INVESTIGATIONS ON JOINT-MULTIGRAM MODELS FOR GRAPHEME-TO-PHONEME CONVERSION

M. Bisani and H. Ney

Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen – University of Technology, D-52056 Aachen, Germany
{bisani,ney}@informatik.rwth-aachen.de

ABSTRACT

We present a fully data-driven, language independent way of building a grapheme-to-phoneme converter. We apply the joint-multigram approach to the alignment problem and use standard language modelling techniques to model transcription probabilities. We study model parameters, training procedures and effects of corpus size in detail. Experiments were conducted on English and German pronunciation lexica. Our proposed training scheme performs better than previously published ones. Phoneme error rates as low as 3.98% for English and 0.51% for German were achieved.

1. INTRODUCTION

The task of grapheme-to-phoneme conversion, or phonetic transcription, can be formalized using Bayes' decision rule as

$$\boldsymbol{\varphi}(\boldsymbol{g}) = \operatorname*{argmax}_{\boldsymbol{\varphi}' \in \Phi^*} p(\boldsymbol{\varphi}', \boldsymbol{g}) \tag{1}$$

This means, for a given orthographic form (sequence of letters) $\boldsymbol{g} \in G^*$ we seek the most likely pronunciation (phoneme sequence) $\boldsymbol{\varphi} \in \Phi^*$.

Most work on grapheme-to-phoneme conversion has neglected the alignment problem. A popular approach is using hand-crafted rules to align letters and phonemes (e.g. [1]). Only after this alignment has been produced, machine learning techniques are applied to perform the actual mapping. In developing a grapheme-to-phoneme conversion system for a new language it is inconvenient to write alignment rules by hand. But doing with just one-to-one alignment does not give acceptable results. Furtunatelly alignments can be inferred using joint-multigram models, an approach pioneered by S. Deligne, F. Yvon and F. Bimbot [2][3].

2. JOINT MULTIGRAM MODELS

For the convenience of the reader we provide a brief review of the joint-multigram model in the context of grapheme-to-phoneme conversion [2]. A grapheme-phoneme joint multigram, or *graphone* for short, is a pair $q = (\boldsymbol{g}, \boldsymbol{\varphi}) \in Q \subseteq G^* \times \Phi^*$ of a letter sequence and a phoneme sequence of possibly different length. We use the expressions $\boldsymbol{g}_q$ and $\boldsymbol{\varphi}_q$ to refer to the first and second component of q respectively. In the joint multigram model we assume that for each word its orthographic form and its pronunciation are generated by a common sequence of graphones.

For example, the pronunciation of "speaking" may be regarded as a sequence of five graphones:

"speaking"	=	s	p	ea	k	ing
[spiːkɪŋ]		[s]	[p]	[iː]	[k]	[ɪŋ]

However the segmentation into graphones may be not unique. The joint probability $p(\boldsymbol{\varphi}, \boldsymbol{g})$ is determined by summing over all matching graphone sequences:

$$p(\boldsymbol{\varphi}, \boldsymbol{g}) = \sum_{\boldsymbol{q} \in S(\boldsymbol{g}, \boldsymbol{\varphi})} p(q_1, \ldots, q_L) \tag{2}$$

where $S(\boldsymbol{g}, \boldsymbol{\varphi})$ is the set of all joint segmentations of $\boldsymbol{g}$ and $\boldsymbol{\varphi}$.

$$S(\boldsymbol{g}, \boldsymbol{\varphi}) := \left\{ q_1^L \in Q^* \,\middle|\, \begin{matrix} \boldsymbol{g}_{q_1} \sqcup \ldots \sqcup \boldsymbol{g}_{q_L} \\ \boldsymbol{\varphi}_{q_1} \sqcup \ldots \sqcup \boldsymbol{\varphi}_{q_L} \end{matrix} = \begin{matrix} \boldsymbol{g} \\ \boldsymbol{\varphi} \end{matrix} \right\} \tag{3}$$

The joint probability distribution $p(\boldsymbol{\varphi}, \boldsymbol{g})$ has thus been reduced to a probability distribution over graphone sequences $p(\boldsymbol{q})$ which we model using a standard M-gram:

$$p(q_1^L) = \prod_{i=1}^{L+1} p(q_i | q_{i-1}, \ldots, q_{i-M+1}) \tag{4}$$

where positions $i < 1$ and $i > L$ are virtually understood to contain a special boundary symbol $q_i = \perp$ which allows modelling of characteristic phenomena at word starts and ends.

$$p(q_1^L) = p(\perp | q_L \ldots) \cdots p(q_2 | q_1, \perp) p(q_1 | \perp) \tag{5}$$

3. TRAINING

Given a training sample $(\boldsymbol{g}_1, \boldsymbol{\varphi}_1), \ldots, (\boldsymbol{g}_N, \boldsymbol{\varphi}_N)$, parameter estimation is performed in two separate phases. In the first phase the graphone set Q is inferred using only unigram statistics ($M = 1$). The resulting unigram graphone model is then used to co-segment the corpus into a stream of graphones according to

$$\boldsymbol{q}_i = \operatorname*{argmax}_{\boldsymbol{q}' \in S(\boldsymbol{g}_i, \boldsymbol{\varphi}_i)} p(\boldsymbol{q}') \tag{6}$$

The segmented corpus $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N$ is then used in the second phase to train the M-gram model $p(q_i | q_{i-1}, \ldots, q_{i-M+1})$ using standard techniques. In this work we used bi- and trigram models with absolute discounting, estimating discount parameters using leaving-one-out [4].

Integrated optimization of the M-gram probabilities should be possible in principle but has not been tried. In the following we focus on the inference of the multigram set, i.e. training of the unigram probabilities.

3.1. Maximum Likelihood Training

Maximum likelihood training can be performed using the expectation maximization (EM) algorithm. In the case of unigrams we can identify the model paramters with the uni-graphone probaility $\vartheta_q \equiv p(q;\boldsymbol{\vartheta})$. The re-estimation equations for the updated parameters $\boldsymbol{\vartheta}'$ are:

$$p(\boldsymbol{q};\boldsymbol{\vartheta}) = \prod_{i=1}^{|\boldsymbol{q}|} \vartheta_{q_i} \quad (7)$$

$$e(q;\boldsymbol{\vartheta}) := \sum_{i=1}^{N} \sum_{\boldsymbol{q}\in S(\boldsymbol{g}_i,\boldsymbol{\varphi}_i)} p(\boldsymbol{q}|\boldsymbol{g}_i,\boldsymbol{\varphi}_i;\boldsymbol{\vartheta})\, n_q(\boldsymbol{q}) \quad (8)$$

$$= \sum_{i=1}^{N} \sum_{\boldsymbol{q}\in S(\boldsymbol{g}_i,\boldsymbol{\varphi}_i)} \frac{p(\boldsymbol{q};\boldsymbol{\vartheta})}{\sum_{\boldsymbol{q}'\in S(\boldsymbol{g}_i,\boldsymbol{\varphi}_i)} p(\boldsymbol{q}';\boldsymbol{\vartheta})} n_q(\boldsymbol{q})$$

$$\vartheta'_q = \frac{e(q;\boldsymbol{\vartheta})}{\sum_{q'} e(q';\boldsymbol{\vartheta})} \quad (9)$$

where $n_q(\boldsymbol{q})$ is number of occurences of q in $\boldsymbol{q}$. The quantity $e(q;\boldsymbol{\vartheta})$, which we call the *evidence* for q, is the expected number of occurences of the graphone q in the training sample under the current set of parameters $\boldsymbol{\vartheta}$. The evidence can be calculated efficiently by a forward-backward procedure [3].

Obviously the above equations do not permit a new graphone to emerge once its probability is zero. Therefore we initialize the model parameters by assigning a uniform distribution to all graphones satisfying certain manually set length constraints. We will use the notation $|\boldsymbol{g}_q| = l_{\min}\ldots l_{\max}$ to indicate that only graphones with at least $l_{\min}$ and at most $l_{\max}$ letters were considered, and $|\boldsymbol{\varphi}_q| = r_{\min}\ldots r_{\max}$ likewise for the number of phonemes.

3.2. Evidence Trimming

Not all graphones satisfying the length constraints are helpful to the transcription task. On the contrary, most of them will receive negligibly small probabilities, and, as we will see later, smaller graphone inventories generally yield better results. To obtain a reasonably sized models we apply thresholding to the evidence values, i.e. in equation (9) we use

$$\hat{e}(q;\boldsymbol{\vartheta}) = \begin{cases} 0 & \text{if } e(q;\boldsymbol{\vartheta}) < \tau \\ e(q;\boldsymbol{\vartheta}) & \text{otherwise} \end{cases} \quad (10)$$

We call this procedure *evidence trimming* and find that it causes the unlikely graphones to gradually die out during the iteration process. (Actually there is always implicit trimming caused by the limited machine precision.) Evidence trimming is superior to model trimming where a similar thresholding is applied to the probability estimates ϑ_q. This is because even graphones with low probabilities $p(q;\boldsymbol{\vartheta})$ can have a conditional probability $p(q|\boldsymbol{g}_i,\boldsymbol{\varphi}_i;\boldsymbol{\vartheta})$ of one in certain words; trimming them would leave the training sample not representable by the model.

3.3. Training with Maximum Approximation

Earlier experiment with the joint multigram approach [2] used the maximum approximation during training. Therefore we have tried this strategy as well. Like in earlier work, we have found that this so-called *Viterbi*-training is very sensitive to initialization and careful selection of graphone trimming thresholds. In particular it is necessary to initialize unigram probabilities proportional to the occurence counts, which is equivalent to setting $p(\boldsymbol{q}|\boldsymbol{g}_i,\boldsymbol{\varphi}_i;\boldsymbol{\vartheta}) = 1$ in equation (8).

4. TRANSCRIPTION

In producing the phonemic transcription from the orthographic form, we restrict ourselves to the maximum approximation:

$$p(\boldsymbol{\varphi},\boldsymbol{g}) \approx \max_{\boldsymbol{q}\in S(\boldsymbol{g},\boldsymbol{\varphi})} p(q_1,\ldots,q_L) \quad (11)$$

This means, we look for the most likely graphone sequence matching the given spelling and project it onto the phonemes. This is performed using a straight-forward A^* implementation using a zero rest-cost term.

5. EXPERIMENTS

We conducted experiments on a German and an English transcription task which we constructed from available pronunciation dictionaries.

For English we used the CELEX Lexical Database of English (version 2.5) [5]. Phrases and abbreviations were removed. All words were converted to lower case, resulting in the usual 26 grapheme symbols. The phoneme set consists of 53 symbols (12 vowels, 8 diphtongs, 4 nasalized vowels, 24 consonants, 3 syllabic consonants, 2 affricates), though some of them are extremely rare. The preprocessed database contains 66278 word forms.

For German we used the Bielefeld Lexicon Database VM-II, version 14.0 (LEXDB) [6]. Preprocessing steps included removal of hyphenated compounds, abbreviations and pronunciation variants. All words were converted to lower case, resulting in 30 grapheme symbols (including 3 umlauts and sz-ligature). The phoneme set consists of 46 symbols (18 vowels, 3 diphtongs, 21 consonants, 4 affricates). After preprocessing there were 71358 word forms.

From each database we randomly selected an evaluation test set of 15000 words and a training set of 40000 words, which are disjoint, of course. Details about the corpus sizes can be found in table 1. Performance is measured by the *phoneme error rate*, which is the Levenshtein distance[1] between automatic transcription result and reference pronunciation divided by the number of phonemes in the reference pronunciation.

Table 1. Statistics of the corpora used

	LEXDB German		CELEX English	
	train	eval	train	eval
words	40,000	15,000	40,000	15,000
graphemes	417,264	156,497	334,583	125,696
phonemes	359,750	134,858	282,732	106,143

The minimum graphone length was one letter and one phoneme in all experiments. As for the maximum length we tried

[1] This is the minimum number of insert, delete and substitute operations required to transform one sequence into the other.

Table 2. Selected results using marignal trimming (40k words training sets)

English

length constraints $\lvert g_q \rvert$	length constraints $\lvert \varphi_q \rvert$	$\lvert Q \rvert$	phoneme error rate [%] $M=1$	$M=2$	$M=3$
1 . . . 1	1 . . . 1	417	53.02	37.93	34.31
1 . . . 2	1 . . . 1	1155	34.18	12.92	6.38
1 . . . 2	1 . . . 2	1920	30.38	7.20	**4.02**
1 . . . 3	1 . . . 1	1119	31.66	12.76	6.35
1 . . . 3	1 . . . 2	3847	24.46	6.26	4.41
1 . . . 3	1 . . . 3	7313	20.20	**5.22**	4.77
1 . . . 4	1 . . . 4	15789	13.78	6.22	6.29
1 . . . 5	1 . . . 5	21637	10.42	7.30	7.28
1 . . . 6	1 . . . 6	26319	**9.83**	8.68	9.10

German

length constraints $\lvert g_q \rvert$	length constraints $\lvert \varphi_q \rvert$	$\lvert Q \rvert$	phoneme error rate [%] $M=1$	$M=2$	$M=3$
1 . . . 1	1 . . . 1	170	41.54	31.59	29.98
1 . . . 2	1 . . . 1	521	20.20	4.16	0.89
1 . . . 2	1 . . . 2	1120	14.08	0.94	**0.52**
1 . . . 3	1 . . . 1	431	17.80	4.15	0.89
1 . . . 3	1 . . . 2	1611	9.92	0.85	0.53
1 . . . 3	1 . . . 3	3370	6.58	**0.72**	0.70
1 . . . 4	1 . . . 4	5762	3.67	0.96	1.00
1 . . . 5	1 . . . 5	8062	2.82	1.56	1.58
1 . . . 6	1 . . . 6	11181	**2.79**	2.27	2.30

all combinations of length constraints up to six symbols on both sides.

We experimented with the setting of the trimming threshold τ and found that the resulting model is affected mostly by the value of τ during the first couple of iterations. In later iterations τ can be increased to speed up convergence without changing the result significantly. A first series of tests (cf. table 2) was conducted with what we call *marginal trimming*: Starting with very small values (10^{-15}) τ is increased gradually (by a factor of ten in five iterations) up to a maximum value of 0.1. Additional test used higher, but constant thresholds (cf. table 6).

To see how performance is affected by the amount of training data available, we repeated some of the experiments on training sets of 5, 10 and 20 thousand words (cf. table 3).

6. RESULTS AND DISCUSSION

In summary the phoneme error rates are lower on the German task because the spelling is closer to the pronunciation than in English. (Interestingly also the number of inferred multigrams $|Q|$ is smaller for German.) Apart from that, all result are structurally similar. The best phoneme error rate obtained with marginal trimming for German is 0.52%, for English 4.02%, which seems quite competitive, given the simplicity of the model.[2]

The large error rates for the experiments where the graphone length was restricted to one letter, proves the importance of a proper alignment model. For the unigram model, error rates decrease as longer and longer graphones are considered. Also we find that in the unigram case, marginal trimming yields the best results in all cases.

For higher M-gram model the picture is less clear: On the one hand longer graphones cover a larger context. On the other hand, larger allowed graphone sizes imply that the M-model has to handle a larger number of symbols, which naturally leads to sparseness problems. Therefore the bigram and trigram error rates go up if the graphone lengths are increased beyond three or two respectively.

Applying stronger trimming generally has a negative effect on the unigram error rate, but is effective in restricting the size of the model and consequently keeping the bi- and trigram error rates low (cf. fig. 1). Optimizing τ on the trigram phoneme error rate can slighly improve upon the best results of the marginal trimming strategy in some cases (cf. table 6).

In reducing the amount of training data, we observe that longer graphones become harder to estimate reliably. Therefore the optimal length restrictions decrease (cf. table. 3).

The maximum approximation in training causes infrequent graphones to die out more quickly; sometimes too quickly, making the algorithm more prone to local optima. Careful evidence trimming is needed te achieve good performance. In the unigram case the (true) EM algortihm with summation was consistently superior to Viterbi training (cf. table 4); and had the additional advantange of not having to optimize the trimming parameters. In the trigram case the EM algorithm (with summation) is still slightly superior, but looses this additional advantage (cf. table 5). We have to apply strong trimming in both cases to avoid sparseness problems. This is most likely because for M-gram training we resort to the maximum approximation anyway.

Table 3. Results using differently sized training sets and marignal trimming. (Only best unigram and trigram results shown)

English

training set	length constraints $\lvert g_q \rvert$	length constraints $\lvert \varphi_q \rvert$	$\lvert Q \rvert$	PER [%] $M=1$	$M=3$
5000	1 . . . 4	1 . . . 4	6337	22.04	18.77
10000	1 . . . 4	1 . . . 4	8486	17.94	13.80
20000	1 . . . 5	1 . . . 5	15046	13.38	11.30
40000	1 . . . 6	1 . . . 6	26319	9.83	9.10
5000	1 . . . 3	1 . . . 1	619	32.02	11.51
10000	1 . . . 2	1 . . . 2	1396	30.39	9.14
20000	1 . . . 2	1 . . . 2	1658	30.45	6.32
40000	1 . . . 2	1 . . . 2	1920	30.38	4.02

German

training set	length constraints $\lvert g_q \rvert$	length constraints $\lvert \varphi_q \rvert$	$\lvert Q \rvert$	PER [%] $M=1$	$M=3$
5000	1 . . . 4	1 . . . 4	3472	7.48	5.78
10000	1 . . . 4	1 . . . 4	3656	6.20	4.13
20000	1 . . . 5	1 . . . 5	6226	4.11	3.07
40000	1 . . . 5	1 . . . 6	11181	2.79	2.30
5000	1 . . . 3	1 . . . 1	258	17.81	1.94
10000	1 . . . 4	1 . . . 1	291	17.80	1.48
20000	1 . . . 2	1 . . . 2	1025	14.11	0.89
40000	1 . . . 2	1 . . . 2	1120	14.08	0.52

[2] Unfortunatelly we cannot provide direct comparison with other methods, but to get a rough idea: Torkolla [1] reports a mapping accuracy of 90.8% on an English task with 18000 words for training. Besling [7] reports a phoneme error rate of 3.55% on a German task with 103766 words for training. Please keep in mind that the conditions used in those studies were possibly harder.

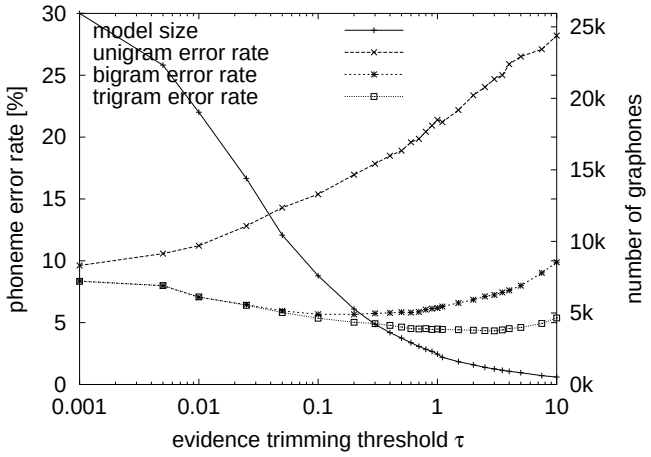

Fig. 1. Effect of the evidence trimming threshold τ on model size and error rates for different M-gram models (English; 40k training set; length constraints: $|\boldsymbol{g}_q| = 1 \ldots 6$, $|\boldsymbol{\varphi}_q| = 1 \ldots 6$)

Table 4. Comparison of unigram results using Viterbi and EM training (40k training set; $M = 1$; trimming optimized only for Viterbi)

English

length constraints		Viterbi		EM	
$\lvert\boldsymbol{g}_q\rvert$	$\lvert\boldsymbol{\varphi}_q\rvert$	$\lvert Q\rvert$	PER [%]	$\lvert Q\rvert$	PER [%]
$1 \ldots 2$	$1 \ldots 2$	813	30.41	1920	30.38
$1 \ldots 3$	$1 \ldots 3$	3776	20.64	7313	20.20
$1 \ldots 4$	$1 \ldots 4$	16267	14.94	15789	13.78

German

length constraints		Viterbi		EM	
$\lvert\boldsymbol{g}_q\rvert$	$\lvert\boldsymbol{\varphi}_q\rvert$	$\lvert Q\rvert$	PER [%]	$\lvert Q\rvert$	PER [%]
$1 \ldots 2$	$1 \ldots 2$	1113	14.67	1120	14.08
$1 \ldots 3$	$1 \ldots 3$	2719	7.19	3370	6.58
$1 \ldots 4$	$1 \ldots 4$	6100	3.79	5762	3.67

Table 5. Comparison of trigram results using Viterbi and EM training (40k training sets; $M = 3$; trimming optimized in Viterbi and EM training)

English

length constraints		Viterbi		EM	
$\lvert\boldsymbol{g}_q\rvert$	$\lvert\boldsymbol{\varphi}_q\rvert$	$\lvert Q\rvert$	PER [%]	$\lvert Q\rvert$	PER [%]
$1 \ldots 2$	$1 \ldots 2$	1775	3.99	1714	3.98
$1 \ldots 3$	$1 \ldots 3$	1673	4.42	1474	4.29
$1 \ldots 4$	$1 \ldots 4$	1681	4.29	1596	4.24

German

length constraints		Viterbi		EM	
$\lvert\boldsymbol{g}_q\rvert$	$\lvert\boldsymbol{\varphi}_q\rvert$	$\lvert Q\rvert$	PER [%]	$\lvert Q\rvert$	PER [%]
$1 \ldots 2$	$1 \ldots 2$	1101	0.52	1126	0.51
$1 \ldots 3$	$1 \ldots 3$	2069	0.61	1714	0.54
$1 \ldots 4$	$1 \ldots 4$	2089	0.66	1760	0.58

Table 6. Selected results with trimming optimized for the trigram model (40k training sets; $M = 3$)

English

length constraints				phoneme
$\lvert\boldsymbol{g}_q\rvert$	$\lvert\boldsymbol{\varphi}_q\rvert$	τ_{opt}	$\lvert Q\rvert$	error rate [%]
$1 \ldots 2$	$1 \ldots 2$	0.4	1714	**3.98**
$1 \ldots 4$	$1 \ldots 4$	3.0	1121	4.38
$1 \ldots 6$	$1 \ldots 6$	3.0	1087	4.34

German

length constraints				phoneme
$\lvert\boldsymbol{g}_q\rvert$	$\lvert\boldsymbol{\varphi}_q\rvert$	τ_{opt}	$\lvert Q\rvert$	error rate [%]
$1 \ldots 2$	$1 \ldots 2$	0.25	1126	**0.51**
$1 \ldots 4$	$1 \ldots 4$	0.6	1760	0.58
$1 \ldots 6$	$1 \ldots 6$	0.6	1627	0.60

7. SUMMARY AND OUTLOOK

We have investigated several variations on the multigram approach to grapheme-to-phoneme conversion. Experiments on German and English demonstrate that very good performance can be achieved with relatively simple models. We have shown that evaluating the sum in the EM training algorithm yields consistently better results than using the maximum-approximation and allows us to get by with fewer empirical parameters.

Currently we train the M-gram models in a separate step at the same time resorting to a maximum approximation. Our results seems to indicate that using an integrated training procedure, which optimizes M-gram probabilities and graphone boundaries simultaneously, might be beneficial.

Acknowledgments: This work was partially funded by the European Commission under the Human Language Technologies project CORETEX (IST-1999-11876).

8. REFERENCES

[1] K. Torkkola, "An efficient way to learn englisch grapheme-to-phoneme rules automatically," in *Proc. ICASSP*, Minneapolis (MN), USA, Apr. 1993, vol. 2, pp. 199 – 202.

[2] S. Deligne, F. Yvon, and F. Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," in *Proc. Eurospeech*, Madrid, Sep. 1995, pp. 2243 – 2246.

[3] S. Deligne and F. Bimbot, "Inference of variable-length acoustic units for continuous speech recognition," *Speech Communication*, vol. 23, pp. 223–241, 1997.

[4] H. Ney, S. Martin, and F. Wessel, "Statistical language modeling using leaving-one-out," in *Corpus-Based Methods in Language and Speech Processing*, S. Young and G. Bloothooft, Eds., pp. 174 – 207. Kluwer, 1997.

[5] "CELEX lexical database," http://www.kun.nl/celex/.

[6] H. Lüngen, K. Ehlebracht, D. Gibbon, and A. P. Q. Simões, "Bielefelder Lexikon und Morphologie in VERBMOBIL Phase II," Tech. Rep. ISSN 1434-8845, Universität Bielefeld, November 1998.

[7] S. Besling, "Heuristical and statistical methods for grahpeme-to-phoneme conversion," in *Konferenz zur Verarbeitung natürlicher Sprache (KONVENS)*, Vienna, Austria, Sep. 1994, pp. 24 – 31.