In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 87–90, Lisbon, Portugal, 2000.

Generating Synthetic Speech Prosody with Lazy Learning in Tree Structures

Laurent Blin
IRISA-ENSSAT
F-22305 Lannion, France
blin@enssat.fr

Laurent Miclet
IRISA-ENSSAT
F-22305 Lannion, France
miclet@enssat.fr

Abstract

We present ongoing work on prosody prediction for speech synthesis. This approach considers sentences as tree structures and infers 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, using either the nearest neighbour algorithm or an analogy-based approach. We introduce two different tree structure representations, the tree similarity metrics considered, and then we discuss the different prediction methods. Experiments are currently under process to qualify this approach.

1 Introduction

Natural prosody production remains a problem in speech synthesis systems. Several automatic prediction methods have already been tried for this, including decision trees (Ross, 1995), neural networks (Traber, 1992), and HMMs (Jensen et al., 1994). The original aspect of our prediction approach is a tree structure representation of sentences, and the use of tree similarity measurements to achieve the prosody prediction. 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 (Ostendorf et al., 1995). The prosodic information consists of ToBI labeling of accents and breaks (Silverman et al., 1992). The syntactic and part-of-speech informations were obtained from the part of the corpus processed in the Penn Treebank project (Marcus et al., 1993).

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 (Gee and Grosjean, 1983). Their comparison in use should provide some interesting knowledge about the usefulness or the limitations of the elements of information included in each one.

2.1 Syntactic Structure

The syntactic structure considered is built exclusively from the syntactic parsing of the given sentences. This parsing, with the relative syntactic tags, constitute the backbone of the structure. Below this structure lie the words of the sentence, with their part-of-speech tags. Additional levels of nodes can be added deeper in the structure to represent the syllables of each word, and the phonemes of each syllable.

The syntactic structure corresponding to the sentence "Hennessy will be a hard act to follow" is presented in Figure 1 as an example (the syllable level has been omitted for clarity).

2.2 Performance Structure

The performance structure used in our approach is a combination of syntactic and phonological informations. Its upper part is a binary tree where 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.

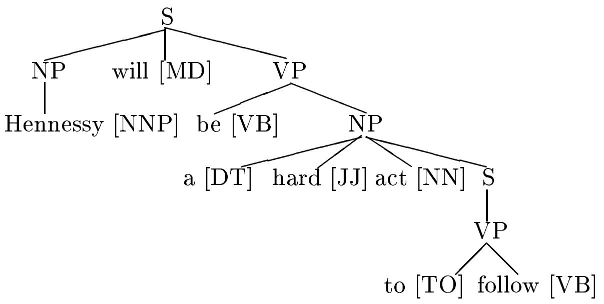

Figure 1: Syntactic structure for the sentence "Hennessy will be a hard act to follow". (Syntactic labels: *S*: simple declarative clause, *NP*: noun phrase, *VP*: verb phrase. Part-of-speech labels: *NNP*: proper noun, *MD*: modal, *VB*: base form verb, *DT*: determiner, *JJ*: adjective, *NN*: singular noun, *TO*: special label for "to").

The lower part represents the phonological phrases into which the whole sentence is divided by the binary structure, and uses the same representation levels as in the syntactic structure. The only difference comes from a simplification performed by joining the words into phonological words (composed of one content word – noun, adjective, verb or adverb – and of the surrounding function words). Each phonological phrase is labeled with a syntactic category (the main one), and no break is supposed to occur inside.

A possible performance structure for the same example: "Hennessy will be a hard act to follow" is shown in Figure 2.

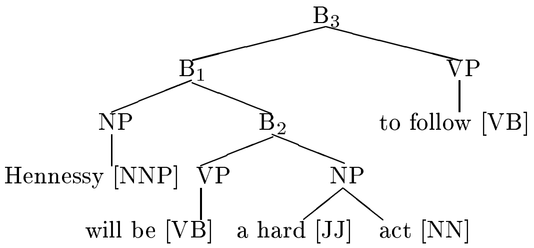

Figure 2: Performance structure for the sentence "Hennessy will be a hard act to follow". The syntactic and part-of-speech labels have the same meaning as in Figure 1. B_1, B_2 and B_3 are the break-related nodes.

Unlike the syntactic structure, a first step of prediction is done in the performance structure with the break values. This prosody information is known for the sentences in the corpus, but has to be predicted for new ones (to put our system in a full synthesis context where no prosodic value is available). The currently used method (Bachenko and Fitzpatrick, 1990) provides rules to infer a default phrasing for a sentence. Not only the effects of this estimation will have to be quantified, but we plan to develop a more accurate solution to predict this structure accordingly to any corpus speaker characteristics.

3 Tree Metrics

Now that the tree structures are defined, we need the tools to predict the prosody. We have considered two similarity metrics to calculate the "distance" between two tree structures, inspired from the Wagner and Fisher's editing distance (Wagner and Fisher, 1974).

3.1 Principles

Introducing a small set of elementary transformation operators upon trees (insertion or deletion of a node, substitution of a node by another one) it is 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, two metrics are tested. They both preserve the order of the nodes in the trees, an essential condition in our application.

The first one (Selkow, 1977) 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 other one (Zhang, 1995) allows the substitutions of nodes whatever theirs locations are inside the structures. It also allows the insertion or deletion of lonely nodes inside the structures. Compared to the previous metric, these less rigorous stipulations should not only retrieve the very close structures, but also other ones which wouldn't have been found.

Moreover, these two algorithms also provide a mapping between the nodes of the trees. This mapping illustrates the operations 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.

3.3 Operation Costs

As exposed in section 3.1, a tree is "close" to another one because of the definition of the operators costs. In this work, they have been set to allow the only comparison of nodes of same structural nature (break-related nodes together, syllable-related nodes together...), and to represent the linguistic "similarity" between comparable elements (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 for setting these values.

4 Prosody Prediction

The tree representations and the metrics can now be used to predict the prosody of a sentence.

4.1 Nearest Neighbour Prediction

The simple method that we have firstly used is the nearest neighbour algorithm: given a new sentence, the closest match among the corpus of sentences of known prosody is retrieved and used to infer the prosody of the new sentence. The mapping from the tree distance computations can be used to give a simple way to know where to apply the prosody of one sentence onto the other one, from the words linked inside.

Unfortunately, this process may not be as easy. The ideal mapping would be that each word of the new sentence had a corresponding word in the other sentence. 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 unlinked parts is a problem.

4.2 Analogy-Based Prediction

A potential way to improve the prediction is based on analogy. The previous mapping between the two structures defines a tree transformation. The idea of this approach is based on the knowledge brought by other pairs of structures from the corpus sharing the same transformation.

This approach can be connected to the analogical framework defined by Pirrelli and Yvon, where inference processes are presented for symbolic and string values by the mean of two notions: the analogical proportion, and the analogical transfer (Pirrelli and Yvon, 1999).

Concerning our problem, and given three known tree structures T_1, T_2, T_3 and a new one T', an analogical proportion would be expressed as: T_1 is to T_2 as T_3 is to T' if and only if the set of operations transforming T_1 into T_2 is equivalent to the one transforming T_3 into T', accordingly to a specific tree metric. There are various levels for defining this transformation equivalence. A strict identity would be for instance the insertion of the same structure at the same place, representing the same word (and having the same syntactic function in the sentence). A less strict equivalence could be the insertion of a different word having the same number of syllables. Weaker and weaker conditions can be set. As a consequence, these different possibilities have to be tested accordingly to the amount of diversity in the corpus to prove the efficiency of this equivalence.

Next, the analogical transfer would be to apply on the phrase described by T_3 the prosody transformation defined between T_1 and T_2 as to get the prosody of the phrase of T'. The formalization of this prosody transfer is still under development.

From these two notions, the analogical inference would be therefore defined as:

- firstly, to retrieve all analogical proportions involving T' and three known structures in the corpus;
- secondly, to compute the analogical transfer for each 3-tuple of known structures, and to store its result in a set of possible outputs if the transfer succeeds.

This analogical inference as described above may be a long task in the retrieval of every 3-tuple of known structures since a tree transformation can be defined between any pair of them. For very dissimilar structures, the set of operations would be very complex and uneasy to employ. A first way to improve this search is to keep the structure resulting of the nearest neighbour computation as T_3. The transformation between T' and T_3 should be one of the simplest (accordingly to the operations cost; see section 3.3), and then the search would be limited to the retrieval of a pair (T_1,T_2) sharing an equivalent transformation. However, this is still time-consuming, and we are trying to define a general way to limit the search in such a tree structure space, for example based on tree indexing for efficiency (Daelemans et al., 1997).

5 First Results

Because of the uncompleted development of this approach, most experiments are still under progress. So far they were run 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.

These 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 pairwise 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 tree structures. Tree similarity metrics and analogy-based learning in a corpus of such structures are used to predict the prosody of a new sentence. Further experiments are needed to validate this approach.

An additional 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 nature and location, 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

J. Bachenko and E. Fitzpatrick. 1990. A computational grammar of discourse-neutral prosodic phrasing in English. *Comp. Ling.*, 16(3):155–170.

W. Daelemans, A. van den Bosch, and T. Weijters. 1997. IGTree: Using trees for compression and classification in lazy learning algorithms. In *Artif. Intel. Review*, volume 11, pages 407–423. Kluwer Academic Publishers.

J. P. Gee and F. Grosjean. 1983. Performance structures: a psycholinguistic and linguistic appraisal. *Cognitive Psychology*, 15:411–458.

U. Jensen, R. K. Moore, P. Dalsgaard, and B. Lindberg. 1994. Modelling intonation contours at the phrase level using continuous density HMMs. *Comp. Speech and Lang.*, 8:247–260.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Comp. Ling.*, 19.

M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel. 1995. The Boston University Radio News Corpus. Technical Report ECS-95-001, Boston U.

V. Pirrelli and F. Yvon. 1999. The hidden dimension: a paradigmatic view of data-driven NLP. *J. of Exp. and Theo. Artif. Intel.*, 11(3):391–408.

K. Ross. 1995. *Modeling of intonation for speech synthesis*. Ph.D. thesis, Col. of Eng., Boston U.

S. M. Selkow. 1977. The tree-to-tree editing problem. *Inf. Processing Letters*, 6(6):184–186.

K. Silverman, M. E. Beckman, J. Pitrelli, M. Ostendorf, C. W. Wightman, P. J. Price, J. B. Pierrhumbert, and J. Hirschberg. 1992. TOBI: A standard for labelling English prosody. In *Int. Conf. on Spoken Lang. Processing*, pages 867–870.

C. Traber, 1992. *Talking machines: theories, models and designs*, chapter F0 generation with a database of natural F0 patterns and with a neural network, pages 287–304.

R. A. Wagner and M. J. Fisher. 1974. The string-to-string correction problem. *J. of the Asso. for Computing Machinery*, 21(1):168–173.

K. Zhang. 1995. Algorithms for the constrained editing distance between ordered labeled trees and related problems. *Pattern Reco.*, 28(3):463–474.