of speech processing tools e. g. see *Praat* or *Visi-Pitch* or *TarsosDSP* [18] to analyse prosodic characteristics as per pitch detection and estimation algorithms [19], [20]. A perfect guideline for special software operation can be found in [21].

There have been much research, discussion and critics on prosody-based methods applicability and limits. Now they go far beyond simple pitch measurement and exist as components for complex analytic frameworks: e. g. see P. Roach [22] or A. Meftah et al. in [23] for prosody-based systems of emotion recognition. A deep insight into the contribution of prosody-based techniques to corpus linguistics was made by M. Warren [24].

On the assumption that automatic keyword extraction can benefit from prosody-based analysis we propose to add processing of prosodic features to automatic keyword extraction algorithms as far as speech content is concerned. We present *PitchKeywordExtractor* - a prosody-based tool for automatic keyword extraction. Operating together with a third-party ASR and speech processing software *PitchKeywordExtractor* searches for keywords in speech content by matching their prosodic characteristics to ASR output text.

II. Methodology and materials

It is widely recognized that keywords in speech have not only statistically measurable features or occupy a certain sentence position, but are usually highlighted by intonation because they frequently act as speech signals for given and new information [25]. The way to make this tonal emphasis may be different depending upon the context and background of the speakers. For our analysis we have taken 4 tones from the tonal model of discourse intonation developed by D. Brazil [16] which is widely used in linguistics to describe the semantic aspect of speech prosody. This model comprises 5 principle tones of English speech: fall, rise, fall-rise, rise-fall and level. D. Brazil also defines the speech situations when each of these tones occurs.

Fall tone (p-tone) and rise-fall tone (p+-tone) are defined by Brazil as proclaiming tones, so they are used to mark new information introduced by a speaker, therefore, these tones may indicate the keywords entries. Among those the rise-fall tone is defined as "dominant proclaiming" and it highlights not only new, but important information, so it can be a strong keyword entry marker.

At the same time, fall-rise and rise tones are "referring", r-tone and r+-tone respectively. In speech they mark the already known information, i. e. the common ground of the speakers. These tones may also indicate keyword entries.

We consider four model tone units (fall, rise, fall-rise, rise-fall) to be searched for in speech. Strictly speaking, Brazil tones describe phrasal intonation and refer not to one word but to a whole semantic unit, i. e. a syntagm. A tone pattern has complex structure, namely, a pre-head, head, nucleus and tail and refers to a part of a phrase (or to a whole phase, if it is short); while a keyword can be marked by the nucleus only. However, the entire tone pattern can be located more accurately with correlation, while nucleus is too short to provide a good correlation peak. Thus, we are looking for keywords inside a phrasal tone pattern provided corresponding phrase pitch contour is obtained, compare to model tone units and map it to ASR output to retrieve the keywords.

The architecture of *PitchKeywordExtractor* consists of 4 main parts (see Fig.1):

1) Pitch Detector
2) Tone Unit Detector
3) Speech Recognizer
4) Segment-to-word Mapper

A. Pitch Detector

Pitch Detector obtains pitch series $s[k]$ for a given speech record. We use a third-party YIN [26] pitch detection algorithm provided with *TarsosDSP* [18].

B. Tone Unit Detector

Pitch series $s[k]$ are subsequently processed by Tone Unit Detection Algorithm (see Sec. III for details). The output of Tone Unit Detector is a set of segments (time intervals) where model tone units were found.

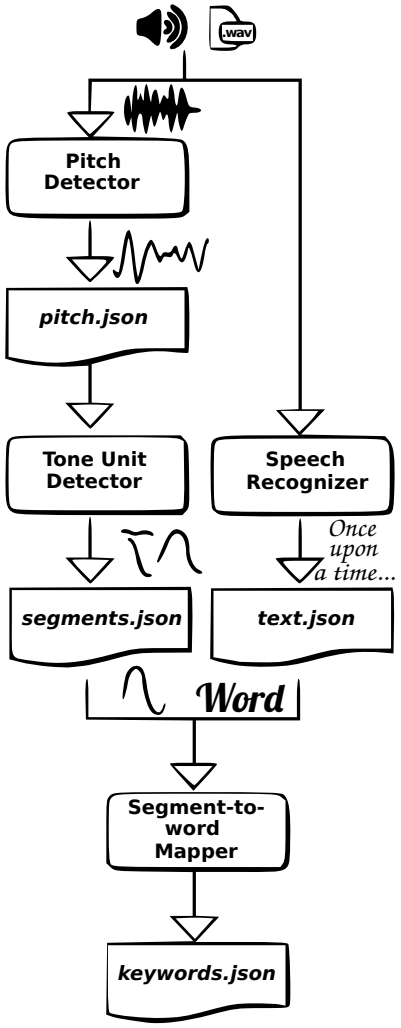


Fig. 1: *PitchKeywordExtractor* Flowchart

C. Speech Recognizer

Speech Recognizer produces text for a given speech record to create the reference wordlist. Sphinx [27] is used in *PitchKeywordExtractor* prototype by now, while this block may be implemented with any alternative solution for speech recognition.

D. Segment-to-word Mapper

The segments received from Tone Unit Detector and Speech Recognizer output file are mapped to each other to locate a word within a segment (see Sec.IV for details). Segment-to-word Mapper output is the final keyword list.

III. Tone Unit Detection Algorithm

Tone unit detection is based on the correspondence of a syntagm pitch contour and one or more model tone units.

A. Preliminary Assumptions

Tone unit detection is performed on evenly distributed pitch series $s[k]$ obtained as per pitch detection algorithm.

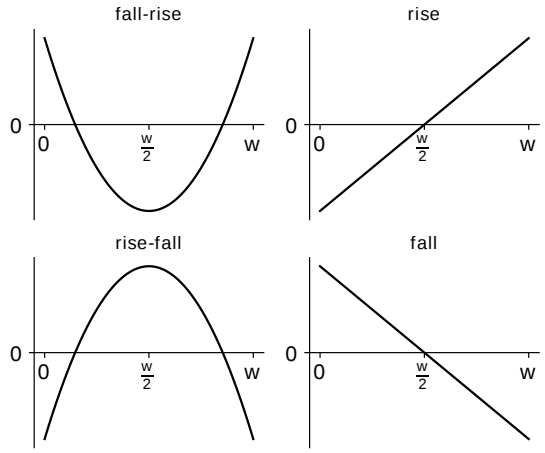


Fig. 2: Model tone units

Let us define 4 discrete-time limited basic functions: $\phi_w^f(x), \phi_w^{rf}(x), \phi_w^{fr}(x), \phi_w^r(x)$ of w length, $w \in [w_{min}, w_{max}]$, where w_{min}, w_{max} are the empirically chosen syntagm boundaries; $x \in \mathbb{Z}$, $0 \le x \le w$. These functions correspond to Brazil tone model (see Fig. 2) as follows:

$\phi_w^f(x)$ – "fall tone" p-tone
$\phi_w^{rf}(x)$ – "rise-fall" p+-tone
$\phi_w^{fr}(x)$ – "fall-rise" r-tone
$\phi_w^r(x)$ – "rise" r+-tone

B. Pre-Processing

1) Median filtering [28] is applied to remove single prominences in $s[k]$.
2) $s[k]$ is divided into the datasets $\{s_j[k]\}$, bounded by natural pauses in speech (silence).
3) Too short datasets $\{s_j[k]\}$ are not processed as statistically inconsistent.

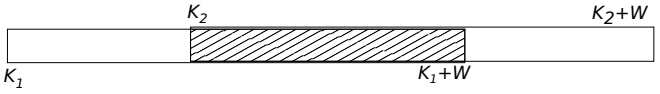
C. Processing

The following Algorithm 1 is subsequently applied to all datasets $\{s_j[k]\}$ and all model tone units $\phi_w(x)$. Values of correlation coefficient $r_\phi(k,w) \in [-1,1]$ are used to estimate the similarity between the model tone unit ϕ_w and the pitch contour of a segment, which starts at k and ends at $k+w$. $r_\phi(k,w)$ is calculated only for full-size segments, i. e. k varies in the range of $[0, K_j - w]$ that discards the edge issues. Eq.1 shows Algorithm 1 output.

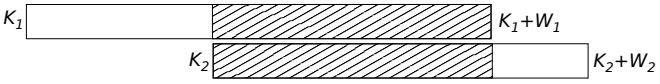
$$\begin{matrix} r_\phi(0,w_{min}) & \dots & r_\phi(k-w_{min},w_{min}) \\ r_\phi(0,w_{min}+1) & \dots & r_\phi(k-(w_{min}+1),w_{min}+1) \\ \vdots & \ddots & \vdots \\ r_\phi(0,w_{max}-1) & \dots & r_\phi(k-(w_{max}-1),w_{max}-1) \\ r_\phi(0,w_{max}) & \dots & r_\phi(k-w_{max},w_{max}) \end{matrix} \quad (1)$$

Algorithm 1 Tone Unit Detection (Search)

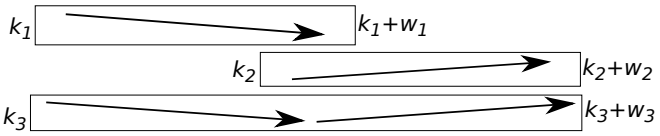
$J \leftarrow NUM\_OF\_DATASETS(\{s_j[k]\})$
$K_j \leftarrow LENGTH(s_j[k])$
for all $0 \le j \le J-1$ **do**
  for all $w_{min} \le w \le w_{max}, w \in \mathbb{Z}$ **do**
    for all $\phi_w(x) \in \{\phi_w^f, \phi_w^{rf}, \phi_w^{fr}, \phi_w^r\}$ **do**
      for all $0 \le k \le K_j - w$ **do**
        $r_\phi(k,w) = corrcoef(s_j[k:k+w], \phi_w(0:w))$
      end for
    end for
  end for
end for



(a) Horizontal overlap



(b) Vertical overlap



(c) Tone unit collision

Fig. 3: Cases for post-processing

Thus, each segment is defined by k, w, ϕ_w, and $r_\phi(k,w)$.

D. Post-Processing

Post-Processing (see Algorithm 2) is applied to all the segments and comprises 4 steps (see Fig. 3):

1) Tresholding
2) Resolving horizontal segment overlap for different k at fixed w
3) Resolving vertical segment overlap for different w
4) Resolving tone unit collision

The first three steps are applied to each group of segments referring to one tone unit ϕ_w, while the last step is applied only to segments where several tone units were found.

Tresholding checks the statistical significance of correlation. Tresholding parameter, $Q_{Treshold}$ sets the significance level, e. g. 0.95 or 0.98.

To locate model tone unit accurately k takes all the integer values in $[0, K_j - 1]$. For two neighbour values k_1, k_2 the corresponding $r_\phi(k_1,w)$, $r_\phi(k_1,w)$ will be very close to each other, because they are calculated over almost identical datasets leading to a significant redundancy of the output data. We call this issue "horizontal overlap". It is resolved now by keeping the only one segment with the largest $r_\phi(k,w)$

Algorithm 2 Tone Unit Detection (Post-processing)

for all $w_{min} \leq w \leq w_{max}$, $w \in \mathbb{Z}$ **do**
  for all $0 \leq k \leq K_j - w$ **do**
    if $r_\phi(k, w) \geq Q_{Treshold}$ **then**
      $CREATE\_SEGMENT(tone, k, w, r_\phi)$
    end if
  end for
end for
for all $SEGMENTS$ **do**
  $RESOLVE\_HOR\_OVERLAP(SEGMENT)$
end for
for all $SEGMENTS$ **do**
  $RESOLVE\_VERT\_OVERLAP(SEGMENT)$
end for
for all $SEGMENTS$ **do**
  $PRIORITIZE(SEGMENT)$
end for

for further processing among all the overlapping segments for given w, which are discarded.

"Vertical overlap", i. e. the overlap of segments with different k and w, is also possible. It is resolved in exactly the same manner. Again, only the segment with the largest $r_\phi(k, w)$ is kept for further processing.

The last step processes tone unit collision, i. e. the overlapping segments which correspond to different model tone units. In this case, the priority is given to "complex" units (p+ and r).

IV. Segment-to-word Mapper

Keyword search is performed by Segment-to-word Mapper, which operates with an ASR output text labelled with the timestamps and Tone Unit Detector output file containing the segments. The goal of Segment-to-word Mapper is to find a word that was pronounced during the given segment; this word is deemed to be a keyword. Partial coincidence between segments and word timestamps is allowed and can be set in Algorithm 3 by ratio parameter. Fig. 4 illustrates a fragment of Segment-to-word mapping results achieved for the online lecture The Great Reversal: The "Rise of Japan" and the "Fall of China" after 1895 as Historical Fables delivered by Benjamin Elman from Harvard University's Fairbank Center for Chinese Studies. Table I shows 35 keywords marked with proclaiming tones (fall and rise-fall) found by Segment-to-word Mapper in a 2-minute piece of lecture. The keywords are sorted in the same order as they are mentioned in the text; keywords given in boldface refer to Fig.4.

V. Results and Discussion

To summarize, an algorithm to process ASR output text for keywords by their prosodic features is presented. The first prototype has custom Tone Unit Detector and Segment-to-word Mapper, it also operates with pitch detection and speech

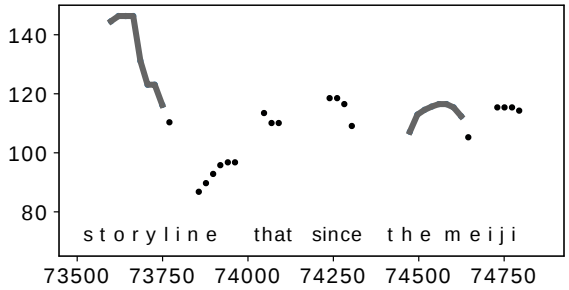


Fig. 4: Example of Segment-to-word mapping: words "storyline" and "meiji" are deemed to be keywords

Algorithm 3 Segment-to-word Mapping

for all $SEGMENTS$ **do**
  for all ASR_WORDS **do**
    $MAP(SEGMENT, ASR\_WORD)$
  end for
end for

recognition performed by third-party tools. As the result, a list of possible keywords is generated.

For our experiments we used a number of audio samples including academic lectures, presentation talks and news recordings. A particularly interesting case is the online lecture of B. Elman mentioned in Section IV and used for segment-to-word mapping evaluation. This use case refers to (not very common but still possible) situations when audio tracks are available with no explicit metadata describing the substance and the internal content of the recorded material. This, apart from obvious applications of the proposed algorithm and related tools, we can also consider solving a problem of mapping the processed recordings to a variety of external resources such as online encyclopedias, historical books, geographical maps, etc. In such a case the process of audio playback (together with prosody-based keyword extraction performed in background) can be enhanced by delivering additional visual and text information retrieved with using the extracted keywords.

Acknowledgment

This work is partially supported by the grant 17K00509 of Japan Society for the Promotion of Science (JSPS).

Authors would like to thank Karina Vylegzhanina, for if it was not for her we would not have taken up this research. Our discussions and work together have greatly influenced this paper.

References

[1] M. Scott and C. Tribble, *Textual patterns: Key words and corpus analysis in language education*. John Benjamins Publishing, 2006, vol. 22.
[2] B. Lott, "Survey of keyword extraction techniques," *UNM Education*, 2012.
[3] S. K. Bharti and K. S. Babu, "Automatic keyword extraction for text summarization: A survey," *arXiv preprint arXiv:1704.03242*, 2017. [Online]. Available: http://adsabs.harvard.edu/abs/2017arXiv170403242B

TABLE I: Segment-to-word Mapper output

Keyword	Tone	Segment (Tone Unit Detector)		Word (Speech Recognizer)	
		Start	End	Start	End
Involved	fall	575	703	410	830
Stories	fall	9557	9685	9390	9870
Tomorrow	fall	13567	13696	13470	14020
Remember	rise-fall	18773	18901	18520	18940
Beginnings	fall	24042	24170	23840	24300
Share	fall	27541	27669	27390	27710
World	rise-fall	29375	29503	29350	29530
Looks	fall	29568	29695	29540	29830
Fine	fall	32512	32639	32140	32860
Make	fall	33066	33194	33100	33250
Progress	fall-rise	33578	33706	33450	33900
Ultimately	fall-rise	40768	40895	40470	41050
Live	fall	41279	41408	41060	41500
China	fall	43263	43391	43110	43400
Rise	fall	45354	45482	45320	45620
Really	rise-fall	46122	46250	46060	46440
Narrative	fall	47402	47530	47320	47780
Make	fall	49984	50111	49990	50120
Endings	fall	53375	53504	53230	53610
Beginnings	fall	54037	54165	53870	54270
Japan	fall	64490	64618	64140	64710
Follows	fall	66624	66751	66540	66920
Educated	fall	67434	67562	67120	67770
Storyline	**fall**	73621	73749	73450	73750
Meiji	**rise-fall**	74496	74624	74460	74920
Ninety	fall	76565	76693	76540	76720
Five	fall	77311	77440	77150	77580
Power	fall	78826	78954	78740	79080
Empire	fall	82986	83114	82750	83300
Thousand	fall	97856	97984	97670	98040
Images	rise-fall	99690	99818	99670	100060
Leaving	rise-fall	102143	102272	102090	102400
Chinese	fall	103935	104064	103770	104120
Harvard	rise-fall	104810	104938	104740	105100
Interpreted	rise-fall	110570	110698	110220	110760
You	fall	117205	117333	117170	117400

[4] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," *Text Mining*, pp. 1–20, 2010. doi: 10.1002/9780470689646.ch1. [Online]. Available: http://dx.doi.org/10.1002/9780470689646.ch1

[5] Z. Xue, D. Zhang, J. Guo, and J. Hao, "Apparatus and method for extracting keywords from a single document," Mar. 30 2017, uS Patent 20,170,091,318.

[6] T. Ö. SUZEK, "Using latent semantic analysis for automated keyword extraction from large document corpora."

[7] S. K. B. Reddy Naidu, K. S. Babu, and R. K. Mohapatra, "Text summarization with automatic keyword extraction in telugu e-newspapers." doi: 10.1145/2980258.2980442. [Online]. Available: https://doi.org/10.1145/2980258.2980442

[8] T. Weerasooriya, N. Perera, and S. Liyanage, "A method to extract essential keywords from a tweet using nlp tools," in *Advances in ICT for Emerging Regions (ICTer), 2016 Sixteenth International Conference on*. IEEE, 2016. doi: 10.1109/ICTER.2016.7829895 pp. 29–34. [Online]. Available: https://doi.org/10.1109/ICTER.2016.7829895

[9] W. I. Grosky and T. L. Ruas, "The continuing reinvention of content-based retrieval: Multimedia is not dead," *IEEE MultiMedia*, vol. 24, no. 1, pp. 6–11, 2017. doi: 10.1109/MMUL.2017.7. [Online]. Available: https://doi.org/10.1109/MMUL.2017.7

[10] E. Pyshkin and V. Klyuev, "On document evaluation for better context-aware summary generation," in *Aware Computing (ISAC), 2010 2nd International Symposium on*. IEEE, 2010. doi: 10.1109/ISAC.2010.5670465 pp. 116–121. [Online]. Available: https://doi.org/10.1109/ISAC.2010.5670465

[11] S. Beliga, "Keyword extraction techniques," 2016.

[12] P. Meladianos, A. J.-P. Tixier, G. Nikolentzos, and M. Vazirgiannis, "Real-time keyword extraction from conversations," *EACL 2017*, p. 462, 2017.

[13] K. Elakiya and A. Sahayadhas, "Keyword extraction from multiple words for report recommendations in media wiki," in *IOP Conference Series: Materials Science and Engineering*, vol. 183, no. 1. IOP Publishing, 2017. doi: 10.1088/1757-899X/183/1/012029 p. 012029. [Online]. Available: http://dx.doi.org/10.1088/1757-899X/183/1/012029

[14] G. Alharbi, "Metadiscourse tagging in academic lectures," Ph.D. dissertation, University of Sheffield, 2016.

[15] D. Brazil *et al.*, *Discourse intonation and language teaching.* ERIC, 1980.

[16] D. Brazil, "Phonology: Intonation in discourse," *Handbook of discourse analysis*, vol. 2, pp. 57–75, 1985.

[17] M. Coulthard and D. Brazil, *The place of intonation in the description of interaction*. Linguistic Agency University of Trier, 1981.

[18] J. Six, O. Cornelis, and M. Leman, "Tarsosdsp, a real-time audio processing framework in java," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=17089

[19] D. M. Chun, "Signal analysis software for teaching discourse intonation," *Language Learning & Technology*, vol. 2, no. 1, pp. 61–77, 1998.

[20] A. Klapuri, "A method for visualizing the pitch content of polyphonic music signals." in *ISMIR*. Citeseer, 2009, pp. 615–620.

[21] Á. Abuczki, "Annotation procedures, feature extraction and query options," *of Electronic Information and Document Processing*, p. 81. doi: 10.1109/IEMBS.2008.4649799. [Online]. Available: https://doi.org/10.1109/IEMBS.2008.4649799

[22] P. Roach, "Techniques for the phonetic description of emotional speech," in *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*, 2000. doi: 10.1016/S0167-6393(02)00070-5. [Online]. Available: http://dx.doi.org/10.1016/S0167-6393(02)00070-5

[23] A. Meftah, Y. Alotaibi, and S.-A. Selouani, "Emotional speech recognition: A multilingual perspective," in *Bio-engineering for Smart Technologies (BioSMART), 2016 International Conference on*. IEEE, 2016. doi: 10.1109/BIOSMART.2016.7835600 pp. 1–4. [Online]. Available: https://doi.org/10.1109/BIOSMART.2016.7835600

[24] M. Warren, "A corpus-driven analysis of the use of intonation to assert dominance and control," *Language and Computers*, vol. 52, no. 1, pp. 21–33, 2004. doi: 10.1163/9789004333772_003. [Online]. Available: https://doi.org/10.1163/9789004333772_003

[25] J. K. Bock and J. R. Mazzella, "Intonational marking of given and new information: Some consequences for comprehension," *Memory & Cognition*, vol. 11, no. 1, pp. 64–76, 1983. doi: 10.3758/BF03197663. [Online]. Available: https://doi.org/10.3758/BF03197663

[26] A. De Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002. doi: 10.1121/1.1458024. [Online]. Available: https://doi.org/10.1121/1.1458024

[27] K.-F. Lee, H.-W. Hon, and R. Reddy, "An overview of the sphinx speech recognition system," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 1, pp. 35–45, 1990. doi: 10.1109/29.45616. [Online]. Available: https://doi.org/10.1109/29.45616

[28] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 1, pp. 13–18, 1979. doi: 10.1109/TASSP.1979.1163188. [Online]. Available: https://doi.org/10.1109/TASSP.1979.1163188