ISCA Archive
http://www.isca-speech.org/archive

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

PROSODY PREDICTION USING A TREE-STRUCTURE SIMILARITY METRIC

Laurent Blin[1] *Mike Edgington*[2]

[1]IRISA-ENSSAT, Lannion, France
[2]STAR Lab, SRI International, Menlo Park, USA

10.21437/ICSLP.2000-508

ABSTRACT

In this paper, we present ongoing work on prosody prediction for speech synthesis. Our approach considers sentences as treelike structures and decides on the prosody from a corpus of such structures 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

Over the past few years, speech synthesis has been the subject of many successful research works. While the synthesis quality has been highly improved, the production of a natural prosody still remains a difficult and challenging problem. Many automatic prediction methods have already been tried for this topic, including decision trees [1], neural networks [2], and HMMs [3]. In this work, 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. Our 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], representing an overall set of 320 sentences.

In the following sections, 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]. They have been chosen for their simplicity, their common character, and to define more than one experimentation universe. Their comparison in 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, regarding our application.

2.1. Syntactic Structure

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

Figure 1 shows the syntactic structure for the sentence: “Hennessy will be a hard act to follow”, extracted from the corpus, accordingly to the syntactic parsing given inside. For clarity aspects, the syllable level has been omitted in the representation.

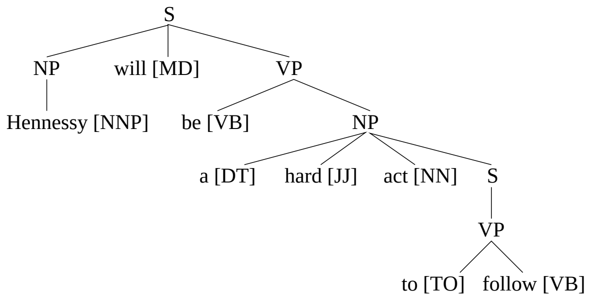

Figure 1: Syntactic structure for the sentence “Hennessy will be a hard act to follow”. The strings in square brackets are syntactic or part-of-speech tags. (The syntactic tags are: *S*: simple declarative clause, *NP*: noun phrase, *VP*: verb phrase. The part-of-speech tags are: *NNP*: proper noun, *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, part-of-speech 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.

The lower part of the structure represents the phonological phrases into which the whole sentence is divided by the binary structure, and uses mainly the same representation levels as in the syntactic structure (cf. section 2.1). A first addition is done with a main syntactic tag for each phonological phrase as to code the syntactic information, no more present in the upper part of the structure (see Figure 2 for an illustration). The second difference comes from a simplification performed by joining the words into phonological words. This is done using basic rules, gathering function words around content words (four content words categories are considered: nouns, adjectives, verbs and adverbs). Finally, no break is supposed to occur inside these phonological words, from their definition and the binary structure above them.

Figure. 2 shows a possible performance structure for the same example: "Hennessy will be a hard act to follow." The syllable representation has also been omitted.

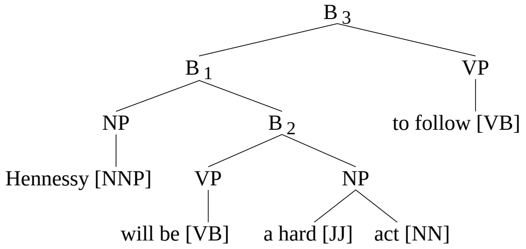

Figure 2: Performance structure for the sentence "Hennessy will be a hard act to follow". The meanings of the syntactic and part-of-speech labels are identical to those in Figure 1. B_1, B_2 and B_3 are break-related nodes. The structure illustrates that there are three breaks in this example sentence: B_3, the more important, between the words "act" and "to"; then B_1 between the words "Hennessy" and "will"; and then B_2, the less salient of the three, between the words "be" and "a".

2.3. Discussion

As exposed in section 2.1, the syntactic structure follows the labels and parsing employed in the corpus description. Its construction does not present any difficulties, for any sentence of known or unknown prosody.

However a problem occurs with the performance structure. It has been shown in section 2.2 that 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 using the tree structures (see sections 3 and 4), so these break data should not be 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 currently used 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.

This first step in the prediction is relatively simple, but it should only be considered as a temporary solution because of its default character. One sentence will always get the same default phrasing, whereas it can be pronounced differently by real speakers, with many prosodic differences for each one. To be fully adaptable with any prosodic corpus, and to reflect the prosodic characteristics of any speaker, we are trying to develop a more corpus-based approach for this phrasing prediction.

Lastly, as a corpus-based method, this work supposes correct syntactic parsing and part-of-speech labeling. Since this initial information is the realization of human annotators [6], it should be kept in mind that the accuracy of the final prediction can be affected by some errors, which should be evaluated regarding the error resulting from the prediction scheme itself.

3. TREE METRICS

In the previous section, we have presented the definition of the tree structures considered in this work. Now, 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 known string editing distance, it is necessary to introduce a small set O of elementary transformation operators between two trees:

- o_I: the insertion of a node;
- o_D: the deletion of a node;
- o_S: the substitution of a node by another one.

It is then possible to determine a set $S_{T_1T_2}$ of specific operation sequences that transform any given tree T_1 into another tree T_2. Specifying costs for each elementary operation (possibly a function c of the node values) allows the evaluation of a whole transformation cost C by adding the operation costs in the sequence:

$$C(s) = \sum_{o \in s} c(o) \; ; s \in S_{T_1T_2}. \qquad (1)$$

Therefore the tree distance D between the two trees T_1 and T_2 can be defined as the cost of the sequence minimizing this sum:

$$D(T_1, T_2) = \min_{s \in S_{T_1T_2}} C(s). \qquad (2)$$

3.2. Considered Metrics

This principle is the base for the development of many metrics. The differences come from the application conditions which can be set on the operators. In our experiments, two such tree metrics are tested.

The first one was defined by Selkow [10]. From its specifications, the insertion or deletion of a node involves respectively the insertion or deletion of the whole subtree depending of the node. Moreover, only substitutions between nodes at the same depth level in the trees are allowed. These strict conditions should permit to locate principally the very close structures, since two identical subtrees, at different depths in their respective structures, are put in relation through a large sequence of elementary operations, leading to a high distance value.

The other one, defined by Zhang [11], allows the substitutions of nodes whatever theirs locations are inside the structures. It also permits the insertion or deletion of lonely nodes in the heart of the structures. Compared to [10], these less rigorous stipulations should not only retrieve the very close structures, but also other ones which would not have been found by the previous metric.

Furthermore, these two metrics preserve the order of the nodes in the trees during a transformation, an essential condition in our application, where this order implies the order of the words in the corresponding sentences.

Finally, 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. This information will be helpful in the final prediction process.

3.3. Operation Costs

The distance algorithm principles exposed in section 3.1 have shown that a tree T_1 is said to be "close" to another tree T_2 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 main goals. The first one is to allow the only comparison of nodes of the same structural nature. For example, in performance structures, a node coding for a break in T_1 should only be compared to a node coding a break in T_2, and not to a node representing a syntactic label or a syllable. The second and most important goal is to represent the linguistic "similarity" between comparable nodes or subtrees, for instance 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 needs some human expertise. The first experiments have shown good results, but a formal validation is needed. Another possibility would be to further automate the process, using machine learning techniques to set these values.

4. PROSODY PREDICTION

Sections 2 and 3 have presented the tree representations and the metrics considered in this work. We now describe how 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 inserted, deleted, substituted and unchanged parts (see section 3.2). 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 in their corresponding structures. To decide on the prosody for these words is a problem. We are currently developing a new technique based on analogy [12], a potential way to improve and complete our method. It is based on the knowledge brought by other similar pairs of structures. 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 structures sharing the same tree transformation. The understanding of the prosody impact of an analogous transformation should allow us to apply a similar prosodic 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 does not 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 some changes. It is necessary 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 a correspondent location in their respective sentences.

In this work, we are focusing at first on the location prediction of phrase breaks and tones. Considering the two tree representations, we expect to obtain more accurate results with the performance structure than with the syntactic one. It is widely agreed upon that there is not a full correspondence between prosodic and syntactic phrases [13], and the information used in the performance structure should better reflect this compromise.

Another point to mention is the computational complexity of our approach. The tree metrics used are based on dynamic programming, a time-consuming technique, the effects of which are accentuated by the searches through the corpus. Therefore 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 [14].

6. CONCLUSION

We have presented the development of 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.

Future work will concentrate on the introduction of focus labels. In a dialogue context, some extra information can be available to precise the speech semantics, which is useful to 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. The prediction process would be kept unchanged, with a simple adaptation of the tree metrics.

7. 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, 287–304, 1992.

3. Jensen, U., Moore, R.K., Dalsgaard, P., and Lindberg, B., Modeling intonation contours at the phrase level using continuous density hidden Markov models, Computer Speech and Language, Vol. 8: 247–260, 1994.

4. Ostendorf, M., Price, P.J., and 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., and Hirschberg, J., TOBI: A standard for labeling English Prosody, International Conference on Spoken Language Processing, Vol. 2: 867–870, 1992.

6. Marcus, M.P., Santorini, B., and Marcinkiewicz, M.A., Building a large annotated corpus of English: the Penn Treebank, Computational Linguistics, Vol. 19, 1993.

7. Gee, J.P., and Grosjean, F., Performance structures: a psycholinguistic and linguistic appraisal, Cognitive Psychology, Vol. 15, 1983.

8. Bachenko, J., and Fitzpatrick, E., A computational grammar of discourse-neutral prosodic phrasing in English, Computational Linguistics, Vol. 16, N. 3: 155–170, 1990.

9. Wagner, R.A., and Fisher, M.J., The string-to-string correction problem, Journal of the Association for Computing Machinery, Vol. 21, N. 1: 168–173, 1974.

10. Selkow, S.M., The tree-to-tree editing problem, Information Processing Letters, Vol. 6, N. 6: 184–186, 1977.

11. Zhang, K., Algorithms for the constrained editing distance between ordered labeled trees and related problems, Pattern Recognition, Vol. 28, N. 3: 463–474, 1995.

12. Pirrelli, V., and Yvon, F., The hidden dimension: a paradigmatic view of data-driven NLP, Journal of Experimental and Theoretical Artificial Intelligence, Vol. 11, N. 3: 391–408, 1999.

13. Steedman, M., Intonation and Syntax in Spoken Language Systems, Speech and Natural Language Workshop, DARPA, 222–227, 1989.

14. Daelemans, W., van den Bosch, A., and Weijters, T., IGTree: Using trees for compression and classification in lazy learning algorithms, Artificial Intelligence Review, Vol. 11: 407–423, 1997.