Proceedings of the IASTED International Conference
HUMAN-COMPUTER INTERACTION
November 14-16, 2005, Phoenix, AZ, USA

OPTIMIZING SPEECH MENUS

Ivan Kopeček
Faculty of Informatics
Masaryk University Brno
Botanická 68a
602 00 Brno
Czech Republic
Email: kopecek@fi.muni.cz

ABSTRACT
The structures of speech menus and their influence on information retrieval time are modeled by ordered trees with weighted edges and a characterization of the optimal speech menus is presented. This characterization enables efficient construction of optimal menu trees. The optimization strategy is to minimize total retrieval time of the whole menus. We show that asymmetrical structures are typical for speech menus. Results of an experiment demonstrating the efficiency of the menu optimization are presented as well.

KEY WORDS
Menu-driven systems, menu trees, optimization of information retrieval, speech hypertexts and dialogue systems.

1. Introduction

For speech menus, i.e. the menus for which the menu items are presented solely in speech form, the time factor increases its meaning. The user is forced to hear the menu items in a linear order, obtaining the information substantially more slowly in comparison to visual scanning. The decision-making time is mostly negligible in comparison to the time taken to pronounce the corresponding menu item. The structure of menus substantially influences the time in which the user retrieves the requested information. In many cases the menus are quite large which can be a serious obstacle to the intelligibility of the system when directly converted to into the speech telephone and dialogue system standard VoiceXML [1,2,3].

In this paper we use mathematical formalisms of ordered trees with weighted edges for modeling speech menus and propose a method for modifying large menus by restructuring them in an optimal way. We present a criterion of optimality of speech menu trees and a characterization of optimal menu trees that enables an easy restructuring to the optimal form. We also present an experiment showing the efficiency of restructuring the speech menus in the optimal way.

Some elementary notions of graph theory are employed in this paper, especially some terms related to ordered trees. They can be found, e.g., in [4].

2. Modeling Optimality of Speech Menu Trees by Ordered Trees with Weighted Edges

Optimal structures of hypertext menus have been extensively studied, particularly in relation to graphic human-computer interfaces and web applications (see, e.g. [5-11]). Our analysis of speech menus differs in some essential aspects.

First, the access to the items of the speech menus is linear; i.e., to get access to a menu item means that the user must hear all previous menu items. The time spent by getting the menu item information is decisive due the necessity to hear the speech form of the menu item.

Second, we analyze not only symmetrical structures of the menus, but allow the menu structures to have a general form. In fact, we show that in our scenario special asymmetrical structures are optimal.

In what follows, we assume that the self-terminating search strategy is used, that is, the search is ended when the target is encountered.

As a mathematical model of menu structure we use ordered trees with weighted edges. This model considers the leaves as menu items, and its structure describes the menu structure. We introduce a function $E(x)$, which assigns a real number to any vertex of the tree T. This number expresses the time needed to get to the corresponding menu item.

Let $w(x, y)$ denote the weight of the edge (x, y). Consider an ordered tree with weighted edges and let $x \in V(T)$. Then $E(x)$ denotes the evaluation of the vertex x which is defined recursively in the following way:

1. If x is the root of T, then $E(x) = 0$.
2. If x_i is i-th successor of the vertex x, then $E(x_n) = E(x) + w(x, x_1) + \dots w(x, x_n)$.

By n-th successor of x we mean the successor of x, which has n-th position between the successors of x with respect to the ordering of the ordered tree T. Let $L(T)$ be the set of all leaves of the ordered tree T. The evaluation $E(T)$ of the tree T is then defined by

$$E(T) = \sum_{x \in L(T)} E(x)$$

When divided by the number of leaves, $E(T)$ expresses the mean access time for the menu corresponding to the tree T.

To say what trees are in some sense optimal, we have to be able to compare them. This leads us to the necessity to slightly reduce the generality of the structures we are considering. This is due to fact that we cannot in reality always know the weights of the edges between internal vertices for all possibilities.

Hence, we restrict ourselves to ordered trees with weighted leaves. An ordered tree with weighted leaves is an ordered tree with weighted edges, for which the weights of edges that connect internal vertices (i.e. the vertices that are not leaves) equal *1*. Hence, only the edges that connect an internal vertex and a leaf can have different value. Therefore, we can take the weight of an edge, which connects a leaf as the weight of the leaf. In this sense we will speak about weights of leaves in what follows.

An ordered tree T with weighted leaves is said to be *E-minimal*, if every ordered tree T_1 with weighted leaves which has the same number of leaves and the same weights of leaves as the tree T satisfies $E(T) \leq E(T_1)$. E-minimal trees correspond to optimal menus.

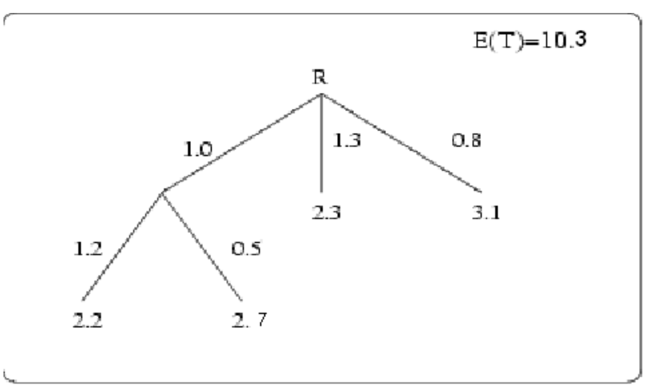

Figure 1. An example of the evaluation of the leaves of an ordered tree with weighted edges.

The exhaustive search for finding E-minimal trees would be complicated and for larger menus not feasible, as it has exponential algorithmic complexity. In order to determine E-minimal trees efficiently, the following characterization of E-minimal trees can be used, allowing us to substantially reduce the number of the trees that have to be evaluated.

We will use the following notation. If T is an ordered tree with weighted edges, then $L_L(T)$ denotes the set of all leftmost leaves (for each level) of the tree T and $L_R(T)$ the set of all rightmost leaves (for each level) of T. $w(x)$ denotes the weight of the leaf x, i.e. $w(x) = w(y, x)$, where y is the predecessor of x. $\lambda(y)$ denotes the number of the leaves in the level to which the vertex y belongs.

It is easy to see that the following assertion holds.

Proposition 1: If T is an E-minimal tree, then all leaves in the same level must be ordered with respect to their weights starting from the minimal weight, i.e. if a leaf x precedes a leaf y, then $w(x) \leq w(y)$.

We proceed with analysis of the optimal tree structures.

Proposition 2: Let T be an E-minimal tree, $x \in L_R(T)$ and $y \in L_L(T)$. Then

$$E(x) - E(y) \leq 2 + w(x) + c(y) \qquad (1)$$

where $c(y) = (1 - w(y))(\lambda(y) - 1)$.

Proof: Suppose that the condition *(1)* is violated, i.e., suppose that there exist $x \in L_R(T)$ and $y \in L_L(T)$ such that

$$E(x) - E(y) > 2 + w(x) + c(y) \qquad (2)$$

Let z be the predecessor of x and v the predecessor of y. Further, let T_1 be the ordered tree which is constructed from the tree T by removing the edge (z, x), renaming y to q and adding two new edges (q, y) and (q, x).

When calculating $E(T_1)$ from $E(T)$, removing (z, x) means to subtract $E(x)$ from $E(T)$, and adding (q, y) and (q, x) means that we have to add $(E(y) + 1) + (E(y) + 1 + w(x))$, subtract the value $E(y)$ in T and add the correction $c(x)$. The correction $c(x)$ corrects the sum of the leaf evaluations in the level $\lambda(y)$, because the edge (v, q) in T_1, corresponding to the edge (v, y) in T, has the value *1*, and the value (v, y) in T has the value $w(y)$. Hence,

$$E(T_1) = E(T) - E(x) + (E(y)+1) + (E(y)+1+ w(x)) - E(y)+c(y)$$
$$= E(T) - E(x) + E(y) + 2 + w(x) + c(y) \qquad (3)$$

However, from *(2)* and *(3)* we obtain immediately that $E(T_1) < E(T)$. Because T_1 and T have the same number of leaves, we get that if the condition *(1)* is violated, T is not E-minimal.

Corollary 1: In an E-minimal ordered tree for which all leaves have the weight *1*, each vertex may have maximally *4* successors that are leaves.

Excluding the trees that do not satisfy this condition substantially reduces the amount of trees that may be considered to be E-minimal.

Proposition 3: Let T be an E-minimal ordered tree with weighted edges, $x \in V(T) - L(T)$, $y \in L_L(T)$ and $w(y) \leq 1$. Then

$$E(x) \leq E(y) + c(y) \qquad (4)$$

where $c(y) = (1 - w(y))(\lambda(y) - 1)$.

Proof: Let the condition *(4)* be violated, i.e., suppose that there is $x \in V(T) - L(T)$ and $y \in L_L(T)$ such that $E(x) > E(y) + c(y)$.

Let T_1 be the subtree of the ordered tree with weighted edges T generated by the vertex x. This subtree consists of the vertex x, being the root of the subtree, and of all vertices that are on a directed path from x.

Further, let T_2 be the ordered tree which is constructed from the tree T by interchanging the vertices x and y. It is easy to see, that $E(T_2) < E(T)$, whereby the number of leaves of T and T_2 is the same. As in the proof of the previous proposition, the correction $c(y)$ corrects the evaluations of the leaves in the level $\lambda(y)$ of the constructed tree T_2. Hence, if the condition (4) is violated, T is not E-minimal.

Corollary 2: Let T be an E-minimal ordered tree with weighted leaves for which all leaves have the weight *1*, and $x \in V(T) - L(T)$, $y \in L_L(T)$ Then $E(x) \leq E(y)$.

Proposition 4: Let T be an E-minimal ordered tree with weighted leaves and $x, y \in L_R(T)$. Then

$$E(x) - E(y) \leq w(x). \qquad (5)$$

Proof: Let the condition *(5)* be violated, i.e., suppose that there are $x, y \in L_R(T)$ such that

$$E(x) - E(y) > w(x). \qquad (6)$$

Let z be the predecessor of x and w the predecessor of y. Further, let T_1 be the ordered tree which is constructed from the tree T by removing the edge *(z, x)* and adding a new edge *(w, q)* as the rightmost successor of the edge w. When calculating $E(T_1)$ from $E(T)$, removing *(z, x)* means that we have to subtract $E(x)$ from $E(T)$ and adding *(w, q)* means that we have to add $E(y) + w(x)$. Hence,

$$E(T_1) = E(T) - E(x) + E(y) + w(x) \qquad (7)$$

From *(6)* and *(7)* we get immediately that $E(T_1) < E(T)$. It means, that if the condition *1* is violated, T is not E-minimal.

Corollary 3: Let T be an E-minimal ordered tree with weighted leaves for which all leaves have the weight *1* and $x, y \in L_R(T)$. Then $E(x) - E(y) \leq 1$.

The presented characterization of E-minimal trees enables a more efficient computation of E-minimal menus for a given number of leaves.

Modification of a menu structure can be performed by adding some inner menu items describing groups of original menu items. This can be done either manually when building a speech dialogue systems or a speech hypertext system, or, for some simple cases, in an automated way, by analyzing the original menu items. The semantic analysis based on Wordnet structures [12,13] and Semantic Web approach [14] may be utilized.

3. Experimental Evaluation

To verify the theoretical approach, we have performed a simple experiment which tested the assumption that the mean access time for menu items is approximately proportional to the function $E(T)$. Two menus were compared; a linear menu consisting of the root vertex and sixteen leaves (E(T) = 136), and its optimal variant plotted in Fig. 2 (E(T) =66). For both ordered trees that correspond to the menus all the weights of the leaves equal 1.

A total of 10 people participated in the experiment. The participants' age ranged form 19 to 58 (mean=32.4). To measure the time, the real time returned by Unix command *time* was utilized. VoiceXML platform OptimTalk [15] was used for testing.

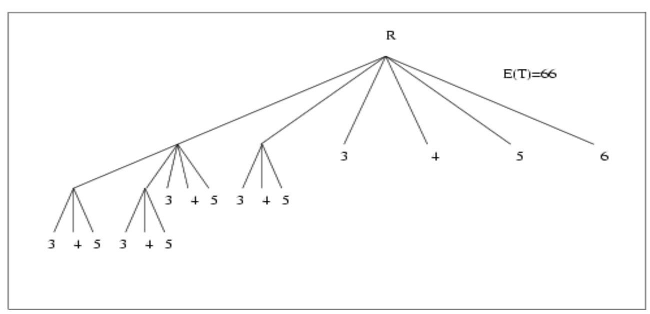

Figure 2. The optimized menu used in the experiment

The achieved mean access time for the linear menu was 8.86 seconds with standard deviation 4.83 seconds, for the optimized menu 5.72 seconds with standard deviation 1.15 seconds. Fig. 3 shows the mean access time for individual items of both menus. The ratio between the mean access time for the optimized and linear menu was 0.65. The expected value is 0.49.

The difference between the theoretical and measured values is caused by several reasons; some nonlinear delays caused by OptimTalk interpreter, different reaction times of the users, computer overall load and inaccuracy in the measurement. Even though the experiment shows that the practical results correspond to the theoretical expectations only approximately, it also demonstrates the efficiency of restructuring the menu into optimal form.

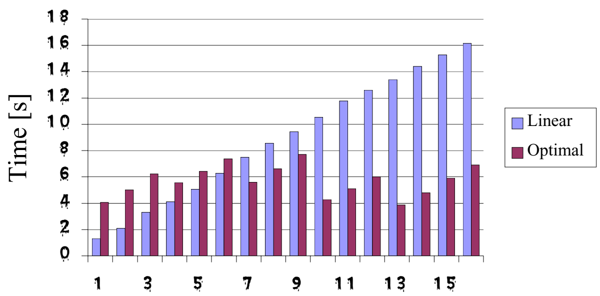

Figure 3. Comparison of mean access time values for individual menu items

4. Conclusion

In the paper, we have shown that optimization of speech menus substantially reduces the access time. An interesting open issue, related to the problem of HTML to VoiceXML conversion, is an (semi)automatic modification of menu structures. Another interesting open problem is to find a full characterization of E-optimal tree structures. Also, more detailed experimental work is another important goal for future research.

Acknowledgement

The author is grateful to James Thomas for reading the draft of the paper.

References

[1] L. Bártek, Automatic Generation of Dialogue Interaces for Web-Based Applications. In *Proc. Text, Speech and Dialogue*, Berlin Heidelberg : Springer-Verlag, LNAI 2166, 443-449, 2001.

[2] *Voice Extensible Markup Language (VoiceXML) Version 2.0*, http://www.w3.org/TR/2004/REC-voicexml20-20040316/

[3] W.K. Edwards, E.D. Mynatt, An Architecture for Transforming Graphical Interfaces. *Proceedings of 7th ACM Symposium on User Interface Software and Technology*, 39-47, 1994.

[4] J. Gross, J. Yellen, *Graph Theory and its Applications*, (CRC Press, London 1999).

[5] D.L. Fisher, E. Yungkurth, S.M. Moss, Optimal Menu Hierarchy Design: Syntax and Semantics. *Human Factors 32*, 665-683, 1990.

[6] J. Jacko, G. Salvendy, Hierarchical Menu Design: Breadth, Depth, and Task Complexity. *Perceptual and Motor Skills, 82*, 1187-1201, 1996.

[7] J. Kleiss, Identifying User's Conceptual Organization of Menu Functions in an Automotive Electronic Navigation Product, *Proceedings of the Human Factors Society, 41th Annual Meeting,* 944-948, 1997.

[8] E.S. Lee, Menu-Driven Systems. in *The Encyclopedia of Microcomputers*, Volume 11, A. Kent and J. G. Williams (eds.), 101-127, 1993.

[9] S. Lewis, Cluster Analysis as a Technique to Guide Interface Design. *International Journal of Man-Machine Studies, 35*, 251-265, 1991.

[10] K. Norman, *The Psychology of Menu Selection* (Norwood, New Jersey, Ablex Publishing Corporation, 1991).

[11] M.L. Toms, M.A. Cummings-Hill, D.G. Curry, S.M. Cone, Using Cluster Analysis for Deriving Menu Structures for Automotive Mobile Multimedia Applications, *SAE Technical Paper Series,* 2001-01-0359, 2001.

[12] P. Vossen, P. (ed.), *EuroWordNet: A Multilingual Database with Lexical Semantic Networks* (Dordrecht, Kluwer Academic Publishers, 1998).

[13] C. Fellbaum, C. (ed.), *WordNet - An Electronic Lexical Database* (Bradford Books, 1998).

[14] *The Semantic Web: An Introduction*, http://infomesh.net/2001/swintro/

[15] *Optimsys, OptimTalk Platform*, http://optimtalk.cz