UDC: 003.2:[004.51/52:004.934.5
Professional paper

NATURAL LANGUAGE PROCESSING AND TEXT-TO-SPEECH TECHNOLOGY

Valbon ADEMI[1], Lindita ADEMI[2]

Faculty of Natural Mathematical Sciences
Faculty of Philology

Abstract

Text-to-speech (TTS) technology is the process by which the computer is made to speak. It uses natural language processing concepts. Despite the advancement of technology that allows information to be stored electronically, textual information remains the most common way of exchanging information. Using text documents is problematic for visually impaired people in many scenarios, such as reading text on the move and accessing text under less-than-ideal conditions. The goal is to allow blind users to touch the printed text and receive the real-time transmission of the words. The development of such systems requires the use of such systems, requires the use of two technologies that are central to these systems, namely optical character recognition (OCR) to extract text information (Text Information Extraction) and text-to-voice (TTS) to convert this text in question. Text information extraction is the first and most important function of any assistive reading system and is an integral part of OCR because this process determines the intelligibility of the extracted word. Recent developments in computer vision, digital cameras, and computers make it possible to develop cameras and products that combine computer vision technology with other existing useful products such as optical character recognition systems used to recognize words. She can recognize characters, words, and sentences without any mistakes. OCR has a high recognition rate which is the electronic conversion of photographed images of typed or typed text into computer-readable text. Developments in computer technology make it possible to help these individuals by developing camera-based products. People with poor vision need portable assistance to read this printed text. The need to develop a voice-assisted text-to-speech system using the optical character recognition method with different sets of input and speech output is simulated.

Keywords: language, processing, synthesis, TTS, voice, visually impaired.

Introduction

The final stage in the text-to-speech system is the synthesizer responsible for producing the synthesized speech output. Several different synthesis techniques are presented, and in this chapter, we will look at concatenative synthesis, norm-based synthesis, and some other synthesis techniques such as MFM (Hidden Markov Model) based synthesis. In concatenative synthesis, small units of real recorded speech are fused together to form a final result. The best concatenation-based synthesizers are capable of rendering relatively natural synthetic speech. However, the disadvantage of this synthesis technique is that most of the speech contextual information is embedded in the data and the database size increases when different phonetic contexts have to be considered and stored in the database in such a way so that a natural sound of speech can be ensured. Norm-based norms or form synthesizers are mainly favored by phoneticians and phonologists because they constitute a cognitive and generative approach to the mechanism of phonation. Shape synthesizers have much less memory and requirements than contact systems, making them suitable for low-memory devices. The disadvantage of norm-based synthesis is that the sound is usually quite mechanical since the norms for controlling the synthesis are very difficult to develop. In MFM (Hidden Markov Model) based

synthesis the speech spectrum and excitation parameters are modeled in context dependent MFM and during MFM speech synthesis they are instantiated according to the input text. MFM-based speech synthesizers are capable of producing natural speech, and their memory requirements are also relatively small.

Concatenative Synthesis

Linking previously recorded natural pronunciation is probably the easiest way to produce clear and natural synthetic speech. However, the disadvantage is that the systems are usually limited to one voice and often require more memory than other methods. One of the most important aspects of concatenative synthesis is finding the correct length of the sound unit. The selection is usually a trade-off between longer and shorter units. Longer units achieve high naturalness, fewer concatenative points, and good coordination control, but increase the number of required units and memory space. Shorter units require less memory, but sample collection and labeling procedures become more difficult and complex. In current systems, units such as words, syllables, and sometimes triphones are commonly used. This section presents two popular concatenative synthesis strategies, namely: unit choice synthesis and bifunctional synthesis. Unit selection synthesis uses a large audio database with typically fixed unit sizes, for example, demilogs. The basic principle is to collect speech units from different phonetic and prosodic contexts and find the best pairing order of text input units. Another concrete synthesis technique we present here is based on diaphragm binding and uses a minimal sound database containing all diphthongs that occur in a given language.

Unit selection synthesis

The basic idea of the unit selection technique is that new natural utterances can be synthesized by selecting appropriate subwords from a natural word count database.

There are many prerequisites that must be met before the unit selection system can work. In unit selection, the system must be able to decide which units to select for synthesis in order to maximize the quality of the speech output in terms of clarity, naturalness, and other quality criteria. Therefore, systems try to find an optimal sequence of units that will minimize the cost of joining. Usually, the cost is divided into a target cost that indicates how close the unit is from the database to the desired one, and a continuity cost that indicates how well the two units are put together. The target cost is calculated for the duration and uses only the text features that can be calculated. Various typical characteristics have been proposed in the literature for the encoding of phonetic context, metrical structure and prosodic context of units. The continuum cost exploits all the properties of the candidate units, and is generally calculated as the Euclidean or Mahalonobian distance between the spectral features that represent the boundaries of the corresponding units. Continuity cost determination is usually computationally expensive, so it is advisable to calculate these costs offline and keep them in a lookup table. However, it is practically impossible to calculate and store the costs for all combinations of units over a large number of units, but since most combinations are very rare, it is sufficient to keep costs down continuously, for a small total database memory without sacrificing synthesis quality. The entire unit selection process is designed to optimally minimize both types of costs. This can be expressed by the following notation of the target cost Ct.

$$C^t(t_i,u_i)=\sum_{j=1}^{P} w_j^t C_j^t(t_i,u_i),$$

which is the weighted sum of the differences of the corresponding features. In this equation C is defined as the target cost, "t" is the third target unit, and it indicates the "i" unit in the unit's database. Similarly, the weighting factor used to calculate the target cost is the number of comparable characteristics. Further, we can define the continuity cost C as a weighted sum of the changes in characteristics between the merged units. This can be presented as:

$$C^C(u_{i-1},u_i)=\sum_{k=1}^{q} w_k^C C_k^C(u_{i-1},u_i).$$

where Cc is defined as a continuity cost, ui-1 is the unit (i - 1) and Ui; points to unit "i" in the units database. Similarly, Wck is the weighting factor used for the constant cost of CCk, a "q" is the number of different features being compared. The weighted sum of these two costs must be minimized to find the order of the best-paired units from the unit database. Each Ui unit; in the database is represented by a grid shape transition network, and the placement costs are given by measuring the unit distortion, i.e., the target cost, and the shape transitions are given by measuring the continuity distortion. The unit selection process resembles the Hidden Markov Model MFM (Hidden Markov Model) based on automatic speech recognition, but instead of using probabilities as in Automated Speech Recognition (ASR-NAF), the unit selection applies cost functions. The unit selection algorithm selects from the database the optimal sequence of units by finding a path through the shape transition network that minimizes them. For example, if the word to be synthesized is found in the database, the algorithm can select the entire word (if it minimizes the total cost) instead of selecting individual units. By choosing longer sequences, the system can reduce the number of segmental coupling points and does not have to worry about how best to combine, for example, different diphthongs or triphthongs.

Database of phonemes

Unit selection systems usually select from a limited set of units in the sound database and try to find the best path through the given set of units. When there are no examples of units that would be relatively similar to the target units, the situation can be observed either as a lack of database coverage or the desired sentence to be synthesized is not in the scope of the TVG system. Therefore, to achieve high-quality synthesis, the sound database must have good coverage. In the simplest sense, this means recording more data from the speaker, since with more data the database is more likely to contain an entity that is similar to the target and will have more continuity. good. On the other hand, the problem with increasing the size of the database is that there will always be gaps i.e. situations where there are no units that are similar or close to the target. This is due to the phenomenon of relatively frequent occurrence of rare events in language. In practice, this means that common occurrences in the language are very frequent, but there are so many rare occurrences that they are also flat. Also, for example, covering all contexts in even a few phrases is impractical as the size of the database would increase, which would be too much for the mass storage systems currently available. So instead of trying to collect a huge database, we try to select the "real data". By "real data" we mean that the items in the database

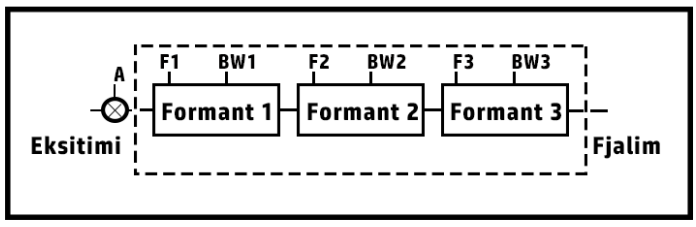
will cover the identified acoustic and phonetic space of the language relatively well. There are many suggestions for database recording and pronunciation recording. For example, one solution is to first model the acoustic space of the speaker, and then find units that are acoustically distinct and frequent enough to merit inclusion in the database. In this method, a cluster tree is first made from a general sound database with good phonetic coverage. Once the tree is created, the number of occurrences of each cluster is calculated using the domain's typical pronunciations, and finally, the top scores and coverage pronunciations are selected. This results in a manageable set of results and thus the database provides better synthesis quality (relative to database size) than poorly designed databases. Most unit selection systems use a fixed unit size, but longer boundary segments can be selected thanks to the selection attribute. The size of the sound database is often reduced by using different methods of encoding the stored sound units. Another possibility is to reduce the number of stored units, and different selection methods will be used to find a balance between the size of the database and the quality of the synthesized speech.

Synthesis of Diaphone

Compared to the unit selection technique, diaphonic synthesis uses a minimal sound database containing all diaphones occurring in a given language. In diaphonic synthesis, only one copy of each diaphone is stored in the speech database. For sentence duration, the target sentence projection is set to these minimal units by means of digital signal processing techniques, such as predictive linear coding, Pitch Synchronous Overlap Add - PSOLA, or MBROLA. Crosstalk synthesis usually suffers from pitch distortion at the connection points, so the resulting speech quality is generally not as good as that of unit choices, but it is therefore more natural-sounding than formant synthesizers.

Formant synthesis

There are two basic filter structures in the formant synthesizer, parallel and cascaded, but a combination of the two is usually used for better performance. In theory, formant synthesis also provides an infinite number of sounds (or sound units), making it flexible. Usually, at least three formants are needed to produce clear speech. Each formant is modeled with a bipolar resonator that allows you to specify both the formant's frequency and its width. Norm-based formant synthesis is based on a set of language/speech-specific norms used to define all the parameters needed to synthesize a desired translation. Some typical parameters used in current formant synthesis systems include: fundamental frequency, open pitch excitation coefficient, pitch, formant frequencies and their amplitudes, additional low frequency and high frequency resonator. The cascading formant synthesizer is shown in Figure 3.5 and consists of band resonators connected in series. In the cascade structure there is only one amplitude control (A), and the relative intensities of the formants are determined by their frequencies (F1, F2, F3) and range (BW 1, BW2, BW3). The output of each resonator formant is then used as the input to the next resonator. The cascading structure seems best for non-nasal sounds. However, generating friction and vibration sources is difficult with cascading structures.



The excitation signal is applied to all formants simultaneously and the results are summed together. Adjacent outputs of the shape resonators must be summed in opposite phases to avoid zeros or antiresonances in the frequency response. The parallel structure allows control of the range (BW 1, BW2, BW3) and gain (A1, A2, A3) for each shape (F1, F2, F3) separately, and has been found to provide better quality for the nose, fraction and stopping constants.

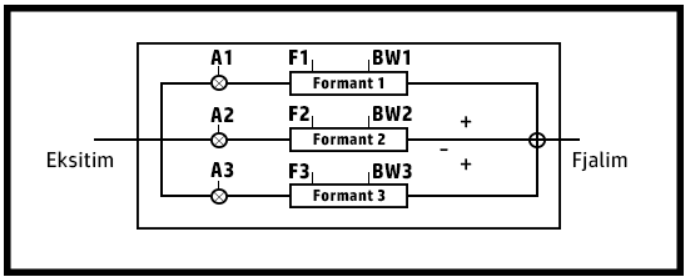


Figure 3.6 Synthesizer parallel format

When applying a formant synthesizer to a Text-to-Speech system, cascading and parallel patterns are combined to provide better quality.

Articulatory Synthesis

The basic idea behind articulatory synthesis is to produce synthetic speech directly by modeling the human articulatory system. This means that a mathematical model is defined for each organ of the human articulating system. So, there are different models for the lungs, vocal cords, vocal tract, tongue, lips, etc., and with their help to model human speech production as closely as possible. Due to the precise modeling of the human articulatory system, articulatory synthesis should theoretically be a good method for producing very natural-sounding speech. However, the problem with articulatory synthesis lies in the complexity of implementation, for example, it is difficult to collect data on articulatory development, and computational efficiency requirements. Therefore, due to such requirements, articulation synthesis is not widely used in real systems.

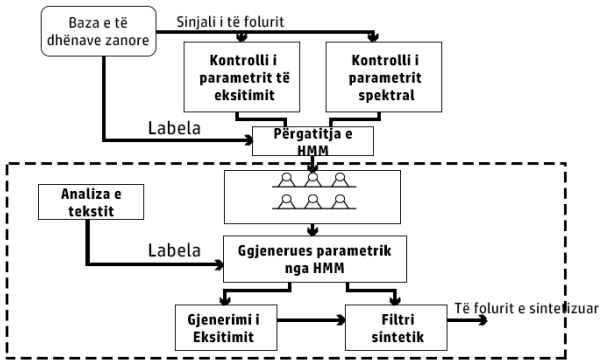
Based methods of linear predicates

Similar to formant synthesis, basic linear predictive coding (LPC) is based on a filtered speech model at the source, and the filter coefficients are automatically calculated from the natural speech frame. The basis of the linear prediction is that the instantaneous speech sample y(n) can be computed from a finite number of previous samples y(n-1) to y(n-p) with a linear combination of a small error e (n). This results in the spoken sample y(n) being represented as:

$$y(n)=\sum_{k=1}^{p} a(k)y(n-k)+e(n)$$

where "p" is the linear predictor order, and a(k) are the linear predictor coefficients obtained by minimizing the sum of the squared errors in the frame. Two methods, the covariance method and the autocorrelation method, are most commonly used to calculate these coefficients, but only with autocorrelation is the stability of the filter guaranteed.

In the synthesis stage, the excitation used is approximated by a series of pulses during speech sounds and by random noise during non-speech sounds. The excitation signal is amplified and filtered with a digital filter for which the coefficients a(k) is and they are usually updated every 5-10 ms. The order of the filter is usually between 10 and 12 at the sampling rate of 8 kHz, but for higher quality, at the sampling rate of 22 kHz, the order is usually between 20 and 24. The main disadvantage of the standard LP method is that it presents a complete pattern, meaning that segments containing antformants (e.g. nasal vowels and nasalized vowels) are poorly patterned. The quality is relatively poor for short skirts, which have a shorter period than the frame size used for analysis. However, modifications and extensions to the basic LP model have been introduced to improve synthesis quality. An example is the Warped Linear Prediction (WLP) model which takes advantage of the characteristics of human hearing by reducing the much-needed filter line of 20-24 for 22 kHz synthesis to 10-14. The basic idea is that the holding units in the filter are replaced by "a11-pass" sections. Depending on the warping feature used, WLP provides better low-frequency resolution and worse high-frequency resolution, which is however similar to the characteristics of human hearing. Several other variations of linear prediction have been developed to increase the quality compared to the basic model. With these methods, the excitation signal used is different from that in the standard LP method. Some examples are: Multi-pulse Linear Prediction (MLPC), where the excitation is built up from several pulses, Residual Excited Linear Prediction (RELP), where the error or repetition signal is used as the excitation signal and the speech signal can be accurately reconstructed, and Code Excited Linear Prediction (CELP), in which a limited number of excitations are stored in a limited book.



Synthesis based on Hidden Markov Model (MFM)

Although many Text-to-Speech systems can synthesize speech with acceptable quality, they are not capable of synthesizing speech with different voice characteristics such as individualities and speaker emotions. To obtain different sound characteristics in Text-to-Speech systems based on the selection and association of acoustic units, a large amount of sound data is required. However, it is relatively difficult to collect and segment large amounts of speech data for different languages. Moreover, it is not possible to store large databases on devices with low memory. Due to these aspects, to design a voice synthesis system that can generate various voice characteristics without large voice data, it is proposed to use Hidden Markov Model (MFM), Hidden

Markov Model-HMM. In the MFM-based speech synthesis system shown in Figure 3.7 the frequency spectrum (vocal tract), fundamental frequency (vocal source, i.e. excitation), and speech duration are simultaneously modeled with MFM. At synthesis time, MFM itself generates waveforms based on maximally similar criteria. The spectrum part of the MFM output vector is usually based on the "mel-cepstral" coefficients including zero coefficients and their first and second derivatives. Similarly, the temporal structure of speech, in other words, the MFM shape, is modeled using multivariate Gaussian distributions. During speech synthesis, the filter is controlled by an output vector MFM, i.e. mel-cepstral coefficients. One solution is to apply a mel-cepstral analysis technique, which allows speech to be resynthesized directly from the mel-cepstral coefficients using the Mel Log Spectrum Approximation (MLSA) filter.
MFM is also used to model the fundamental frequency F0, and the observation sequence for this is composed of one-dimensional values and a discrete symbol indicating whether the phoneme is spoken or not spoken. Therefore, conventional discrete or continuous MFM cannot be used to model F0, and to model such observation sequences, MSM-based Probabilistic Multi-Spatial Distribution (MSD-NMM) is proposed. Many contextual factors the effect of speech spectrum, fundamental frequency pattern, sound duration, and context-dependent MFM are used to cover all these effects. However, as the number of contextual factors increases, the number of possible combinations also increases exponentially, and therefore it is not possible to calculate exactly with a limited amount of training data. To overcome this problem, context-based decision tree clustering is applied to MFM-based automated voice recognition and speech synthesis. Furthermore, these techniques apply to MSD-HMM.

Conclusion

During speech synthesis, an MFM corresponding to the input text is constructed by concatenating the context-dependent MFM. The shape length of the constructed MFM is determined by maximizing the likelihood of the shape duration output. Similarly, the sequence of "mel-cepstral" coefficients and logF0 values including the discrete parameter speech / no response is determined by maximizing the algorithm for generating the speech parameter. Finally, the speech wave is generated directly from the "mel-cepstral" coefficients and F0 values using an MLSA filter. In the synthesis technique based on MFM, the speech characteristics can be changed by modifying the MFM parameters. It has been shown that the characteristics of synthesized speech can be changed by applying a speaker adaptation technique, a speaker interpretation technique, or an (eigenvoice) technique. In addition, in MFM synthesis, adaptive techniques can also be used for language adaptation. MFM can be trained by applying several monolingual corpora from different languages resulting in a multilingual synthesizer. During synthesis, models can be fitted to a particular speaker by applying, for example, MLLR, the Maximum Likelihood Linear Regression fitting model.

References

[1]. Valbon.Ademi -"Напредни техники за развој на интелегентни кориснички интерфејси", 2018 Tetovë
[2]. AlexandreTrilla and Francesc Alías. (2013)„„Sentence-Based Sentiment Analysis for Expressive Text-to-Speech“, IEEE Transactions onAudio, Speech, and Language Processing, Vol. 21, Issue.
[3]. Computer Aided Text to Speech andText to Braille System - Visually Impaired (Shruti Drishti)- West Bengal
[4]. https://www.shvsh.org.al/media/pdfFiles/Te%20njohesh%20brailin.pdf
[5]. https://boniproduction.at.ua/index/ergonomia/0-48
[6]. https://www.orcam.com/en/about/
[7]. https://www.orcam.com/en/myeye2/
[8]. https://www.theverge.com/2017/7/12/15958174/microsoft-ai-seeing-app-blind-ios
[9]. Asakawa C., What's the web like if you can't see it? In Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A). pp. 1–8. 2005. http://dl.acm.org/citation.cfm?id=1061813
[10]. Blenkhorn P., Evans, G., Architecture and requirements for a Windows screen reader. In Speech and Language Processing for Disabled and Elderly People (Ref. No. 2000/025), IEE Seminar on. pp. 1–1. 2000. http://ieeexplore.ieee.org/xpls/abs_all.jsp? arnumber=846939.
[11]. Borodin Yevgen, Bigham P. Jeffrey, Dausch Glenn, Ramakrishnan I.V., More than Meets the Eye: A Survey of Screen-Reader Browsing Strategies, In Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A), ACM New York, 2010.
[12]. Clark A.J. Robert, Richmond Korin, King Simon, Festival 2 - build your own general purpose unit selection speech synthesiser. In Proc. 5th ISCA workshop on speech synthesis, 2004.
[13]. Designing the Moment: Web Interface Design Concepts in Action by Robert Hoekman, New Riders, 2008.
[14]. Feldman R., Sanger J., The text mining handbook - advanced approach in analyzing unstructured data, Cambridge Press 2007.
[15]. Galitz O. Wilbert, The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques (CourseSmart) , Wiley Publishing 2007.
[16]. Gormez Zeliha, Orhan Zeynep, TTTS: Turkish text-to-speech system, 12th WSEAS International Coneference on Computers, Heraklion Greece, pp. 977-981, 2008.
[17]. Johnson Jeff, Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules, Elsevier, 2010.
[18]. Laurel Brenda, The Art of Human-Computer Interface Design, Morgan Kaufmann Series, 1990.
[19]. Mayhew J. Deborah, Principles and Guidelines in Software User Interface Design, Prentice Hall, 2008.
[20]. Saffer Dan, Designing Gestural Interfaces: Touchscreens and Interactive Devices, Oreilly, 2008.
[21]. Scott Bill, Neil Theresa, Designing Web Interfaces: Principles and Patterns for Rich Interactions, O'Reilly, 2009.
[22]. Shao Ling, Shan Caifeng, Luo Jiebo, Etoh Minoru, Multimedia Interaction and Intelligent User Interfaces: Principles, Methods and Applications (Advances in Pattern Recognition Springer, 2010.
[23]. Banjanin M., „Komunikacioni inžinjering**”**, Saobraćajno-tehnički fakultet, Doboj, 2007.