NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Electronics and Telecommunications

Examination paper for TTT4185 Speech Technology

Academic contact during examination:: Torbjørn Svendsen
Phone: +47 930 80 477

Examination date: Monday 9. December 2013

Examination time (from - to): 09.00 - 13.00

Permitted examination support material: **C** – Specified, written and handwritten examination support materials are permitted. A specified, simple calculator is permitted

Other information:

- The examination consists of 3 problems where
  - problem 1 concerns automatic speech recognition
  - problem 2 concerns speech analysis
  - problem 3 concerns speech synthesis
- All sub-problems counts the same
- All problems are to be answered
- Grades will be announced 3 weeks after the examination date.

Language: English

Total number of pages: 9

Of this, number of enclosure pages:

Checked by:

Date Signature

Problem 1

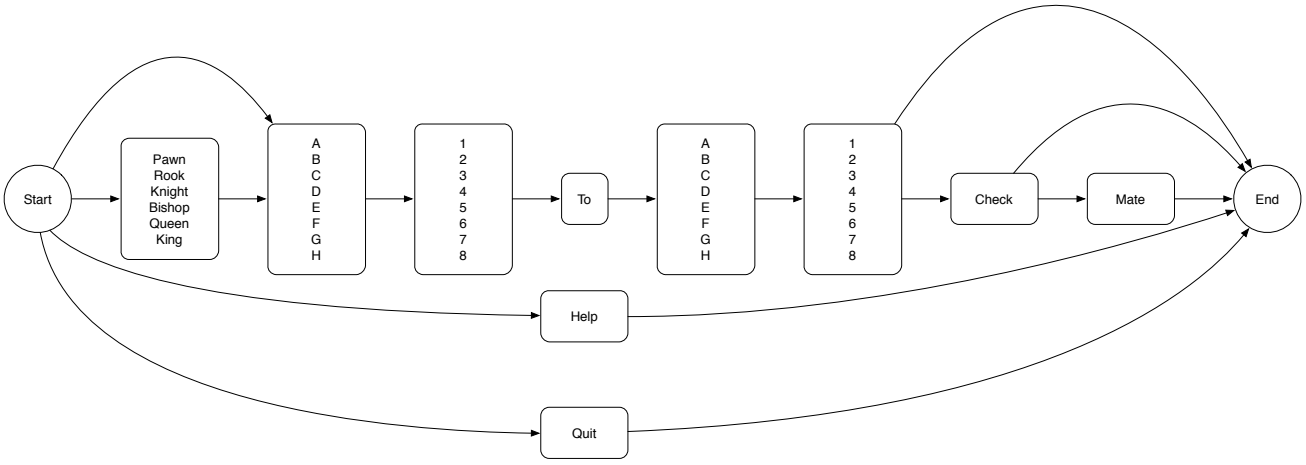
You have been given the task of designing a speech based user interface for a chess application. In the game of chess, each piece is located on a square with coordinates denoted by <Letter+Number>, e.g. *A2*. The user interface shall allow players to utter the start and stop position of their next move. In addition, the player has the choice of naming the piece to be used before the move itself, and the statements *check* or *check mate* can be uttered after the move. The player can also end the game with *quit* or ask for help with *help*. Here are some examples of legal commands:

- G3 to G5.
- Pawn, G3 to G5.
- Bishop, F1 to A6, check.
- F1 to A6, check mate.

The chess pieces are named pawn, rook, knight, bishop, queen and king. The chess board is eight by eight squares with coordinates <A-H,1-8>.

1a) Draw a figure of a finite state machine grammar that only allows command phrases that are valid for the user interface. How many legal command phreses are there? (Hint: It's a finite number)

Example figure. Note that lines between boxes with multiple words denotes all-to-all edges:



Number of possible utterances:

- Help/Quit: 2.
- <Piece/No piece>: 7.
- <Letter><Number> to <Letter><number>: $8^4 = 4096$.
- <Nothing/Check/Check Mate>: 3.

In total we have $2 + 7 \times 4096 \times 3 = 86018$ different utterances.

1b) Assume you have the choice between the following recognition units – words, syllables and phonemes. Discuss the pros and cons of these units in a general setting. Which recognition unit would you choose for the user interface, and why?

There are mainly three characteristics we want a speech unit to have: accuracy, trainability and generalizability.

- Words
  - High accuracy, since the all coarticulation effects are internal to the word.
  - Low trainability since the number of words are is very high.
  - Low generalizability as a new data needs to be collected and models trained if new words are to added
  - Suited for tasks with small vocabularies
- Syllables
  - High accuracy, since the strongest word-internal coarticulation effects are internal to the syllables.
  - Low trainability since the number of syllables that can occur (especially if proper names are included) is very high.
  - Good generalizability.
  - Suited for tasks with abundant training data.
- Phones are the most common speech unit in current systems, and can be further divided into two types:
  - Context independent phones
    * Low accuracy since coarticulation effects are not modeled.
    * Very high trainability since the number of phonemes is low (40-50).
    * Very high generalizability.
    * Suited for tasks with little available training data and little information about the kinds of input the system will receive.
  - Context dependent phones
    * High accuracy since coarticulation effects over triphones are modelled.
    * Low trainability since the number is very high. (However, clustering states where different contexts give rise to similar coarticulation effects can reduce this number considerably.)
    * Good generalizability.
    * Well suited for all tasks where there is sufficient data available to train them.

For small vocabularies, words can be a good choice. For a limited vocabulary and ample training data, syllables are a good choice. For bigger vocabularies, phones are generally the better choice, particularly if they are modeled in context. In this case, words may very well be the best choice.

1c) The grammar from 1a) yields a finite number of valid sentences W. Given an observation $X = \{x_1, \ldots, x_N\}$, we can compute the conditional probability $P(W|X)$ for an arbitrary sentence W. Show how this is done using the Forward algorithm. Explain how this can be used to build a functional, but inefficient, speech recognizer.

We know that

$$\underset{W}{\operatorname{argmax}} P(W|X) = \underset{W}{\operatorname{argmax}} P(X|W)P(W).$$

Since we have a non-probabilistic grammar, $P(W)$ is either zero or one, with $P(W) = 1$ indicating a legal sentence.

The forward algorithm is based on the recursion,

$$\alpha_t(i) = b_i(X_t) \sum_j a_{ji}\alpha_{t-1}(j),$$

where $b_i(\cdot)$ is the pdf of state i, a_{ji} is the transition probability from state j to state i, and X_t is the observation vector at time t. The algorithm is initialized by

$$\alpha_1(i) = b_i(X_1).$$

Using the recursion to compute $\alpha_t(i)$ for all i with t going from 2, 3, 4 up to T, we have

$$P(X|W) = \sum_i \alpha_T(i).$$

In principle we can run the forward algorithm for all W and compare and pick the sentence that yields the largest $P(X|W)$. This is very inefficient however, as there are 86018 sentences that needs to be compared every time.

1d) Explain how Bayes rule can be used to rewrite $P(W|X)$ in the form of an acoustic model and a language model.

Bayes rule:

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)}$$

Since we maximize over the sentence W, we can simplify this as

$$\underset{W}{\operatorname{argmax}} P(W|X) = \underset{W}{\operatorname{argmax}} P(X|W)P(W),$$

as $P(X)$ is independent of W.

The Viterbi algorithm finds the sequence of states $\hat{S}$ and corresponding sentence $\hat{W}$ that maximizes $P(W, S|X)$. Why is $\hat{W}$ a good estimate of the sentence that maximizes $P(W|X)$?

We rewrite our probability of a sentence given the observations as

$$P(W|X) = \sum_S P(W, S|X) = \frac{\sum_S P(X|S, W)P(W, S)}{P(X)}$$

If there is one path S' so that $P(W, S'|X) >> P(W, S|X) \forall S$, then we assume that

$$P(W|X) = \sum_S P(W, S|X) \approx P(W, S'|X)$$

1e) Assume that one initially considers a word-based acoustic model. All words are represented by a 10-state hidden Markov model with a left-right topologi and no skipping of states possible. The feature vector has dimension 40, and we have 10 Gaussian mixture components with diagonal covarience matrices per state. How many parameters will each word model contain?

The parameters consists of mean vectors, covariance matrices, state transition probabilities and mixture weights. There are no initial state probabilities due to the strict left-right properties.

- Number of states transition probabilites: 10
- Number of mixture weights: $10 \times 10 = 100$.
- Number of mean vector components: $10 \times 10 \times 40 = 4000$
- Number of covariance components: $10 \times 10 \times 40 = 4000$

All in all, 8110 parameters per word.

1f) It turns out to be too expensive and difficult to record speech data for the whole word model by yourself. Instead, a general purpose database containing speech data will be purchased. What are the conditions this database needs to meet to be suitable for building a speaker independent speech recognizer? Is using words as a recognition unit still a viable alternative?

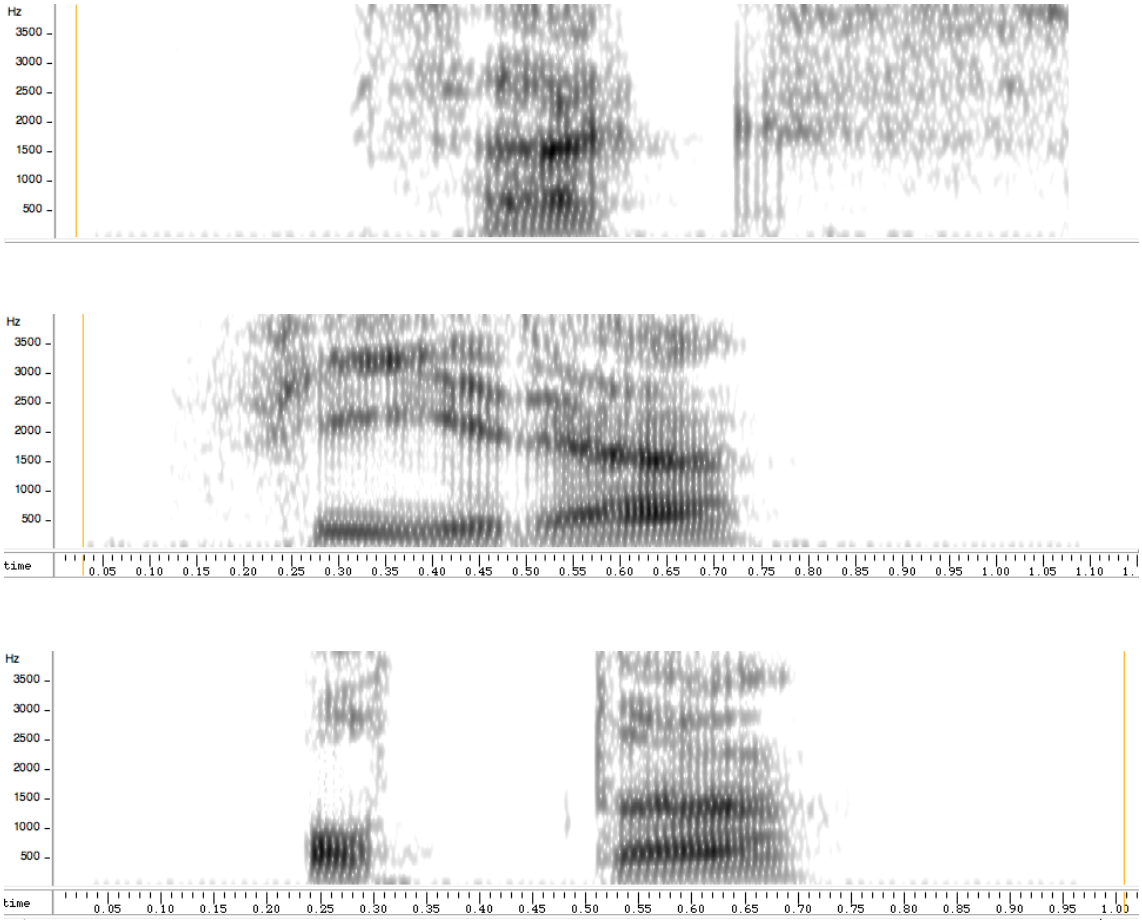
The database should provide good *coverage* of speakers in terms of gender, dialects, etc. The recording conditions, e.g. sampling rates, are also of importance.

Since it is unlikely to find an abundance of exact words used in the user interface, we can not use whole words as recognition units, and should instead use phonemes.

1g) Only 1% of the database is manually transcribed on the phoneme level with proper time boundaries. The rest of the database is only transcribed on a sentence level. Describe the step by step procedure you would follow to build a phoneme based acoustic modell using this database. No mathematical description is necessary, but all techniques used should be explained briefly.

- Use the transcribed data to create single mixture monophone models.
  - The start, middle and end states are initialized by dividing every transcribed phoneme into three equal parts and then computing the mean vector and covariance matrix.
  - Use forward/backbard algorithm to update the monophones until convergence.
- Convert sentences to phoneme strings using the pronunciation lexicon.
- Use forward/backbard algorithm to update the monophones until convergence.
- Increase the number of mixtures in the monophone models
  - Split every mean vector in two, effectively double the number of mixtures or,
  - Split the mean vector of the mixture with largest mixture weight, adding one mixture
  - Use forward/backbard algorithm to update the monophones until convergence.

Problem 2

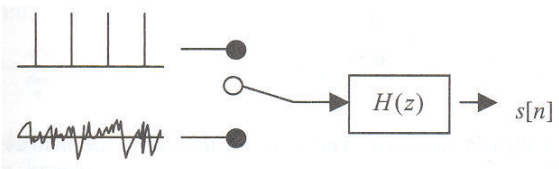


2a) The three figures above shows the spectrograms of the three Norwegian words ”fire”, ”seks” and ”åtte” (the digits four, six and eight). Which spectrogram corresponds to which word and why?

”Seks” and ”åtte” are easy to separate from ”fire” due to the silence after the ”k” and ”t”. They are themselves separable by observing that ”seks” starts and ends with unvoices sounds (no formants in the spectrogram), while ”åtte” starts and ends with a voiced sound.

2b) Draw the source/filter model and explain why it is an adequate model for speech. Which type of phonemes is it particulary well suited for, and for which phonemes is it less so?

The source-filter model:



Time varying filter represents transfer function of vocal tract. Excitation is pulse train (voiced sounds) or pseudo-random noise (unvoiced sounds). The gain control models volume (or loudness).

Physically, the voiced sounds are produced by periodically opening the vocal cords to let a pulse of air through, while unvoiced sounds are produced by constrictions in the vocal tract (or at the lips), resulting in a turbulent, noise-like air-flow.

It is well suited for Voiced and unvoiced phonemes, but not so much for plosives.

2c) Explain the following concepts: Formant, spectral envelope and pitch.

- Formants are the acoustic resonances of the speaking apparatus, or the spectral peaks of the sound spectrum of the voice.
- The spectral envelope is the amplitude gain as a function of frequency which corresponds to the filter in the source-filter model.
- The pitch is the fundamental frequency of speech and is defined for voiced sounds. The pitch depends on the rate of the train of glottal pulses.

2d) The computation of linear prediction coefficients (LPC) is a much used technique for speech analysis. What part of the source/filter model are we trying to represent using LPC?

The filter.

Given that we have a short section of a vowel – what happens with the spectral envelope if we use way too few or too many coefficients in our LPC analysis?

Using too few coefficients will result in a smoothed spectral envelope, where some spectral peaks that are close will be smeared together. Using too many will eventually start representing the source.

2e) Mel-frequency cepstral coefficients (MFCC) are a much used technique for extracting feature vectors for use in speech recognition. Explain how we go from a short window of sampled speech to an MFCC-vector using a series of processing blocks.

- DFT: Computes the Discrete Fourier transform of the signal.
- $|\cdot|^2$: Squared absolute value. Corresponds to the energy in the frequency band.
- MEL FB+Energy computation: Use Mel-scale frequency filter banks to weigh and sum the total energy in different bands. The Mel-scale filters will be wider as frequency increases, modeling the sensitivity of the ear for different frequencies.
- log(·): Compute the log-energy.
- DCT: Discrete cosine transform. Decorrelates the log-energy vector, facilitating the use of diagonal covariance matrices.

Problem 3

3a) What are the four main processing blocks that consitutes a speech synthesis system? Briefly explain the functionality of each block.

- Text analysis: text normalization; analysis of document structure, linguistic analysis
  Output: tagged text
- Phonemic analysis : homograph disambiguation, morphological analysis, letter-to-sound mapping
  Output: tagged phone sequence
- Prosodic analysis: intonation; duration; volume
  Output: control sequence, tagged phones
- Speech synthesis: voice rendering
  Output: synthetic speech

3b) Describe the principles behind diphone speech synthesis and data driven concatenative synthesis.

Diphone synthesis uses a minimal speech database containing all the diphones (sound-to-sound transitions) occurring in a language. In diphone synthesis, only one example of each diphone is contained in the speech database. At runtime, the target prosody of a sentence is superimposed on these minimal units by means of digital signal processing techniques such as linear predictive coding, PSOLA or MBROLA.

Unit selection synthesis uses large databases of recorded speech. During database creation, each recorded utterance is segmented into some or all of the following: individual phones, diphones, half-phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a "forced alignment" mode with some manual correction afterward, using visual representations such as the waveform and spectrogram. An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighboring phones. At run time, the desired target utterance is created by determining the best chain of candidate units from the database (unit selection).

Under which conditions would you choose to use diphone speech synthesis?

- Cost. If one is to develop the voice, unit selection synthesis is much more expensive and time-consuming.
- Comptutional complexity. Diphone synthesis is much more lightweight both in terms of storage requirements and computation.

3c) Explain the phenomenon *Large Number of Rare Events* and explain its impact on data driven concatenative synthesis.

- Large number of units with small probability of occurrence
- If database units are selected randomly, the probability of encountering a unit not in the database approaches certainty for a small sequence of randomly selected sentences.

- Unit inventory must be chosen with care
- Fall-back solutions must exist for non-covered units

3d) Explain how one can use the PSOLA-algorithm to change the pitch of a speech signal.

PSOLA can modify pitch and duration. The starting point for PSOLA is having accurate estimates of the fundamental frequency in the voiced parts of the speech signal. For every vocal pulse you can then construct a waveform segment centered around the pulse. The segments are attenuated towards the end points and typically extend over two fundamental periods (2T0). Through the additive combination of partially overlapping segments, where the degree of overlap is such that the distance between successive vocal pulses is equivalent to the desired fundamental frequency, a voice signal with the same spectral envelope but a new fundamental frequency is constructed. By repeating or skipping segments from the original signal, the manipulation can be done without changing the duration of the signal.