PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Prosody modelling in concept-to-speech generation: methodological issues

Kathleen R. McKeown and Shimei Pan

Phil. Trans. R. Soc. Lond. A 2000 **358**, 1419-1431
doi: 10.1098/rsta.2000.0595

THE ROYAL SOCIETY

Prosody modelling in concept-to-speech generation: methodological issues

By Kathleen R. McKeown and Shimei Pan

Department of Computer Science, Columbia University, New York, NY 10027, USA (kathy@cs.columbia.edu; pan@cs.columbia.edu)

We explore three issues for the development of concept-to-speech (CTS) systems. We identify information available in a language-generation system that has the potential to impact prosody; investigate the role played by different corpora in CTS prosody modelling; and explore different methodologies for learning how linguistic features impact prosody. Our major focus is on the comparison of two machine learning methodologies: generalized rule induction and memory-based learning. We describe this work in the context of multimedia abstract generation of intensive care (MAGIC) data, a system that produces multimedia briefings of the status of patients who have just undergone a bypass operation.

Keywords: concept-to-speech generation; speech synthesis; natural language generation; machine learning

1. Introduction

In many applications where speech is the appropriate medium for human–computer interaction, not only must sound be automatically produced, but the content and wording of what is to be said must be computed as well. This is the case, for example, in spoken-dialogue systems, where, in reply to a question, a system must be able to formulate an answer using results from a database search. It is also the case in systems where eyes-free interaction is important, such as when the user is flying a plane, driving a car, or in a medically demanding situation.

When the system is not merely reading fully formed text, as is the case for text-to-speech (TTS), but is automatically producing content, wording and sound, we should be able to do better than directly using TTS for speech synthesis. The production of natural, intelligible speech depends, in part, on the production of proper *prosody*: variations in pitch, tempo and rhythm. Prosody modelling depends on associating variations of prosodic features with changes in structure, meaning, intent and context of the language spoken. Such information is readily available when language is produced from concepts. Using TTS, however, would require re-deriving such information from text, an inaccurate process in some cases and not yet possible in others.

Developing a concept-to-speech (CTS) system and then testing whether it performs better than TTS in the same context raises a number of difficult methodological issues. Prosody modelling for CTS requires developing rules that use information produced during the generation of language to set prosodic variables. This process involves selecting information that has potential to influence prosody, identifying correlations between this information and prosodic parameters through data exploration, and using learning algorithms to build prosody models from these data. Each stage encompasses variables in how it is carried out that affect both the results and the potential for comparison with TTS.

In this paper, we identify several such variables. We itemize the information that language generation produces. How it differs from the information used in TTS affects the possibility for increase in performance in CTS. We then explore the kind of data that can be used to study correlations between information and prosody, discussing difficulties in obtaining such data. Prosody modelling typically requires annotation of speech corpora and such annotation can be tedious and time-consuming when it has to be done manually. Finally, we look at how different learning mechanisms impact results when applied to different data, contrasting the use of two empirical methods for prosody modelling, whether part of CTS or TTS. We present work that uses automatic rule induction to generalize across multiple speakers and phrases and learn correlations between linguistic and prosodic features. In the second approach, we use memory-based learning to identify close matches between the current input and phrases within a speech corpus previously annotated with prosody. We then borrow the prosody used in that phrase for the current input. Since the first approach generalizes across multiple cases, it captures commonalities, but it loses specificity in representing influences on prosody. In memory-based modelling, the particular correlation that we learn may occur only once in the data.

In the following sections, we illustrate these issues in the context of CTS research that we are carrying out in multimedia abstract generation of intensive care (MAGIC) data, a system that generates multimedia briefings of a patient's status after having a bypass operation (Dalal *et al*. 1996; McKeown *et al*. 1997). We first describe information that MAGIC generates in the process of producing language, turning next to the corpora we collected. We then provide a description of the more traditional approach to prosody modelling, using machine learning that generalizes over many examples, followed by a description of our memory-based approach. Our results show that the memory-based approach yields a better improvement in quality, measured through subjective judgments of output.

2. Information from language generation

In the course of producing language, language generators typically produce a variety of intermediate linguistic representations that contain information that could potentially influence prosody. Some of this information is similar to the kind of information used in TTS, such as part-of-speech (POS) tags or syntactic constituency structure. In these cases, CTS input is more accurate since it was constructed during sentence generation, while TTS input must be approximated from parsing or POS tagging. As a result, we would expect a gain in CTS performance (Pan & McKeown 1998). Other information produced in language generation is semantic or pragmatic in nature and is often not available for TTS prosody modelling. CTS prosody modelling might gain the biggest improvements over TTS by modelling this type of information, but it can be difficult to annotate in speech corpora and, thus, use in training. Sometimes, even in CTS, this information is approximated in training data to make learning practical, and this confounds comparison with TTS. In this section, we describe the information produced by MAGIC's language-generation component, which is similar to that produced by most language generators.

MAGIC is a multimedia briefing system that produces a patient's post-operative status report from a medical database. MAGIC exploits the extensive online data available from the Columbia Presbyterian Medical Centre (CPMC) as its source of content for its briefing, which includes patient demographics, medical history, vital signs, drugs, and other manually entered operative events. MAGIC's language generator is composed of several stages, which, except for the first, are typical in language generation, including *database access and medical inference*, *content planning*, *sentence planning*, and *surface realization*.

In *database selection and medical inference*, medical inference is performed to identify abnormal events from numeric data in the patient record (e.g. that the patient has hypertension). In database selection, relevant attribute-value pairs, such as the patient's name and gender, are selected and placed in a domain ontology, along with inference results. Thus, the features produced at this stage are concepts, as well as their associated semantic classes. Such features may facilitate prosody modelling. For example, semantic concepts make it easier to specify whether a discourse entity is given or new, while semantic abnormality deduced by the inference module may be highlighted using, for example, pitch changes.

The *content planner* uses a presentation strategy to determine and order content. It represents discourse structure, which is a hierarchical topic structure in MAGIC, discourse relations, which can be rhetorical relations, and discourse status, which represents whether a discourse entity is given, new or inferable and whether the entity is in contrast with another discourse entity. Most of the features produced at this stage have been shown to have influence on prosody: discourse structure can affect pitch range, pause and speaking rate (Grosz & Hirschberg 1992); given/new/inferrable can affect pitch-accent placement (Hirschberg 1993); a shift in discourse focus can affect pitch-accent assignment (Nakatani 1998); and contrastive entities can bear a special pitch accent (Prevost 1995).

The *sentence planner* constructs a lexicalized semantic structure to express the selected content, which includes semantic roles and semantic constituent structure. For example, a sentence consists of a *process* that represents the verb, several participants (obligatory arguments), and one or more circumstances (optional arguments to the verb). Each constituent can have different modifiers, such as classifiers, describers and qualifiers. In earlier work, we experimented with the effect that semantic boundary has on various prosodic features (Pan & McKeown 1998).

The *surface realizer* uses an English grammar, transforming a lexicalized semantic structure into a syntactic structure, linearizing the structure, and handling morphology and function-word generation. The features available after surface realization include syntactic constituent structure, syntactic function (subject, object, complements, etc.) and POS. Other information—such as the lexical item, word position and distance—can be easily computed from a string of words. Surface information comprises the most widely used features in existing prosody modelling systems. For example, POS is used in almost all existing speech-synthesis systems, syntactic structure has been used for prosodic phrasing (Bachenko & Fitzpatrick 1990), and word, position and distance information are used in pitch-accent and prosodic phrase-boundary prediction (Wang & Hirschberg 1992).

3. Speech corpora

Speech corpora typically provide the data from which we can draw inferences about the correlation between linguistic features and prosody. We face trade-offs in building a collection including ease of annotation, similarity to target output, and naturalness. In CTS development, there are many features that can be explored, many questions concerning how to represent them, and very difficult practical problems given the time it takes to manually annotate speech. We were particularly concerned with making annotation practical by reducing manual effort, facilitating annotation of the widest range of features, and with the quality and relevance of the corpus.

We collected and used three different types of corpora—spontaneous speech, read speech, and written text—all in the same domain. The written text gave us a large amount of data (1.24 million words in 2422 discharge summaries) from which to carry out statistical modelling based on word counts. The spontaneous-speech corpus was collected at CPMC, where doctors informed residents and nurses about the post-operative status of a patient who had just undergone bypass surgery. These briefings are the targets for MAGIC output, and, because they were recorded as clinicians went about their normal routine, the prosody as well as the content and wording are natural, reflecting MAGIC's real-world counterpart. The read-speech corpus includes recordings of one doctor reading five system-generated reports. In this case, the speech exactly mirrors output we are trying to produce. Both the spontaneous-speech and read-speech corpora were much smaller than the written collection.

Both speech corpora were intonationally labelled with pitch accents by an expert in tone and break index (ToBI; see Silverman *et al.* (1992)). The spontaneous-speech corpus was automatically annotated with POS information, syntactic constituent boundaries, syntactic functions, and lexical repetitions, using approximations provided by POS taggers and parsers. It was also manually labelled with given/new/inferable information. We are still working on manually labelling discourse structure, discourse relations, and semantic abnormality. Since the read speech is actually system output, each word was associated with a set of features that are automatically extracted from the syntactic and semantic representations produced by the text generator, avoiding manual effort and resulting in accurate annotation. We are working on automatically augmenting this corpus with new discourse and semantic features.

While the spontaneous-speech corpus is a larger collection and was recorded in a natural setting, it has differences from the output we want to produce. In spontaneous speech there are disfluencies including insertions, such as 'uh', 'um', 'you know', repairs and ungrammatical sentences, all of which can be omitted from MAGIC output. Furthermore, given differences in wording from system output, the spontaneous speech required manual annotations or the same approximations in labelling the data as is used for TTS (parsing, POS). This will yield errors in the training data and could potentially affect the rules learned, which will ultimately be applied in CTS where approximations are not used.

The read-speech corpus gives us speech that is as close as possible to the output that we want to generate. Since the language was actually produced by MAGIC, only the prosody was manually labelled. This increases the number of features that we can realistically model, providing a practical means of collecting data for learning. A drawback of this corpus is that it is not in a natural setting, and therefore the prosody is not always totally natural. For example, the speaking rate is lower than the normal speed. These two corpora involve different trade-offs; ultimately, we may do better by integrating the best features of each.

4. Generalized rule induction

Generalized rule induction allows us to test and identify the influence of specific features on prosody. It learns rules that quantify the correlation between one or more linguistic and prosodic features, where the rules generalize across many examples. Because we have consistently seen examples of the influence multiple times, reliability is higher. Because the rule generalizes over many examples, some of the variations may be lost when instances are grouped together. Since the resulting rules are understandable, researchers can inspect the results to either confirm or contradict linguistic judgments. Thus, the results not only provide a computational model that can be used to improve speech quality in actual systems, they also provide insight into our understanding of how prosody is determined.

In this paper, we use our work on the influence of word informativeness to illustrate this approach. This new feature could apply equally well to both TTS and CTS approaches, but, as an example, it shows the positive features of generalized rule induction. In previous work (Pan & McKeown 1998), we experimented with syntactic and semantic features available in CTS.

(*a*) *Word informativeness and pitch-accent prediction*

One critical issue in prosody modelling is pitch-accent assignment. Pitch accent is associated with the pitch prominence of a word. Some words may sound more prominent than others within a sentence because they are associated with a significant pitch rise or fall. Usually, the prominent words bear pitch accents, while the less prominent ones do not. Although native speakers of a language have no difficulty in deciding which words in their utterances should be accented, the general pattern of accenting in a language, such as English, is still an open question.

Some linguists speculate that relative informativeness, or semantic weight of a word, can influence accent placement (Ladd 1996; Bolinger 1972), with words that carry more semantic information being more likely to bear a pitch accent. In our medical domain, semantic informativeness should be influenced by the results of our inference component. Our preliminary results show that abnormal results are likely to be communicated by more informative words. As a first step towards capturing semantic informativeness, though, we use the information content (IC) of the word following information theory. IC is relatively easy to compute and we can determine interactions between IC and pitch accent quickly.

(*b*) *Experiments using information content*

Following the standard definition in information theory, the IC of a word can be defined as

$$\mathrm{IC}(w) = -\log(P(w)),$$

where $P(w)$ is the probability of the word w and it is computed using the maximum-likelihood estimation $F(w)/N$, where $F(w)$ is the frequency of w in the corpus and N

Table 1. *Different pitch-accent models*

models	RIPPER performance
baseline	52.02%
IC model	70.06%
POS model	70.52%
POS+IC model	73.71%

is the accumulative occurrence of all the words in the corpus. Intuitively, if the probability of a word increases, its informativeness decreases and, therefore, it is less likely to be an information focus. Similarly, it is therefore less likely to be communicated with pitch prominence.

We use the text corpus to calculate IC, preprocessed to remove endings. In general, most of the least-informative words are function words, such as '*with*' or '*on*'. However, some content words are selected, such as '*patient*' and '*day*'. These content words are very common in this domain and are mentioned in many documents in the corpus. In contrast, the majority of the most informative words are content words, such as '*zphrin*', '*xyphoid*' or '*pyonephritis*'.

In order to verify whether word informativeness is correlated with pitch accent, we employ Spearman's rank-correlation coefficient, ρ, and associated test to estimate the correlations between IC and pitch prominence. IC is closely correlated to pitch accent with a significance level $p = 2.90 \times 10^{-84}$. The positive correlation coefficient ρ (0.34) indicates that the higher the IC, the more likely a word is to be accented.

We also want to show how much performance gain can be achieved by adding this information to pitch-accent models. We used RIPPER (Cohen 1995), a system that learns sets of classification rules from training data, to learn models that predict the effect of informativeness on pitch accent. We trained RIPPER on the speech corpus. Once a set of RIPPER rules are acquired, they can be used to predict which word should be accented in a new corpus. Note that RIPPER rules can be inspected and allow us to understand the exact basis for the correlation between predictor and response variable, seeing whether they confirm linguistic intuition.

(*c*) *Results*

We use a baseline model where all words are assigned a default accent status (accented), which has a performance of 52%. Our results show that when IC is used to predict pitch accent, performance increases to 70.06%, statistically significant with $p < 1.11 \times 10^{-16}$,† using the χ^2 test.

In order to show that IC provides additional power in predicting pitch accent than current models, we also ran experiments that compare IC alone against a POS model for pitch-accent prediction, the most powerful predictor in most TTS systems. In order to create a POS model, we first use MXPOST, a maximum entropy part-of-speech tagger (Ratnaparkhi 1996), mapping all the POS tags into seven categories: 'noun', 'verb', 'adjective', 'adverb', 'number', 'pronoun' and 'others'. Keeping all initial tags (about 45) would drastically increase the requirements for the amount of

† S-plus reports $p = 0$ because of underflow. The real p value is less than 1.11×10^{-16}, which is the smallest value the computer can represent in this case.

training data. As shown in table 1, the performance of the POS model is 70.52%, which is comparable with that of the IC model. When the POS models are augmented with IC, the POS+IC model performance is increased to 73.71%, a statistically significant improvement with $p = 0.005$. These experiments produce new evidence confirming that IC is a valuable feature in pitch-accent modelling.

5. Memory-based prosody modelling

In contrast to generalized rule induction, prosody prediction in memory-based prosody modelling is based on similar pre-stored instances in the speech corpus instead of rules that generalize across instances. Given a sentence for synthesizing, the system will find the best match from the prosodically tagged corpus, and the prosodic features of the given sentence are assigned based on the matching sentence or sentence segments. A similar approach has been used in unit selection for concatenative synthesizers (Yi 1998; Conkie 1999); in our work, only the prosody is selected and reused. We use the word 'inventory' to refer to the speech corpus used specifically for memory-based modelling.

Memory-based prosody modelling has many advantages. First, it captures the co-occurrence of the prosodic features of many words at a time. It uses many linguistic features to match against the inventory, and all prosodic features associated with the instance are selected. Thus, in this approach, many features, both input and output, are modelled simultaneously. Moreover, existing prosody modelling uses a fixed window to model context. It is hard to capture long-distance dependencies with a fixed window unless the window size is very large. In our memory-based approach, the number of words that can be modelled at a time can vary significantly. Another advantage of memory-based prosody modelling is its ability to keep specificity. Generally, a sentence can be verbalized in several equally appropriate ways. Speech with variation sounds more vivid and less repetitive.

Despite the advantages, a memory-based approach works well only when new sentences are relatively similar to the sentences stored in the inventory. If the system cannot find good matches from the inventory most of the time, the strength of this approach diminishes. Thus, it will not work well if the input is unrestricted text unless there is a huge pre-analysed speech inventory available. For most CTS systems, however, this approach can work quite well. Most generation systems are designed for a specific application and the language generator usually produces sentences with a limited vocabulary. Even with a reasonably small inventory, the system can have many good matches, which makes the memory-based approach effective and attractive. For example, MAGIC employs a flexible, advanced sentence generator that produces different sentence structures using opportunistic clause aggregation (Shaw 1998). Even in MAGIC, given two randomly selected system-generated patient reports, *ca.* 20% of the sentences and 80% of the vocabulary overlap.

In this section, we describe the signature feature vector, which represents the linguistic features used for matching, followed by the matching algorithm and results.

(*a*) *Signature feature vector*

We automatically extract a set of signature features to describe various aspects of a word from the output of a text generator. A feature is selected based on whether it is

```
4-3-1,he,pronoun,subject,c-patient,aparb,1,3.630408,h*,1,npa,nbt.
4-3-2,is,verb,predicate,c-has-attribute,bparb,8,3.8158112,na,4,h-,l.
4-3-3,fifty,cardinal,subj-comp_head,c-measurement,wb,1,5.571203,l+h*,1,npa,nbt.
4-3-4,eight,cardinal,subj-comp_head,c-measurement,wb,1,5.645311,h*,1,npa,nbt.
4-3-5,kilograms,noun,subj-comp_head,c-measurement,alib,3,7.2939696,h*,4,h-,l.
```

Figure 1. The feature vector in the speech inventory.

available in the text generator and whether it is related to different prosodic features. We use the read corpus as the inventory, where physicians read output produced by MAGIC. Currently, eight signature features are automatically extracted from MAGIC's output as shown below. Of these, Concept, SemBoundary and SemLength are available only in CTS systems. Other features, such as SynFunc, are used both in CTS and TTS, but we can expect quite a few errors in this feature in TTS where it is derived by parsing.

(1) **ID:** the ID of a feature vector. It is encoded as *dd–ss–ww*, where *dd* is the document ID, *ss* is the sentence ID, and *ww* is the position of the word in a sentence.

(2) **Lex:** the word itself, such as 'the', 'patient' or 'is'.

(3) **Concept:** the semantic category of a content word. For example, the **concept** for 'packed red blood cells' is 'blood product'.

(4) **SynFunc:** the syntactic function of a word (e.g. 'subject', 'object', 'subject-complement').

(5) **SemBoundary:** the type of semantic constituent boundary after a word (e.g. a participant boundary, a circumstance boundary; see Pan & McKeown (1998) for details).

(6) **SemLength:** the length, in number of words, of the semantic constituent associated with the current **SemBoundary**.

(7) **POS:** the part-of-speech of a word.

(8) **IC:** the semantic informativeness of a word.

In this experiment, we model all four major ToBI prosody features in English: pitch accent, break index, phrase accent, and boundary tone. Each feature can take any of the original values proposed in ToBI, thus yielding a fine-grained model of prosody variation. There are six pitch-accent classes, five break-index classes, three phrase-accent classes, and three boundary-tone classes.

Figure 1 shows the feature vectors associated with the words in the sentence 'He is fifty-eight kilograms'. The first eight features are the signature features and the last four are prosodic features.

(*b*) *Matching algorithm*

Based on the eight signature features, we define two cost functions for matching: target cost (TC) and concatenation cost (CC). Target cost measures similarity between two words based on their signature features. The lower the target cost, the

Table 2. *Memory-based prosody modelling performance*

approach	pitch accent	break index	phrase accent	boundary tone
baseline1	38.77%	54.59%	62.24%	69.39%
new test case	60.41%	78.69%	81.73%	83.76%
perfect match	66.67%	81.77%	84.91%	82.39%

more similar the words. Concatenation cost measures the smoothness of the transition from one word to another. We use the first signature feature, ID, to compute concatenation cost: 0 if two words are adjacent in the original sentence; 1 if they are not. TC is the weighted sum of the distance of the other seven features:

$$\mathrm{TC}(W_j, W_k) = \sum_i \mathrm{weight}_i \times \mathrm{Dis}(F_{j,i}, F_{k,i}),$$

where W_j and W_k are word j and k; weight_i is the weight for feature i. It measures the relative importance of each feature in the vector. It is estimated automatically using linear regression. $\mathrm{Dis}(F_{j,i}, F_{k,i})$ is the distance between the ith feature of word j and k. For a categorical feature, it is 0 if the two features are the same, otherwise it is 1. For numerical features, it is the absolute difference between them.

We employ the Viterbi algorithm (Forney 1973) to find a match from the inventory for a given sentence. It produces a matching sentence by piecing together matching words from the inventory. The Viterbi process constructs an optimal word sequence with the minimum sum of the combined cost (SoCC):

$$\mathrm{SoCC} = W_{\mathrm{t}} \times \mathrm{TC} + W_{\mathrm{c}} \times \mathrm{CC},$$

where W_{t} and W_{c} are the weights for target cost and concatenation cost, respectively. This results in the construction of a sentence from different words, phrases or entire sentences.

(*c*) *Results*

We evaluated the memory-based prosody modelling by randomly picking a new patient's report not in the inventory. We asked the same doctor to read it, recorded the speech, and the same ToBI expert transcribed the prosodic features. We measured how well the prosody assignment algorithm performs, using the new speech as the gold standard. Table 2 shows the results, where the baseline is computed by assigning a majority class to all the words in the test sentences. The memory-based model achieves a statistically significant improvement over the baseline models for all four prosodic features using the χ^2 test with $p < 0.001$.

The real performance should be better, however, because the current evaluation is biased. Our system is unfairly punished in cases where there is speaker variation in the prosody of identical sentences. 20.88% of the inventory sentences have an exact match elsewhere in the inventory (i.e. there were two instances of the same sentence in the inventory). In general, the prosody pattern in the matching sentence is also appropriate because it is produced by the same speaker and used in similar contexts. However, the speaker may vary prosody from time to time resulting in two identical

Table 3. *Subjective pair evaluation*

experiments	memory versus rule	memory versus TTS	rule versus TTS
average score μ	3.375	3.417	3.333
statistical significance for $\mu > 3$	0.0022	0.0026	0.0096

sentences with different prosody. The system is penalized in such cases. Table 2 also shows the performance for sentences with a perfect match in the inventory, illustrating that a significant negative effect was introduced by the current evaluation approach; we should have a near-perfect performance for these cases.

6. A direct comparison

In order to directly compare generalized rule induction with memory-based learning, we conducted another experiment, which uses rule induction over the same set of features used for memory-based learning derived from the read corpus. Thus, the only factor that differs is the form of learning. In order to avoid the bias against memory-based learning discussed above, we used a subjective evaluation in place of a quantitative one. This also allows us to make comparisons with a specific TTS model, although experimental variables across the TTS system and our CTS models are not consistent. We tested three prosody models (memory-based, rule-induction, and the Bell Labs' TTS model), using the same synthesizer (Bell Labs' TTS version nov92) (Sproat 1997) augmented with the different prosody models to synthesize speech.

We used a pairwise comparison between sentences produced by the different methods in order to capture judgments of the prosody and not the synthesizer in general. We randomly selected eight sentences from MAGIC output and, for each sentence, constructed three pairs: TTS versus memory-based output; TTS versus rule-based output; and memory-based versus rule-based output. The resulting 24 pairs were presented in random order, with order within pairs also randomly determined, to six native English speakers, yielding a total of 144 pair comparisons. Subjects were asked to rank the pairs stating whether system A was much better than system B, slightly better, the same, slightly worse, or much worse, which results in scores ranging from 5 to 1. Therefore, a score of 3 means there is no difference between systems A and B, and a score greater than 3 means system A is better than system B.

Table 3 indicates that the memory-based system performs better than both the rule-based system and TTS, while the rule-based system performs better than TTS. The Student t-test results in table 3 (labelled as 'statistical significance') showed that the difference in system rating is statistically significant with $p < 0.01$. We also conducted an analysis of variance (ANOVA) test on the experiment data, testing two additional variables: the subject and the sentence. Our ANOVA results show that 'subject' is indeed another significant factor which affects the rating (with $p < 0.005$). Based on subject feedback, it appears that some subjects prefer the memory-based output because it is more vivid and has many prosody variations. Others find the variations unnatural and, therefore, prefer the more neutral ones. Our ANOVA results do not show any significant difference between different sentences. This is expected, because the sentences for the experiment were randomly selected.

7. Discussion and current directions

We have explored two different methods for learning correlations between linguistic features and prosody. While these methods could be used for prosody modelling both for TTS and CTS, our goal is the development of CTS systems, and this affected the choice of parameters in the different experiments we carried out. Practical concerns with obtaining enough annotated data, where results could be directly used in CTS, dominated. This motivated our use of a corpus of read-system output, allowing us to automatically annotate the corpus with features extracted from generation-system output. We experimented with approximated features for annotation of the spontaneous-speech corpus given the difficulty in manual annotation, even though the resulting rules may not provide the true power possible with accurate CTS features; this was done with both syntactic features using parsing and informativeness using statistically derived values.

While our subjective evaluation found memory-based learning to be superior for CTS, each learning methodology has its strengths. Generalized rule induction provides a mean to test and model linguistic intuition, it results in a more robust model, and the resulting set of rules can be augmented by human expert knowledge where appropriate. However, generalized rule induction requires a lot of data, and when adequate data are not available the system may not be able to form any rule at all for a specific predictor value. Furthermore, rule-based learning typically uses coarse-grained classes as response variables; learning fine-grained classes requires prohibitively large amounts of data. In our informativeness experiments, for example, pitch accent had two values (accent or no accent), while in memory-based modelling, pitch accent had six classes. Memory-based learning, on the other hand, retains variation, since it uses the prosody associated with specific instances and can yield better results with a small number of data as long as the target speech of the system is similar to corpus examples. It does not require collapsing of ToBI values into general classes and it uses prosody from specific instances when no generalization would be possible. However, performance may change drastically if input data do not have a good match in the underlying corpus; it will not work well in an unrestricted domain and will need training on a new corpus if the application is changed. As a result, it is inherently more suited to CTS, which typically works in restricted domains, than TTS, which is domain independent. Furthermore, while results may yield better system performance, they do not provide linguistic insight.

Our current work investigates combining the two approaches as well as adding semantic and discourse features. By learning rules that can predict which model will yield best results in which cases, we may be able to develop a single system that incorporates the best features of each. Such an approach would also allow us to integrate the benefits of the different speech corpora. In our work to date, the majority of features that we have investigated are surface semantic and syntactic features. We are currently annotating the speech corpus with features closely related to meaning and discourse. We have found that some semantic features produced by the content planner of the language generator directly correlate with informativeness. In particular, in this domain, semantic features conveying measurements (e.g. lab results, vital signs) where the patient did not do as expected, whether better or worse, are important to communicate, and may be marked in speech. We are currently exploring correlation of unexpected or abnormal results with prosodic features.

References

Bachenko, J. & Fitzpatrick, E. 1990 A computational grammar of discourse-neutral prosodic phrasing in English. *Comp. Ling.* **16**, 155–170.

Bolinger, D. 1972 Accent is predictable (if you're a mind-reader). *Language* **48**, 633–644.

Cohen, W. 1995 Fast effective rule induction. In *Proc. 12th Int. Conf. Machine Learning, Lake Taho, CA, 1995*, pp. 115–123.

Conkie, A. 1999 A robust unit selection system for speech synthesis. In *Proc. 137th Mtg of the Acoustical Society of America*.

Dalal, M., Feiner, S., McKeown, K., Pan, S., Zhou, M., Hollerer, T., Shaw, J., Feng, Y. & Fromer, J. 1996 Negotiation for automated generation of temporal multimedia presentations. In *Proc. ACM Multimedia, 1996*, pp. 55–64.

Forney, G. D. 1973 The Viterbi algorithm. In *Proc. IEEE* **61**, 268–278.

Grosz, B. & Hirschberg, J. 1992 Some intonational characteristics of discourse structure. In *Proc. Int. Conf. Spoken Language Processing, Banff, Canada, 1986*, vol. 1, pp. 429–432.

Hirschberg, J. 1993 Pitch accent in context: predicting intonational prominence from text. *Artificial Intell.* **63**, 305–340.

Ladd, D. R. 1996 *Intonational phonology.* Cambridge University Press.

McKeown K. R., Pan, S., Shaw, J., Jordan, D. & Allen, B. 1997 Language generation for multimedia healthcare briefings. In *Proc. 5th Conf. Applied Natural Language Processing, 1997*, pp. 277–282.

Nakatani, C. 1998 Constituent-based accent prediction. In *Proc. COLING-ACL '98, Montreal, Canada*, pp. 939–945.

Pan, S. & McKeown, K. R. 1998 Learning intonation rules for concept to speech generation. In *Proc. COLING-ACL '98, Montreal, Canada*, pp. 1003–1009.

Prevost, S. 1995 A semantics of contrast and information structure for specifying intonation in spoken language generation. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA.

Ratnaparkhi, A. 1996 A maximum entropy part of speech tagger. In *Proc. Conf. Empirical Natural Language Processing, University of Pennsylvania, 1996*.

Shaw, J. 1998 Clause aggregation using linguistic knowledge. In *Proc. 9th Int. Workshop on Natural Language Generation, Niagara-on-the-lake, Canada, 1998*, pp. 138–147.

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J. & Hirschberg, J. 1992 ToBI: a standard for labeling English prosody. In *Proc. Int. Conf. Spoken Language Processing, 1992*, vol. 2, pp. 867–870.

Sproat, R. 1997 *Multilingual text-to-speech synthesis: the Bell Labs approach.* Boston, MA: Kluwer.

Wang, M. & Hirschberg, J. 1992 Automatic classification of intonational phrase boundaries. *Comp. Speech Lang.* **6**, 175–196.

Yi, J. 1998 Natural-sounding speech synthesis using variable-length units. Master's thesis, MIT, Boston, MA, USA.

Discussion

Unreported speaker. The 'read' example, even being read by a doctor, had the problem that we observed yesterday, of not being as natural as the spoken speech. Have you thought of using trained actors for reading these things?

K. R. McKeown. We know that it's not as natural as spontaneous speech, but, despite that, we get better results using it. In terms of who reads it, we find it has to be someone with a medical background, because of potential problems with conveying the semantic import.

K. Spärck Jones (*University of Cambridge, UK*). Why do you have to have this stuff as speech at all? Why don't the doctors just read the text, which would be quicker?

K. R. McKeown. I should have brought out that aspect of the system more in the talk. We spent a lot of time with different clinicians in the early stages of getting the system running, and they prefer it. They're very busy, and they need to be able to continue with their other tasks while the briefing is given.

Unreported speaker. What struck me about the difference between the spontaneous and read speech was the size of the pieces of information. In the spontaneous speech it was very short phrases, whereas in the system-generated sentence there was a long chunk of read data. Have you looked at the ability of people to assimilate data, in either small chunks, well separated in time, or very long streams? I was thinking of weather forecasts, for instance, where people find it very difficult to assimilate the data from a long stream.

K. R. McKeown. I think that was mostly a result of other differences in the data. In the lists of drug names and doses, you could generate that in small chunks. I don't know of any work on that particular question. I do know that in our particular domain the quicker you can say it the better.

K. Spärck Jones (*University of Cambridge, UK*). The very long drug names and other technical words: do you find any problems embedding these in otherwise prosodically natural speech? They seem to me to be rather indigestible lumps.

K. R. McKeown. Physicians and nurses are very familiar with these terms. It shouldn't cause problems for them. Embedding technical words won't cause problems because the prosody models were trained in this domain.