Prosody Prediction from Treelike Structure Similarities

Laurent Blin[1] and Mike Edgington[2]

[1] IRISA-ENSSAT, F-22305 Lannion, France
blin@enssat.fr
[2] Speech Technology and Research Lab, SRI International,
Menlo Park, CA-94025, USA
mde@speech.sri.com

Abstract We present ongoing work on prosody prediction for speech synthesis. This approach considers sentences as treelike structures and decides on the prosody from a corpus of such structures using machine learning techniques. The prediction is achieved from the prosody of the closest sentence of the corpus through tree similarity measurements in a nearest neighbour context. We introduce a syntactic structure and a performance structure representation, the tree similarity metrics considered, and then we discuss the prediction method. Experiments are currently under process to qualify this approach.

1 Introduction

Producing a natural prosody remains a problem in speech synthesis. Several automatic prediction methods have already been tried for this topic, including decision trees [1], neural networks [2], and HMMs [3]. We are introducing a new prediction scheme. The original aspect of our approach is to consider sentences as treelike structures and to decide on the prosody from a corpus of such structures. The prediction is achieved from the prosody of the closest sentence of the corpus through tree similarity measurements using the nearest neighbour algorithm. We think that reasoning on a whole structure rather than on local features of a sentence should better reflect the many relations influencing the prosody. This approach is an attempt to achieve such a goal.

The data used in this work is a part of the Boston University Radio (WBUR) News Corpus [4]. The prosodic information consists of ToBI labeling of accents and breaks [5]. The syntactic and part-of-speech informations were obtained from the part of the corpus processed in the Penn Treebank project [6].

We firstly describe the tree structures defined for this work, then present the tree metrics that we are using, and finally discuss how they are manipulated to achieve the prosody prediction.

2 Tree Structures

So far we have considered two types of structures in this work: a simple syntactic structure and a performance structure [7]. The comparison of their use should be helpful for providing some interesting knowledge about the usefulness or the limitations of the different elements of information included in each structure.

2.1 Syntactic Structure

The syntactic structure considered is built exclusively from the syntactic parsing of the given sentences. Three main levels can be viewed in this structure:

- a syntactic level, representing the syntactic parsing of the sentence, which can be identified as the backbone of the structure, and which can extend over several depth levels in the tree; each node is coding one syntactic label;
- the words of the sentence, with their part-of-speech tagging;
- the syllable description of the words; each node at the previous level has as many sons as syllables it is composed of.

Fig. 1 shows the syntactic structure for the sentence: "Hennessy will be a hard act to follow" extracted from the corpus. For clarity aspects, the syllable level has been omitted.

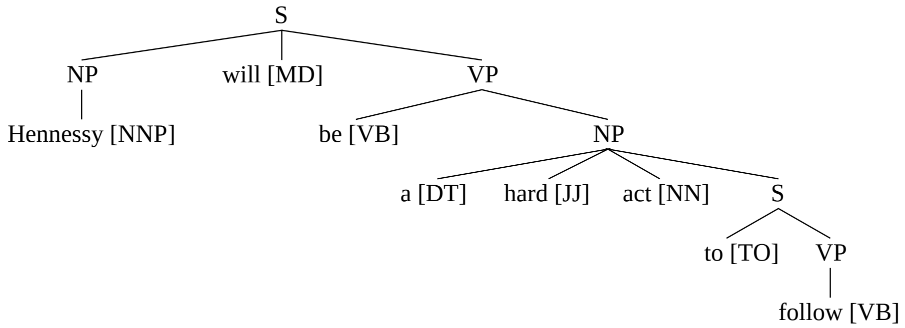

Figure1. Syntactic structure for the sentence "Hennessy will be a hard act to follow". (Syntactic labels: *S*: simple declarative clause, *NP*: noun phrase, *VP*: verb phrase. Part-of-speech labels: *NNP*: proper noun, *MD*: modal, *VB*: verb in base form, *DT*: determiner, *JJ*: adjective, *NN*: singular noun, *TO*: special label for "to")

2.2 Performance Structure

The performance structure used in our approach is a combination of syntactic and phonological informations. It can be divided in two main parts:

- the upper part of the structure is a binary tree in which each node represents a break between the two parts of the sentence contained into the subtrees of the node. This binary structure defines a hierarchy: the closer to the root the node is, the more salient (or stronger) the break is.

– the lower part represents the phonological phrases into which the whole sentence is divided by the binary structure. The subtree for each phonological phrase can be divided in three depth levels:
  - a first one to label the phrase with a syntactic category (the main one);
  - a second level for the words of the phrase; a simplification has been performed by joining them into phonological words: they are composed of one content word and of the surrounding function words (4 content words categories are considered: nouns, adjectives, verbs and adverbs);
  - a last level to represent the syllables of each phonological word of the previous level.

  No break is supposed to occur inside such a phonological phrase.

Fig. 2 shows a possible performance structure for the same example: "Hennessy will be a hard act to follow." The syllable level is also not represented.

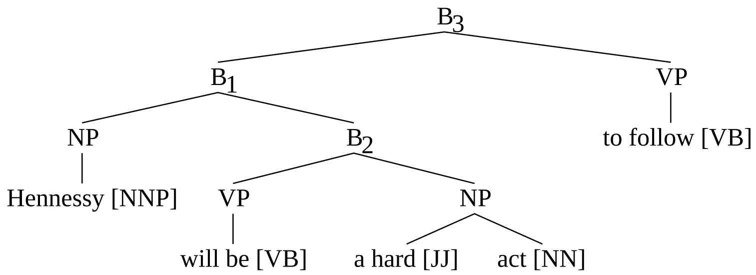

Figure2. Performance structure for the sentence "Hennessy will be a hard act to follow". The meanings of the syntactic and part-of-speech labels are identical to those in Fig. 1. B_1, B_2 and B_3 are break-related nodes.

2.3 Discussion

The syntactic structure follows the labels and parsing employed in the corpus description. Its construction presents no difficulty, for any sentence inside or outside the corpus.

However a problem occurs with the performance structure. As exposed above, this structure contains not only syntactic and part-of-speech information but also prosodic information with the break values. Building this structure for the sentences in the corpus can be done since the real prosodic values are available. Nevertheless, the aim of this work is to predict the prosody, so these data are not available in practice for a new sentence. Therefore, to achieve a prediction using this structure representation, we firstly need to predict the location and the salience of the breaks in a given sentence. The chosen method, defined by Bachenko and Fitzpatrick [8], provides rules to infer a default phrasing for a sentence. Basically, it firstly divides a sentence into phonological words and phrases (the lower parts of our structure), and then establishes the salience of the breaks between the phrases, using simple considerations about the length of the phonological phrases (defining the hierarchy of the upper binary part of our structure). Since this process furnishes an estimation of the phrasing, we will have to quantify its effects.

3 Tree Metrics

Once the tree structures have been defined, we need to determine the tools to manipulate them to predict the prosody. We have considered several similarity metrics to calculate the “distance” between two tree structures. These metrics are inspired from the Wagner and Fisher’s editing distance [9].

3.1 Principles

In an analogous way to this well know string editing distance, it is necessary to introduce a small set of elementary transformation operators between two trees: the insertion of a node, the deletion of a node, and the substitution of a node by another one. It is then possible to determine a set of specific operation sequences that transform any given tree into another one. Specifying costs for each elementary operation (possibly a function of the node values) allows the evaluation of a whole transformation cost by adding the operation costs in the sequence. Therefore the tree distance can be defined as the cost of the sequence minimizing this sum.

3.2 Considered Metrics

Many metrics can be defined from this principle. The differences come from the application conditions which can be set on the operators. In our experiments, three such tree metrics are tested. They all preserve the order of the leaves of the trees, an essential condition in our application.

The first one, defined by Selkow [10], allows only substitutions between nodes at the same depth level in the trees. Moreover, the insertion or deletion of a node involves respectively the insertion or deletion of the whole subtree depending of the node. These strict conditions should be able to locate very close structures.

The two other ones, defined by Taï [11] and Zhang [12], allow the substitutions of nodes whatever theirs locations are inside the structures. They also allow the insertion or deletion of lonely nodes inside the structures. Compared to [10], these less rigorous stipulations should not only retrieve the very close structures, but also other ones which wouldn’t have been found by the previous metric.

3.3 Operation Costs

As exposed in the algorithm principles in section 3.1, a tree is “close” to another one because of the definition of the operator costs. From this consideration, and from the definition of the tree structures in this work, these costs have been set to achieve two goals:

– to allow the only comparison of nodes of the same structural nature (break-related nodes together, syllable-related nodes together...);
– to represent the linguistic "similarity" between comparable nodes or subtrees (to set that an adjective may be "closer" to a noun than to a determiner...).

These operation costs are currently manually set. To decide on the scale of values to affect is not an easy task, and it needs some human expertise. One possibility would be to further automate the process, using machine learning techniques to set these values.

4 Prosody Prediction

Now that the tree representations and the metrics have been defined, they can be used to predict the prosody of a sentence. The simple method that we are currently using is the nearest neighbour algorithm: given a new sentence, the principle is to find the closest match among the corpus of sentences of known prosody, and then to use its prosody to infer the one of the new sentence.

From the tree metric algorithms manipulated, it is possible to retrieve the relationships which led to the final distance value: the parts of the trees which were inserted or deleted, and the ones which were substituted or unchanged. This mapping between the nodes of the two structures also links the words of the represented sentences. It then gives a simple way to know where to apply the prosody of one sentence onto the other one.

Unfortunately, this process may not be as easy. The ideal mapping would be that each word of the new sentence has a corresponding word in the closest sentence (preserving the order of the words). Hopeless, the two sentences may not be as closed as desired, and some words may have been inserted or deleted. To decide on the prosody for these words is a problem. We are currently developing a new technique based on analogy to complete this method. As exposed above, the distance provides a mapping between the two structures. We would like to find in the corpus one or more couples of sentences sharing the same tree transformation. The understanding of the impact of this transformation on the prosody should allow us to apply a similar modification onto the initial couple.

5 First Results

So far we have run experiments to find the closest match of held-out corpus sentences using the syntactic structure, and the performance structure for each of the distance metrics. We are using both the "actual" and estimated performance structures to quantify the effects of this estimation. Cross-validation tests have been chosen to validate our method.

The experiments are not all complete, but an initial analysis of the results doesn't seem to show many differences between the tree metrics considered. We believe that this is due to the small size of the corpus we are using. With only around 300 sentences, most structures are very different, so the majority of pair-wise comparisons should be very distant. We are currently running experiments where the tree structures are generated at the phrase level. This strategy implies to adapt the tree metrics to take into consideration the location of the phrases in the sentences (two similar structures should be privileged if they have the same location in their respective sentences).

6 Conclusion

We have presented a new prosody prediction method. Its original aspect is to consider sentences as treelike structures. To predict the prosody of a sentence, we are using tree similarity metrics to find the closest match in a corpus of such structures, and then its prosody is used to infer the one of the first sentence. Further experiments are needed to validate this approach.

A development of our method would be the introduction of focus labels. In a dialogue context, some extra information can refine the intonation. With the tree structures that we are using, it is easy to introduce special markers upon the nodes of the structure. According to their locations, they can indicate some focus either on a word, on a phrase or on a whole sentence. With the adaptation of the tree metrics, the prediction process is kept unchanged.

References

1. Ross, K.: Modeling of intonation for speech synthesis, PhD. Thesis, College of Engineering, Boston University (1995)
2. Traber, C.: F0 generation with a database of natural F0 patterns and with a neural network, Talking machines: theories, models and designs (1992) 287-304
3. Jensen, U., Moore, R.K., Dalsgaard, P., Lindberg, B.: Modelling intonation contours at the phrase level using continuous density hidden Markov models, Computer Speech and Language, Vol. 8 (1994) 247-260
4. Ostendorf, M., Price, P.J., Shattuck-Hufnagel, S.: The Boston University Radio News Corpus, Technical Report ECS-95-001, Boston University (1995)
5. Silverman, K., Beckman, M.E., Pitrelli, J., Ostendorf, M., Wightman, C.W., Price, P.J., Pierrehumbert, J.B., Hirschberg, J.: TOBI: A standard for labelling English Prosody, Int. Conf. on Spoken Language Processing, Vol. 2 (1992) 867–870
6. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: the Penn Treebank, Comp. Linguistics, Vol. 19 (1993)
7. Gee, J.P., Grosjean, F.: Performance structures: a psycholinguistic and linguistic appraisal, Cognitive Psychology, Vol. 15 (1983)
8. Bachenko, J., Fitzpatrick, E.: A computational grammar of discourse-neutral prosodic phrasing in English, Comp. Linguistics, Vol. 16, N. 3 (1990) 155–170
9. Wagner, R.A., Fisher, M.J.: The string-to-string correction problem, Journal of the Association for Computing Machinery, Vol. 21, N. 1 (1974) 168–173
10. Selkow, S.M., The tree-to-tree editing problem, Information Processing Letters, Vol. 6, N. 6 (1977) 184–186
11. Taï, K.C., The tree-to-tree correction problem, Journal of the Association for Computing Machinery, Vol. 26, N. 3 (1979) 422–433
12. Zhang, K., Algorithms for the constrained editing distance between ordered labeled trees and related problems, Pattern Recognition, Vol. 28, N. 3 (1995) 463–474,