An Automatic Intonation Recognizer for the Polish Language Based on Machine Learning and Expert Knowledge

Mikołaj Wypych

Institute of Fundamental Technological Research
Polish Academy of Sciences, Warsaw, Poland
mik@polfonetika.com

10.21437/Interspeech.2005-578

Abstract

In the paper a new automatic intonation recognizer for the Polish language is presented. The recognizer design combines Machine Learning and expert knowledge techniques. Machine Learning is used in pitch stylization (Artificial Neural Network), speech alignment (external design based on Hidden Markov Model) and intonation decoding (Hidden Markov Model). Expert knowledge drives phonemization, syllabification, lexical accentuation and lemmatization. In the recognizer, a recently available intonation annotation system for Polish is used. The intonation annotation system and the related expert knowledge allowed for substantial reduction in the number of designed HMM decoder parameters. Software integration problems emerging from the number of modules comprising the recognizer are approached using an original software environment for speech processing. The environment has data-centric message-driven Blackboard-like architecture with an annotation graph as a shared memory.

1. Introduction

The early attempts to describe intonation independently of orthographic text can be attributed to Joshua Steele who, in the 18th century, transcribed intonation in the productions by the famous Shakespearian actor David Garrick (after [1]). Since then, several systems of intonation representation were invented varying from low-level representation of fundamental frequency trajectory to high-level linguistically motivated intonation annotations. The fundamental frequency extraction problem was addressed by several researchers in the last 50 years. Also, mid-level representations of intonation based on production and perceptual models of pitch have had a rich bibliography within the last 20 years. Recently, an interest has grown for automatic recognition of higher-level intonation representations, especially linguistically motivated intonation annotations.

Intonation annotation is known to be useful for Natural Language Understanding and Speech-to-Speech translation as it is connected to semantic and pragmatic features of natural language. The annotation can also improve Automatic Speech Recognition accuracy by the reduction of variability on the segmental level and independently by providing the means of modeling speech disfluencies. Finally, maybe the most straightforward applications of intonation annotation are speech research and didactics.

The automatic intonation recognizer is a unit that converts speech signals into intonation annotations. The use of underlying orthographic text is known to improve intonation recognition accuracy ([2], [3]). In parallel, as noted above, intonation annotation can be used to improve Automatic Speech Recognition accuracy ([5]). Therefore, some authors suggest that an intonation recognizer should not take advantage of orthographic text ([6]).

Several intonation (prosody) annotation systems have been proposed, amongst which the British School system and ToBI can be regarded as the most influential.

In recent years a few attempts to solve the problem of automatic intonation recognition using ToBI annotation have been reported ([2], [3], [7], [4]).

The present paper describes an automatic intonation recognizer for the Polish language based on a new British School intonation grammar model presented in [9]. The reason for modeling intonation solely rather than prosody in general is that intonation is known to be the single most important prosodic cue in Polish ([8]).

2. Elements of the Intonation Recognizer

The present intonation recognizer can be logically split into three functional blocks: text analyzer, pitch stylizer and intonation decoder. The text analyzer block takes unrestricted (unnormalized) orthographic text as input and produces an underlying sequence of phones, syllable boundaries and lexical accents. The pitch stylizer block takes a speech signal as input and additionally a sequence of phones and syllable boundaries from the text analyzer block and produces a floating point vector encoding the pitch course for each syllable. The intonation decoder, given lexical accents and floating point vectors of stylized syllables, searches for the maximum likelihood sequence of intonation annotation tags.

The use of lexical information in the recognizer is seen as an important asset here, as it establishes a link between acoustic and lexical cues of intonation. Future implementations of automatic intonation recognizers should be incorporated directly into the search stage and in such cases, the intonation recognizer should take advantage of the hypothetical word path associated with the current search state in the decoder.

The next subsections present a more in-depth description of the functional blocks of the recognizer.

2.1. Text analyzer

The text analyzer block can be seen as a generalization of an electronic pronunciation dictionary. For any character sequence given as input, the text analyzer produces a phonetic-segmental transcription, syllable boundaries, lexical lemmas (where applicable) and lexical accents. There was no electronic pronunciation dictionary available for Polish at the time of designing the present recognizer. In fact, such a dictionary is one of by-products of the research on the text analyzer block.

The Polish language is highly inflectional. For 120 thousand lexemes in a standard dictionary 2.5 million wordforms exists which makes all-wordform dictionaries hard to maintain. Phonetic properties of Polish spoken words depend significantly on neighboring words. These include inter-word assimilations, non-syllabic words that need to be appended to neighboring syllables and deaccentuation in various lexical collocations. The Polish language has substantial regularity in the relation between the orthographic text and the data that need to be produced by the text analyzer. As a result phoneticians have been able to formulate highly accurate, although sometimes complex algorithms answering most of the research problems associated with the block (excluding lexical accent rules). Considering the facts presented above, a design choice was made to implement the text analyzer based on expert-rule approach. The resulting text analyzer outperforms classical pronunciation dictionaries in terms of coverage, context sensitivity and and storage size. There are seven modules comprising the text analyzer: tokenizer, normalizer, lemmatizer, phonetizer, syllabifier and lexical accent assigner.

2.1.1. Tokenizer

The tokenizer is a simple module that splits the input orthographic text into tokens, distinguishing wordforms and non-lexical tokens.

2.1.2. Normalizer

The normalizer is a module that converts non-lexical elements to appropriate (sequences of) lexical elements. Currently, the module is capable of converting any number to corresponding wordforms including the appropriate inflection. The module is implemented as a cascade of Finite State Transducers (FSTs) compiled by means of the FSA6 toolkit described in [10].

2.1.3. Lemmatizer

The lemmatizer is a module that provides each wordform with lemma and Part-of-Speech information. The information provided by lemmatizer is essential for the subsequent lexical accent assignment. The lemmatizer is a reimplementation of the SAM lemmatizer first presented in [11]. The lemmatizer takes advantage of the affix stripping algorithm with affix tables and lemma tables based on the so-called Tokarski's index. The affix index consists of 16371 affixes and 64845 lemmas.

2.1.4. Phonetizer

The phonetizer is a module that converts normalized orthographic text into a sequence of phones. The present recognizer, described in [12], is based on a set of letter-to-sound rules proposed in [13]. The rules were tuned and a total number of 1254 exceptions to the rules were included. Additionally, rules were extended to support regional variants of pronunciation and various transcription systems. A compiler converting transcription tables with rule definitions and exceptions into FSTs was developed. The phonetizer achieves very high reliability estimated as 99,78% Word Error Rate on a normalized newspaper text excluding words taken from foreign languages (e.g. foreign names).

2.1.5. Syllabifier

The syllabifier is a module that splits a sequence of words into syllables. The syllabifier was based on expert rules proposed in [14]. The rules were improved by adding a set of morphologically-based exceptions. A dedicated formalism for syllabifier rules specification was defined together with a compiler converting the rules into FSTs. The syllabifier takes into account cross-word syllables encountered in Polish.

2.1.6. Lexical accent assigner

The lexical accent assigner is a module that assigns lexical accents to syllables taking into account wordform contexts. An original lexical accent typology for Polish was introduced with special concern for the applicability in intonation recognition. In general, each syllable is assigned two binary attributes describing the possibility of occurrence of a strong and nuclear intonation accent for the syllable. Lexical accent rules are defined in terms of a dedicated formalism and a compiler converting the rules into FSTs is provided.

2.2. Pitch stylization block

The pitch stylization block extracts intonationally relevant information from each syllable from the input speech signal. The block takes advantage of information provided by the text analyzer.

At the early stage of the development of the recognizer, a perceptual stylization module as described in [15] was used. The stylizer is implemented by means of Praat program ([16]) and takes advantage of built-in Praat's pitch extractor. One of the main problems with similar stylization algorithms is that errors made on earlier processing stages accumulate and are hard to recover in the later processing stages. A cascade of thresholding performed in the stylizer (especially by pitch tracking) could be avoided if replaced by more informed global maximum likelihood search in the decoder.

A decision was made to develop a dedicated pitch stylizer for intonation recognition. The main differences between the dedicated stylizer and the previously used stylizer are: syllable-wise stylization, adaptability, build-in speech aligner, better plausibility of the output for statistical modeling at the cost of lowering human interpretability, diminished accumulation of pitch extraction-related errors, decreased processing power consumption.

The dedicated pitch stylizer consists of the following modules: spectral transformer, pitch extractor, speech aligner and syllable stylizer.

2.2.1. Spectral transformer

The spectral transformer is a module that performs signal preprocessing and transforms the time-domain speech signal into a frequency-domain representation. The preprocessing routines include preemphasis, low-pass filtering (cutoff=2kHz), downsampling, DC removal and windowing (64ms Hamming window, 16ms step). In the subsequent stage the windowed signal is transformed using Fast Fourier Transform (FFT) or Wavelet Transform (WT). According to the characteristics of the pitch extraction module, presented in the next subsection, the application of Wavelet Transform improves extraction accuracy by providing clearer high frequency components. The preprocessor is implemented based on the Intel Performance Primitives (IPP) signal processing library which takes advantage of the Single Instruction Multiple Data (SIMD) instruction set increasing DSP processing power of the modern CPUs.

2.2.2. Pitch extractor

The pitch extractor parametrizes frequency-domain speech representation into three parameters: pitch height, loudness and harmonicity. A novel method based on frequency-domain comb filtering is used. The method employs gradient descent algorithm for the training of filter coefficients which is implemented by means of an Artificial Neural Network. Currently a multi-component sine wave is used for training the filtering coefficients but the design allows for real-time adaptation of parameters from the speech signal which will be the subject of the further developments. In the pitch extractor all the output parameters are normalized using the z-score method with mean and variance computed from the most recent 20 seconds of the speech signal.

2.2.3. Speech aligner

The speech aligner locates phone boundaries in the speech signal which allows for syllable-wise pitch stylization. The speech aligner module is an interface to a multi-lingual highly accurate Sonic Continuous Speech recognizer ([17]) developed at the Center for Spoken Language Research in Colorado. The Polish training database for the Sonic was prepared taking advantage of the text analyzer block presented in the paper. The Polish version of the Sonic Speech Recognizer was developed as a part of a larger project presented in [18].

2.2.4. Syllable stylizer

The syllable stylizer parametrizes the pitch trajectory over each syllable in the speech signal into a 6-element floating point vector. The stylization vector elements are as follows: a weighted averages of pitch and delta pitch at the initial and final part of the syllable (4 elements), weighted averages of pitch at maximum and minimum pitch values in the syllable (2 parameters). The weighed averages are computed using composed harmonicity and loudness parameters which forms a weighting coefficient.

The information stored in the stylization vector is sufficient to make possible discrimination amongst the most informative tunes allowed by the intonational grammar.

2.3. Intonation decoder

The intonation decoder block is based on two-stream Continuous Density Gaussian Mixture Hidden Markov Model. The extended pitch stylization and the text analysis block were meant to reduce data variability and in consequence the number of parameters in the decoder. The input data are grouped into two streams: a 2-dimensional lexical accent stream and a 6-dimensional pitch stylization stream. The lexical accent stream is modeled with 1-component Gaussian Mixtures with full correlation matrix. The pitch stylization stream is modeled with 4-component Gaussian Mixtures with a diagonal correlation matrix.

The topology of the HMM was designed on the basis of expert data found in [9]. The overall phrase structure presents as follows (in Bacchus-Naur notation): [wPT]{sPT}NT, where wPT is a non-accented weak pre-nuclear tune, and sPT and NT are both accented and named strong prenuclear and nuclear tunes respectively. The initial syllables of the accented tunes are the only accented syllables in a phrase. There are several types of sPT, wPT and NT tunes differing in pitch height, pitch contour shape and location with respect to neighboring tunes (see [9] for details). Each tune from the grammar is given a separate HMM consisting of two or three states. The resulting HMM is composed of 22 HMMs for individual tunes and contains 59 states in total. To further reduce the number of parameters, Gaussian Mixtures parameters are tied between similarly shaped tunes.

The intonation decoder is implemented using HTK toolkit ([19]).

3. Software integration

In view to the number of modules and non-serializable dependencies between the modules of the system, a Software Architecture for Language Engineering (SALE) was developed in parallel with the recognizer. For the definition and examples of other SALEs see [20].

3.1. General characteristics

The SALE developed with the intonation recognizer is called SLOPE (Spoken Language Open Processing Environment) and is to be published on a free software license when it reaches beta stage. SLOPE takes advantage of a data-centric software integration approach resembling a Blackboard architecture. Data-centric here means that software modules are not connected based on module identity but based on types of data produced/consumed. The major features of the SLOPE include: shared memory in the form of an annotation graph (similar to the one defined in [21]), ontology repositories for module integration (collections of data types associated with the meaning), data persistence, round-trip engineering development technology (using Universal Modeling Language), portability (implemented in Java, C and C++) and deployability, including a low-memory footprint and fast startup time.

3.2. The structure of the recognizer

Being implemented by means of SLOPE, modules of the recognizer communicate through a central data structure (annotation graph). An ontology repository is used for module coupling. Fig. 1 presents dependencies between modules based on consumed/produced ontologies.

After reading the diagram it should become clear that a basic pipeline architecture is not feasible for the recognizer's implementation.

4. Database, training and testing

The training database for the recognizer consists of a subset of PoInt speech corpus presented in [22]. Intonation annotations for the speech signal are taken from [9]. The resulting speech corpus consists of 397 intonational phrases annotated with orthographic text and intonational tags. Given the limited size of the corpus, a v-fold training procedure is used with the constant size of testing set equal to 98 phrases.

5. Tentative results

The preliminary performance tests were performed using procedures similar to those used in [2] or [3]. For each syllable in the test set, we check whether it is accented properly (including tune identity) and compute the percentage of properly accented syllables, falsely accented syllables and falsely deaccented syllables. The results are as follows: 78.2% syllables receive proper accentuation, 9.4% of syllables are falsely accented and 12.4% are falsely deaccented. The results cannot be easily compared with the results presented by other researchers

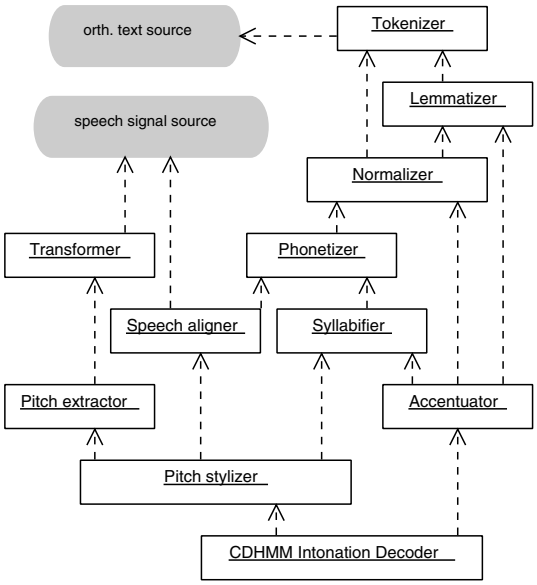

Figure 1: *Module dependencies in the intonation recognizer.*

because their findings refer to languages other than Polish and a different intonation system is used. It should be noted that the present recognizer is still under development and the presented results should be regarded as a baseline for further improvements.

6. Conclusion and future work

In the present intonation recognizer, expert knowledge and machine learning approaches are combined using specialized software architecture. There are three major profits from the extensive use of expert knowledge in the system: the first is an intonation annotation system, the second is a HMM decoder design and the third is a robust text processing module that replaces standard pronunciation dictionary with significant added value. The future plans in the development of the system include a more detailed evaluation and fine tuning, the extension of the training database using a semi-automatic procedure involving the present recognizer and, finally, improving the system adaptability.

7. References

[1] Gussenhoven, C., The Phonology of Tone and Intonation, Cambridge University Press, Cambridge, 2004.

[2] Conkie, A., Riccardi, G. and Rose, R. C., "Prosody Recognition from Speech Utterances Using Acoustic and Linguistic Based Models of Prosodic Events", In the Proceedings of Eurospeech 99, Budapest, 1999.

[3] Ananthakrishnan, S. and Narayanan, S. S., "An Automatic Prosody Recognizer Using a Coupled Multi-Stream Acoustic Model and a Syntactic-Prosodic Language Model", In the Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Philadelphia, 2005.

[4] Chen, K., Hasegawa-Johnson, M., Cohen, A. and Cole, J., "A Maximum Likelihood Prosody Recognizer", in the Proceedings of Speech Prosody 2004, Nara, 2004.

[5] Chen, K., Borys, S., Hasegawa-Johnson, M., "Prosody Dependent Speech Recognition With Explicit Duration Modelling at Intonational Phrase Boundaries", in the Proceedings of Eurospeech 2003, Geneva, 2003.

[6] Sung-Suk, K., Hasegawa-Johnson, M. and Chen, K., "Automatic Recognition of Pitch Movements using Time-Delay Recursive Neural Network", in: IEEE Signal Processing Letters, Vol. 11, 2004.

[7] Braunschweiler, N., Automatic Detection of Prosodic Cues, PhD Thesis, Universität Stuttgart, 2003.

[8] Jassem, W., Akcent języka polskiego, Wydawnictwo Polskiej Akademii Nauk, Warszawa, 1962.

[9] Jassem, W., Real and Potential Accent in Spontaneous Polish (in preparation).

[10] van Noord, G., FSA 6 toolkit, http://odur.let.rug.nl/~vannoord/Fsa/, 2004.

[11] Szafran, K., "Analizator morfologiczny SAM-95 - opis uzytkowy", Raport Instytutu Informatyki UW, TR 96-05(226), Warszawa, 1996.

[12] Wypych, M., Baranowska, E. and Demenko, G., "A Grapheme-to-Phoneme Transcription Algorithm Based on the SAMPA Alphabet Extension for The Polish Language", in the Proceedings of International Congress of Phonetic Science, Barcelona, 2003.

[13] Steffen-Batog, M. and Nowakowski, P., "An Algorithm for Phonetic Transcription of Orthographic Texts in Polish", Studia Phonetica Posnaniensia, vol. 3, Poznan, 1992.

[14] Jassem, W., Syllabe division rules for Polish (unpublished).

[15] Mertens, P., "The Prosogram: Semi-Automatic Transcription of Prosody based on a Tonal Perception Model", in the Proceedings of Speech Prosody 2004, Nara, 2004.

[16] Boersma, P., Weenink, D., Praat, http://www.fon.hum.uva.nl/praat/, 2004.

[17] Pellom, B. and Hacyoglu, K. Sonic: The University of Colorado Continuous Speech Recognier, Technical Report, CSLR, University of Colorado, Boulder, 2004.

[18] Dziubalska, K., Cole, R., Pellom, B., Sobkowiak, W., Wypych, M., Bogacka, A., Ma, J., Struemph, T., Krynicki, G., "The Use of Metalinguistic Knowledge in a Polish Literacy Tutor", in Proceedings of GlobE, Warsaw, 2004.

[19] Young., S., Kershaw, D., et. al., The HTK Book, Microsoft Corporation, 2004.

[20] Cunningham, H., Software Architecture for Language Engineering, PhD thesis, University of Sheffield, 2000.

[21] Bird, S. and Liberman, M., "A formal framework for linguistic annotation", Speech Communication 33 (1,2), 2001.

[22] Karpiński, M., "The Corpus of Polish Intonational Database (PoInt)", Investigationes Linguisticae VIII, Poznań, 2002.