Feedback
Speech Processing, first assignment, November 2018

This year's theme is anaphoric reference, and in particular the use of the **pronoun referent**

"this"

to refer to a previously-introduced idea.
"This" is a common cause of ambiguity in students' writing.

I skimmed all submitted lab reports, and found examples of ambiguous "this" in more than half of them. I picked the first one that I found in each report.

The format of the following examples is:

The extracted sample, possibly cleaned up a little, with citations mostly removed to save space.

Explanation of the ambiguity

A suggested unambiguous version

The suggested text is not necessarily perfect - it might still contain errors that were present in the original version. Use it principally to learn how to avoid ambiguous "this".

A problem that can occur at this stage is homograph ambiguity, where one word can have multiple meanings (e.g. “record” can be a verb and a noun), as will be explored later in the paper. **This** means that POS tagging errors arise .

“This” could refer to: “A problem”, “homograph ambiguity”, “where one word can have multiple meanings”.

One orthographic word (e.g. “record”) can have multiple meanings (e.g., a verb or a noun). This is called *homograph ambiguity*, which will be explored in Section X.Y.Z. Ambiguity makes POS tagging non-trivial and there will always be some errors.

G2p converts a sequence of orthographic characters to a sequence of phones using a training corpus with transcriptions. **This** is done probabilistically by finding the most probable phone P, given a letter sequence L.

“This” could refer to: “G2p converts”, “using a training corpus”.

G2P converts a sequence of letters to a sequence of phones, using a model trained on a corpus of words and their pronunciations, i.e., a dictionary. The model is used to find the most probable phone sequence, given the letter sequence.

The second method is the use of a probabilistic model, looking at the POS tag assigned to the preceding and following words in the previous step and using **this** to determine the probability of a break

“this” could refer to: “a probabilistic model”, “the POS tag”.

The second method uses a probabilistic model to determine the probability of a break given the POS tags assigned to the preceding and following words.

This is taken care of with POS tagging, which assigns a part of speech to each token. A Hidden Markov Model is implemented for POS tagging. **This** allows us to disambiguate homographs since it takes into account the probability of specific tags and tag sequences.

“This” could refer to: “POS tagging”, “A Hidden Markov Model”.

Part Of Speech (POS) is needed to disambiguate homographs, and is assigned to each word token by a POS tagger. The tagger is typically based on a Hidden Markov Model, which models the probability of a tag sequence, given the words.

CHATR aimed to reduce the amount of signal processing needed so that the naturalness of output was not compromised. **This(1)** was planned such that the path with the least cost (target and concatenation) would be chosen during waveform synthesis. **This(2)** was set up to train cost functions, which then took charge of unit selection

“This(1)” could refer to: “CHATR”, “the amount of signal processing”, “naturalness of output”, “not compromised”.

“This(2)” could refer to: “CHATR”, “the path”, “waveform synthesis”.

CHATR aimed to reduce the amount of signal processing needed, so that naturalness of output was not compromised. The path with the smallest sum of target and concatenation costs is chosen during waveform synthesis. The cost functions are trained…

Diphones are small waveforms that represent the signal in transitions between phones corresponding to different phonemes, providing for a smoother transition between phones. **This** is because the middles of syllables tend to be less influenced by their contexts than their outer parts

“This” could refer to: diphones representing "the signal in transitions between phones”, “providing for a smoother transition”.

Diphones are fragments of waveform, capturing the transitions between phones. Compared to using phone waveform units, they provide a smoother transition between phones because the middles of phones tend to be less influenced by context than their edges.

Festival in particular splits the phonemes into diphones - concatenating the latter half of the first phone with the first half of the second phone. **This** is because there is less acoustic erraticness in the middle of a phone when compared to the end/start of a phone, allowing a smoother transition between phones.

“This” could refer to: “Festival…splits”, the abstract notion of a diphone, how diphones are constructed.

Festival uses diphone waveform units. A diphone contains the second half of a phone followed by the first half of the subsequent phone. There is less acoustic variability in the middle of a phone than at its boundaries with adjacent phones. A sequence of diphones is thus joined at points of minimum acoustic variability, resulting in smoother transitions.

Festival uses a linear predictive sequence analysis method to join waveforms smoothly. **This** involves identifying filter coefficients, and separating a sound at the source/filter level in order to be able to independently manipulate both F0 value and the spectral shape, interpolating using linear prediction over spectral inconsistencies and thus smoothing.

“This” could refer to: “linear predictive analysis”, “join waveforms smoothly”.

Festival uses linear prediction to join waveforms. Linear prediction is a source filter model, so requires the filter coefficients and source signal (called the *residual*) to be extracted and stored for every frame of every unit in the database. It can independently manipulate F0 and spectral shape by manipulating the source and filter independently. To join waveforms smoothly, the filter coefficients are interpolated across the join position. Optionally, F0 can also be smoothed by performing Pitch-Synchronous Overlap-and-Add (PSOLA) on the residual.

Text Normalisation takes substrings that are in a non-standard written form and converts them into a canonical word. **This** may also vary based on context.

“This” could refer to: “Text Normalisation”, “non-standard written form”, “canonical word”.

Text Normalisation converts all Non-Standard Words (NSWs) into words. The conversion takes into account the context in which each NSW occurs.

Since written convention dictates that words are separated with whitespace, it is useful to split text by whitespace before categorising tokens into their respective sentences. Text accomplishes **this** using simple rules.

"this" could refer to: "split text by whitespace", "categorising tokens".

Since written convention dictates that words are separated with whitespace, it is useful to split text by whitespace before categorising tokens into their respective sentences. `Text` splits text using simple rules.

Next, for each of the identified words, the part of speech must be determined in order to look up the correct lexical item in the dictionary. **This** is achieved by a part of speech tagger…

"This" could refer to: "the part of speech must be determined", "look up…in the dictionary".

In order to look up the correct dictionary entry for a word, its part of speech (POS) is required. POS is determined by a part of speech tagger…

Once non-standard words have been expanded, the processing of Part-of-Speech (POS) tagging is implemented. **This** is defined as "the process of assigning a part-of-speech or other syntactic class marker to each word in a corpus" (*citation*).

"This" could refer to: POS tagging, the implementation of POS tagging.

Once non-standard words have been expanded, the Part-of-Speech (POS) of each word is assigned.

The unit selection synthesis is only slightly different from diphone synthesis, firstly in that it has many copies of each diphone, and secondly, in that signal processing is minimally applied. **This** avoids artefacts - properties of signal which is not present in the original speech.

"This" could refer to: "unit selection synthesis", "many copies of each diphone", "signal processing is minimally applied".

Unit selection synthesis differs from diphone synthesis in that many copies of each diphone are available in the database. This means that less signal processing is required, which in turn reduces artefacts (properties of the signal not present in the original recorded speech).

As expected, Festival did not assign any breaks until the final period. **This** obviously affects the naturalness of the speech…

"This" could refer to: not assigning any breaks, the break at the final period.

As expected, Festival did not assign any breaks until the final period. The lack of within-sentence phrase breaks reduced the naturalness of the speech…

Festival works through a structured scheme. When a user inputs text, **this(1)** is stored as an object with variables inside it. Each stage of the pipeline will add info upon **this(2)** structure.

"this(1)" could refer to: "text", "a structured scheme".

"this(2)" could refer to: "an object".

Festival creates a data structure called an *Utterance* for each input text. Linguistic information is stored within an Utterance object in *Relation* data structures. Each stage of Festival's pipeline adds Relations, or modifies existing ones. For example, the input text is stored in a list Relation called…

In this version of Festival, phrase boundaries are predicted using the Classification And Regression Tree (CART) method. CART tree is a statistical method that learns automatically from labelled data the value of variables. **This** is done in a training phase where the system learns which features of the linguistic context (predictors) influence the choice of a specific value (predictees).

"This" could refer to: phrase boundary prediction, "the CART method", "learns automatically".

In this version of Festival, phrase boundaries are predicted by a classification tree. The tree is learned in advance from data labelled with the *predictee* (here, presence or absence of a phrase boundary at every word juncture) and *predictors* that must be known for any sentence to be synthesised, such as the Part Of Speech (POS) of the words surrounding every potential phrase break position.

In the absence of better options, it therefore selects diphones which don't match in f0. **This** could be resolved by performing pitch smoothing via TD-PSOLA.

"This" could refer to: the selection of diphones which don't match in f0, the mismatch in f0.

In the absence of better options, it therefore selects diphones which don't match in f0. The resulting f0 discontinuity could be reduced by performing pitch smoothing via TD-PSOLA.

The addition of silence also important: pauses of different lengths are used to signify phrase breaks of different degrees - in English text, **this** is predictable from punctuation.

"this" could refer to: "pauses of different lengths", "phrase breaks of different degrees".

The addition of silence is also important: pauses of different lengths are used to signify phrase breaks of different degrees. In English text, phrase break locations and strengths are predictable from punctuation.

In order to be able to more accurately label it [*the POS of a particular example word*] in any context, the model would have to be trained on a wider range of sentences. **This** is less likely to be an issue in commercial TTS systems because they will use more data…

"This" could refer to: incorrect POS of the particular example word, the training of the model, the requirement for "a wider range of sentences".

In order to be able to more accurately label it in any context, the model would have to be trained on a wider range of sentences. POS tagging errors are expected occur less frequently in commercial TTS systems, which use taggers trained on more data…

For most normal sentences, modern POS taggers, including the one used in Festival, don't make many mistakes in tagging sentences. Proverbs are a good example of sentences that might be problematic as they don't always follow well-defined structures. In the proverb "Live and let live." **this** results in an audible problem…

"this" could refer to: the fact that modern POS taggers "don't make many mistakes", the implication that POS taggers will make more mistakes on proverbs, the fact that proverbs "don't always follow well-defined structures".

For most normal sentences, modern POS taggers, including the one used in Festival, don't make many mistakes. However, sentences with atypical grammatical structures, such as proverbs, are still problematic. In the proverb "Live and let live.", a POS tagging error results in…

Intonation Prediction

This module is performed when a diphone voice is used, more sophisticated voices will perform this differently. **This** is comprised of three steps, prediction of placement, type, and F0 realization targets.

"This" could refer to: "intonation prediction", "module".

Intonation prediction is necessary for diphone speech synthesis and comprises three steps of prediction: accent placement, accent type, and F0 realisation. Unit selection speech synthesis may do only the first one or two steps, all three, or - as in the case of Festival - none at all.

The prediction of phrase-breaks is also obtained through training data in which each word has a prediction of whether there is a phrase break after it. **This** therefore does not concentrate as much on constituent phrases, but the probability of breaks between words using the POS of the words surrounding the word being analysed.

"This" could refer to: "prediction of phrase-breaks", "training data", prediction of phrase breaks.

The prediction of phrase-breaks is done using a model trained on data in which every word is annotated with whether there is a phrase break after it. The model is not one of constituent phrases, but simply of the probability of a break at each word juncture, given the POS of the words in a fixed window centred on the juncture.

The errors could be addressed by improving the models that underlie each task. **This** also illustrates the strength of the pipeline architecture, since each module can be improved independently.

"This" could refer to: "errors", the addressing of those errors, "improving the models".

The errors could be addressed by improving the models used in the module responsible for each error. A strength of the pipeline architecture is that each module can be independently improved.

Another problem is that diphone synthesis captures only the articulation due to a single neighbouring phone. For **this** reason, the word unit, which contains required speech specification, would be better than diphone…

"this" could refer to: "problem", "articulation".

Another problem is that diphone synthesis captures only the articulation due to a single neighbouring phone. Word units would be better choice than diphones because they capture longer-term co-articulation.

Both "Dr." and "Colin" are interpreted by the POS tagger as proper nouns, however, the full-stop should also be interpreted as a POS. **This** is most likely due to a prediction sequence error

"This" could refer to: interpreting the full-stop as a POS, the error in tagging "Dr." and "Colin".

Both "Dr." and "Colin" are interpreted by the POS tagger as proper nouns, however, the full-stop should also be interpreted as a POS. The tagging error is most likely due to…

Storing expressions with clitics as wholes thus does not cover unusual or multiple clitics and creates mistakes. **This** is solved by tokenizing these expressions as one and looking up their parts in the morpheme dictionary .

"This" could refer to: "mistakes", not covering "unusual or multiple clitics".

Storing whole clitics will not cover unusual or multiple clitics (e.g., "shouldn't've"). A possible solution would be to tokenise clitics, then look up their parts in a morpheme dictionary (e.g., "should" + "n't "'ve").

Diphones are a better choice than regular phones, since they extend from within one phone to the middle of the next. **This** is because the sound is more stable around the middle,…

"This" could refer to: diphones being a "better choice", the fact that they "extend from within one phone to the middle of the next".

Diphones extend from within one phone to the middle of the next. This is a better choice than whole phones because speech is more acoustically stable around the middles of phones than at phone boundaries.

An LTS model employs rules in order to map from a sequence of letters to the most probable sequence of phonemes. In festival **this** is done automatically, using a CART tree.

"this" could refer to: "employs rules", "map from a sequence of letters…phonemes".

A LTS model uses rules to map from a sequence of letters to the most probable sequence of phonemes. Festival uses a CART tree, which is effectively a set of ordered rules learned from data.

As opposed to syllables, these units [*diphones*] of speech will make the transition between utterances smoother and a finer grained context dependency. However, **this** also means that there will be a higher number of diphones required…

"this" could refer to: the choice of diphones, smoother transitions, "finer grained context dependency".

Compared to syllables, diphones generally have smoother transitions between units. However, a higher number of diphones will be required…

POS tagging is crucial further along in the pipeline for the pronunciation and prosody predictions. **This** is especially useful when there are homograph disambiguities in the input text.

"This" could refer to: "POS tagging", "pronunciation and prosody predictions".

POS tagging is especially useful when there are ambiguous homographs in the input text. It is also crucial further along in the pipeline for the pronunciation and prosody predictions.

Another prosodic feature predicted using CART is accent. **This** has a few predictees…

"This" could refer to: "accent", the CART used to predict accent.

Another prosodic feature predicted using a CART is accent. Only a few predictees are used to make this prediction…

…but "Prof" receives no such marker, and the period is left as is. **This** is because it isn't stored in the list of regular expressions used to detect such abbreviations.

"This" could refer to: "Prof", "the period".

…but "Prof" is not in the list of regular expressions used to detect such abbreviations, and as a consequence receives no such marker, and the period is left as is.

Different from literature and poetry, prosody usually refers to the study in intonation and rhythm. In speech processing, we usually relate **this** word with the acoustic feature like duration or F0.

"this" could refer to: "prosody", "intonation", "rhythm".

In contrast to literature and poetry, in spoken language *prosody* usually refers to intonation and rhythm and relates to acoustic features including duration or F0.

Phrase breaks are somewhat subjective, and it is hard to tell without punctuations; and **this** may be the reason why Festival only gives a phrase break to where punctuation appears.

"this" could refer to: "Phrase breaks are somewhat subjective", "hard to tell without punctuations".

Phrase breaks are somewhat subjective. Predicting where to place them is hard, especially without punctuation. These are two reasons why Festival only places phrase breaks where punctuation appears.

While Festival correctly expand “$5” into “five dollars”, it does not recognize other units of measurements like “Hz” for Hertz. Consequently, “50Hz” is expanded into “five zero H Z”. As Festival uses hand-craft rules to deal with numbers, **this** indicates that Festival does not have the relevant rules.

“this” could refer to: “50Hz is expanded into five zero H Z”, “does not recognize other units of measurements”, “Festival uses hand-craft rules”.

While Festival correctly expands “$5” into “five dollars”, it does not recognise other units of measurements like “Hz” for Hertz. Consequently, “50Hz” is expanded into “five zero H Z”. Festival uses hand-crafted rules to deal with numbers, and this error indicates that Festival does not have the relevant rules.

Another missing diphone is s_r found in words like “x-ray”, “disreputable”, and “disregard”. **This(1)** backs off three times in each case using the #_# diphone which leaves artefacts in the waveform. **This(2)** is likely due to a small speaker database.

“This(1)” could refer to: Festival?

“This(2)” could refer to: “backs off three times”, “artefacts”, “missing diphone”.

Another missing diphone is /s_r/ found in the words “x-ray”, “disreputable”, and “disregard”. Festival backs off in each case, finally using the silence diphone /#_#/ which produces artefacts in the waveform. Missing diphones are a result of a small database of diphones.

The pronunciation is correctly identified as [d-oo-l-f-i-n] but the waveform sounds like [dolofin]. **This** simply means a wrong entry in the diphone database.

“This” could refer to: “pronunciation is correctly identified”, “waveform sounds like [dolofin]”.

The pronunciation is correctly identified as [d-oo-l-f-i-n] but the waveform sounds like [dolofin], because the diphone selected from the database for [l-f] actually sounds like [l-o-f].

Tokenisation involves breaking user-inputted text into ‘tokens’ by whitespace. **This** includes punctuation…

“This” could refer to: “Tokenisation”, “user-inputted text”, “whitespace”.

Tokenisation involves breaking user-inputted text into ‘tokens’ according to whitespace and punctuation…

Let’s first consider Festival’s output from syntax level. It tagged ‘his’ as a possessive pronoun, however, there was no context to show it is a pronoun. **This** suggests one potential way to solve this error is…

“This” could refer to: tagging ‘his’ as a possessive pronoun, “no context”.

First, consider Festival’s output at the level of syntax. It tagged ‘his’ as a possessive pronoun, however, the context in this example does not indicate that it is a pronoun. Therefore, one potential way to solve this error is…

These errors are audible in the synthesized waveform, but may occur at any stage of TTS. **This** is due to the sequential nature of the pipeline.

“This” could refer to: “errors are audible”, “may occur at any stage”.

The sequential nature of the pipeline means that audible errors in the synthesised waveform could have occurred at any stage of TTS.

Now that the Front End information is gathered, waveform synthesis can begin. In Festival this is done using a unit selection synthesis and uses diaphones. **This** is because the process of coarticulation occurs in natural language everywhere…

“This” could refer to: the choice of unit selection, the choice of diphone units.

After the linguistic specification has been completed, the waveform is synthesised. In Festival this is done using unit selection with diphone units. Diphones are the preferred unit type because they capture the process of coarticulation…

When the system checks the wrong entry in the dictionary, the pronunciation is wrong. This is mainly cause by the HMM tagger.

“This” could refer to: “the wrong entry”, “the pronunciation is wrong”.

Incorrect POS tags can cause the wrong entry to be retrieved from the dictionary, leading to an incorrect pronunciation.

In Festival, it is possible to further label NSW tokens to identify their type, and therefore remove the ambiguity. **This(1)** is done in the [`Token_POS`] module. In order to do **this(2)**, NSW’s must first be identified…

“This(1)” could refer to: “label NSW tokens”.

“this(2)” could refer to: “label NSW tokens”, execute the `Token_POS` *module.*

In Festival, the *`Token_POS`* module labels NSW tokens to identify their type, and therefore resolve ambiguity. Before identifying their type, NSWs must first be identified…

Festival fails to detect the period in “W. Schiller” as a symbol of abbreviation so that the “w” and the ”.” are normalised into two tokens. The reason for **this** is…

“this” could refer to: failure to detect the abbreviation, normalisation into two tokens.

Festival fails to detect the period in “W. Schiller” as a symbol of abbreviation, and thus the “w” and ”.” are split into two tokens. The abbreviation is not detected because…

I printed the relation “Token” after doing “Token_Pos”, a step which detects and assigns specific labels to NSW. However, no label is assigned to the temperature expression. **This** can further be proved by the relation

“This” could refer to: the fact that Token_Pos detects NSWs, “no label is assigned”.

“Token_Pos” detects NSWs and labels them with categories. However, no label is assigned to the temperature expression, as can be seen in the relation…
