COMBINATION OF N-GRAMS AND STOCHASTIC CONTEXT-FREE GRAMMARS FOR LANGUAGE MODELING[*]

José-Miguel Benedí and Joan-Andreu Sánchez
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia (Spain)
e-mail: {jbenedi,jandreu}@dsic.upv.es

Abstract

This paper describes a hybrid proposal to combine n-grams and Stochastic Context-Free Grammars (SCFGs) for language modeling. A classical n-gram model is used to capture the local relations between words, while a stochastic grammatical model is considered to represent the long-term relations between syntactical structures. In order to define this grammatical model, which will be used on large-vocabulary complex tasks, a category-based SCFG and a probabilistic model of word distribution in the categories have been proposed. Methods for learning these stochastic models for complex tasks are described, and algorithms for computing the word transition probabilities are also presented. Finally, experiments using the Penn Treebank corpus improved by 30% the test set perplexity with regard to the classical n-gram models.

1 Introduction

Language modeling is an important aspect to consider in large-vocabulary speech recognition systems (Bahl et al., 1983; Jelinek, 1998). The n-gram models are the most widely-used for a wide range of domains (Bahl et al., 1983). The n-grams are simple and robust models and adequately capture the local restrictions between words. Moreover, it is well-known how to estimate the parameters of the model and how to integrate them in a speech recognition system. However, the n-gram models cannot adequately characterize the long-term constraints of the sentences of the tasks.

On the other hand, Stochastic Context-Free Grammars (SCFGs) allow us a better modeling of long-term relations and work well on limited-domain tasks of low perplexity. However, SCFGs work poorly for large-vocabulary, general-purpose tasks because learning SCFGs and the computation of word transition probabilities present serious problems for complex real tasks.

In the literature, a number of works have proposed ways to generalize the n-gram models (Jelinek, 1998; Siu and Ostendorf, 2000) or combining with other structural models (Bellegarda, 1998; Gilet and Ward, 1998; Chelba and Jelinek, 1998).

In this paper, we present a combined language model defined as a linear combination of n-grams, which are used to capture the local relations between words, and a stochastic grammatical model which is used to represent the global relation between syntactic structures. In order to capture these long-term relations and to solve the main problems derived from the large-vocabulary complex tasks, we propose here to define: a category-based SCFG and a probabilistic model of word distribution in the categories. Taking into account this proposal, we also describe here how to solve the learning of these stochastic models and their integration problems.

With regard to the learning problem, several algorithms that learn SCFGs by means of estimation algorithms have been proposed (Lari and Young, 1990; Pereira and Schabes, 1992; Sánchez and Benedí, 1998), and promising results have been achieved with category-based SCFGs on real tasks (Sánchez and Benedí, 1999).

In relation to the integration problem, we present two algorithms that compute the word transition probability: the first algorithm is based on the Left-to-Right Inside algorithm

[*] This work has been partially supported by the Spanish CICYT under contract (TIC98/0423-C06).

(LRI) (Jelinek and Lafferty, 1991), and the second is based on an application of a Viterbi scheme to the LRI algorithm (the VLRI algorithm) (Sánchez and Benedí, 1997).

Finally, in order to evaluate the behavior of this proposal, experiments with a part of the Wall Street Journal processed in the Penn Treebank project were carried out and significant improvements with regard to the classical n-gram models were achieved.

2 The language model

An important problem related to language modeling is the evaluation of $\Pr(w_k \mid w_1 \ldots w_{k-1})$. In order to compute this probability, we propose a hybrid language model defined as a simple linear combination of n-gram models and a stochastic grammatical model G_s:

$$\begin{aligned}\Pr(w_k|w_1 \ldots w_{k-1}) &= \alpha \Pr(w_k|w_{k-n} \ldots w_{k-1}) \\ &+(1-\alpha)\Pr(w_k|w_1 \ldots w_{k-1}, G_s), \qquad (1)\end{aligned}$$

where $0 \leq \alpha \leq 1$ is a weight factor which depends on the task.

The expression $\Pr(w_k|w_{k-n} \ldots w_{k-1})$ is the word probability of occurrence of w_k given by the n-gram model. The parameters of this model can be easily estimated, and the expression $\Pr(w_k|w_{k-n} \ldots w_{k-1})$ can be efficiently computed (Bahl et al., 1983; Jelinek, 1998).

In order to define the stochastic grammatical model G_s of the expression $\Pr(w_k|w_1 \ldots w_{k-1}, G_s)$ for large-vocabulary complex tasks, we propose a combination of two different stochastic models: a category-based SCFG (G_c), that allows us to represent the long-term relations between these syntactical structures and a probabilistic model of word distribution into categories (C_w).

This proposal introduces two important aspects, which are the estimation of the parameters of the stochastic models, G_c and C_w, and the computation of the following expression:

$$\begin{aligned}&\Pr(w_k|w_1 \ldots w_{k-1}, G_c, C_w) \\ &= \frac{\Pr(w_1 \ldots w_k \ldots |G_c, C_w)}{\Pr(w_1 \ldots w_{k-1} \ldots |G_c, C_w)} \qquad (2)\end{aligned}$$

3 Training of the models

The parameters of the described model are estimated from a training sample, that is, from a set of sentences. Each word of the sentence has a part-of-speech tag (POStag) associated to it. These POStags are considered as word categories and are the terminal symbols of the SCFG. From this training sample, the parameters of G_c and C_w can be estimated as follows.

First, the parameters of C_w, represented by $\Pr(w|c)$, are computed as:

$$\Pr(w|c) = \frac{\mathrm{N}(w,c)}{\sum_{w'} \mathrm{N}(w',c)}, \qquad (3)$$

where $\mathrm{N}(w,c)$ is the number of times that the word w has been labeled with the POStag c. It is important to note that a word w can belong to different categories. In addition, it may happen that a word in a test set does not appear in the training set, and therefore some smoothing technique has to be carried out.

With regard to the estimation of the category-based SCFGs, one of the most widely-known methods is the Inside-Outside (IO) algorithm (Lari and Young, 1990). The application of this algorithm presents important problems which are accentuated in real tasks: the time complexity per iteration and the large number of iterations that are necessary to converge. An alternative to the IO algorithm is an algorithm based on the Viterbi score (VS algorithm) (Ney, 1992). The convergence of the VS algorithm is faster than the IO algorithm. However, the SCFGs obtained are, in general, not as well learned (Sánchez et al., 1996).

Another possibility for estimating SCFGs, which is somewhere between the IO and VS algorithms, has recently been proposed. This approach considers only a certain subset of derivations in the estimation process. In order to select this subset of derivations, two alternatives have been considered: from structural information content in a bracketed corpus (Pereira and Schabes, 1992; Amaya et al., 1999), and from statistical information content in the k-best derivations (Sánchez and Benedí, 1998). In the first alternative, the IOb and VSb algorithms which learn SCFGs from partially bracketed corpora were defined (Pereira and Schabes, 1992; Amaya et al., 1999). In the second alternative, the kVS algorithm for the estimation of the probability distributions of a SCFG from the k-best derivations was proposed (Sánchez and Benedí, 1998).

All of these algorithms have a time complexity $O(n^3|P|)$, where n is the length of the input string, and $|P|$ is the size of the SCFG.

These algorithms have been tested in real tasks for estimating category-based SCFGs (Sánchez and Benedí, 1999) and the results obtained justify their application in complex real tasks.

4 Integration of the model

From expression (2), it can bee seen that in order to integrate the model, it is necessary to efficiently compute the expression:

$$\Pr(w_1 \dots w_k \dots | G_c, C_w). \tag{4}$$

In order to describe how this computation can be made, we first introduce some notation.

A *Context-Free Grammar* G is a four-tuple (N, Σ, P, S), where N is the finite set of nonterminals, Σ is the finite set of terminals ($N \cap \Sigma = \emptyset$), $S \in N$ is the axiom or initial symbol and P is the finite set of productions or rules of the form $A \to \alpha$, where $A \in N$ and $\alpha \in (N \cup \Sigma)^+$ (only grammars with non empty rules are considered). For simplicity (but without loss of generality) only context-free grammars in *Chomsky Normal Form* are considered, that is, grammars with rules of the form $A \to BC$ or $A \to v$ where $A, B, C \in N$ and $v \in \Sigma$.

A *Stochastic Context-Free Grammar* G_s is a pair (G, p), where G is a context-free grammar and $p : P \to]0, 1]$ is a probability function of rule application such that $\forall A \in N$: $\sum_{\alpha \in (N \cup \Sigma)^+} p(A \to \alpha) = 1$.

Now, we present two algorithms in order to compute the word transition probability. The first algorithm is based on the LRI algorithm, and the second is based on an application of a Viterbi scheme to the LRI algorithm (the VLRI algorithm).

Probability of generating an initial substring

The computation of (4) is based on an algorithm which is a modification of the LRI algorithm (Jelinek and Lafferty, 1991). This new algorithm is based on the definition of $\Pr(A << i, j)) = \Pr(A \overset{*}{\Rightarrow} w_i \dots w_j \dots | G_c, C_w)$ as the probability that A generates the initial substring $w_i \dots w_j \dots$ given G_c and C_w. This can be computed with the following dynamic programming scheme:

$$\Pr(A << i, i) = \sum_c (p(A \to c) \Pr(w_i|c)$$
$$+ \sum_D Q(A \Rightarrow D) p(D \to c) \Pr(w_i|c)),$$
$$\Pr(A << i, j) = \sum_{B,C \in N} \sum_{l=i}^{j-1} Q(A \Rightarrow BC)$$
$$\Pr(B < i, l >) \Pr(C << l+1, j) \ .$$

In this way, $\Pr(w_1 \dots w_k \dots | G_c, C_w) = \Pr(S << 1, k)$.

In this expression, $Q(A \Rightarrow D)$ is the probability that D is the leftmost nonterminal in all sentential forms which are derived from A. The value $Q(A \Rightarrow BC)$ is the probability that BC is the initial substring of all sentential forms derived from A. $\Pr(B < i, l >)$ is the probability that the substring $w_i \dots w_l$ is generated from B given G_c and C_w. Its computation will be defined later.

It should be noted that the combination of the models G_c and C_w is carried out in the value $Pr(A << i, i)$. This is the main difference with respect the LRI algorithm.

Probability of the best derivation generating an initial substring

An algorithm which is similar to the previous one can be defined based on the Viterbi scheme. In this way, it is possible to obtain the best parsing of an initial substring. This new algorithm is also related to the VLRI algorithm (Sánchez and Benedí, 1997) and is based on the definition of $\widehat{\Pr}(A << i, j)) = \widehat{\Pr}(A \overset{*}{\Rightarrow} w_i \dots w_j \dots | G_c, C_w)$ as the probability of the most probable parsing which generates $w_i \dots w_j \dots$ from A given G_c and C_w. This can be computed as follows:

$$\widehat{\Pr}(A << i, i) = \max_c (p(A \to c) \Pr(w_i|c),$$
$$\max_D (\widehat{Q}(A \Rightarrow D) p(D \to c) \Pr(w_i|c))),$$
$$\widehat{\Pr}(A << i, j) = \max_{B,C \in N} \max_{l=i \dots j-1} (\widehat{Q}(A \Rightarrow BC)$$
$$\widehat{\Pr}(B < i, l >) \widehat{\Pr}(C << l+1, j)) \ .$$

Therefore $\widehat{\Pr}(w_1 \dots w_k \dots | G_c, C_w) = \widehat{\Pr}(S << 1, k)$.

In this expression, $\widehat{Q}(A \Rightarrow D)$ is the probability that D is the leftmost nonterminal in the most probable sentential form which is derived from A. The value $\widehat{Q}(A \Rightarrow BC)$ is the probability that BC is the initial substring of most the probable sentential form derived from A. $\widehat{\Pr}(B < i, l >)$ is the probability of the most probable parse which generates $w_i \ldots w_l$ from B.

Probability of generating a string

The value $\Pr(A < i,j >) = \Pr(A \overset{*}{\Rightarrow} w_i \ldots w_j | G_c, C_w)$ is defined as the probability that the substring $w_i \ldots w_j$ is generated from A given G_c and C_w. To calculate this probability a modification of the well-known *Inside* algorithm (Lari and Young, 1990) is proposed. This computation is carried out by using the following dynamic programming scheme:

$$
\begin{aligned}
\Pr(A < i,i >) &= \sum_c p(A \to c) \Pr(w_i|c) \ , \\
\Pr(A < i,j >) &= \sum_{B,C \in N} \sum_{l=i}^{j-1} p(A \to BC) \\
&\qquad \Pr(B < i,l >) \Pr(C < l+1, j >).
\end{aligned}
$$

In this way, $\Pr(w_1 \ldots w_n | G_c, C_w) = \Pr(S < 1, n >)$.

As we have commented above, the combination of the two parts of the grammatical model is carried out in the value $\Pr(A < i,i >)$.

Probability of the best derivation generating a string

The probability of the best derivation that generates a string, $\widehat{\Pr}(w_1 \ldots w_n | G_c, C_w)$, can be evaluated using a Viterbi-like scheme (Ney, 1992). As in the previous case, the computation of this probability is based on the definition of $\widehat{\Pr}(A < i,j >) = \widehat{\Pr}(A \overset{*}{\Rightarrow} w_i \ldots w_j | G_c, C_w)$ as the probability of the best derivation that generates the substring $w_i \ldots w_j$ from A given G_c and C_w. Similarly:

$$
\begin{aligned}
\widehat{\Pr}(A < i,i >) &= \max_c \ p(A \to C) \Pr(w_i|c) \ , \\
\widehat{\Pr}(A < i,j >) &= \max_{B,C \in N} \max_{l=i \ldots j-1} p(A \to BC) \\
&\qquad \widehat{\Pr}(B < i,l >) \widehat{\Pr}(C < l+1, j >) \ .
\end{aligned}
$$

Therefore, $\widehat{\Pr}(w_1 \ldots w_n | G_c, C_w) = \widehat{\Pr}(S < 1, n >)$.

Finally, the time complexity of these algorithms is the same as the algorithms they are related to, therefore the time complexity is $O(k^3|P|)$, where k is the length of the input string and $|P|$ is the size of the SCFG.

5 Experiments with the Penn Treebank Corpus

The corpus used in the experiments was the part of the Wall Street Journal which had been processed in the Penn Treebank project[1] (Marcus et al., 1993). This corpus consists of English texts collected from the Wall Street Journal from editions of the late eighties. It contains approximately one million words. This corpus was automatically labelled, analyzed and manually checked as described in (Marcus et al., 1993). There are two kinds of labelling: a POStag labelling and a syntactic labelling. The size of the vocabulary is greater than 25,000 different words, the POStag vocabulary is composed of 45 labels[2] and the syntactic vocabulary is composed of 14 labels.

The corpus was divided into sentences according to the bracketing. In this way, we obtained a corpus whose main characteristics are shown in Table 1.

Table 1: Characteristics of the Penn Treebank corpus once it was divided into sentences.

No. of senten.	Av. length	Std. deviation	Min. length	Max. length
49,207	23.61	11.13	1	249

We took advantage of the category-based SCFGs estimated in a previous work (Sánchez and Benedí, 1998). These SCFGs were estimated with sentences which had less than 15 words. Therefore, in this work, we assumed such restriction. The vocabulary size of the new corpus was 6,333 different words. For the experiments, the corpus was divided into a training corpus (directories 00 to 19) and a test corpus (directories 20 to 24). The characteristics of these sets can be seen in Table 2. The part of the

[1] Release 2 of this data set can be obtained from the Linguistic Data Consortium with Catalogue number LDC94T4B (http://www.ldc.upenn.edu/ldc/noframe.html)

[2] There are 48 labels defined in (Marcus et al., 1993), however, three of them do not appear in the corpus.

corpus labeled with POStags was used to estimate the parameters of the grammatical model, while the non-labeled part was used to estimate the parameters of the n-gram model. We now describe the estimation process in detail.

Table 2: Characteristics of the data sets defined for the experiments when the sentences with more than 15 POStags were removed.

Data set	No. of senten.	Av. length	Std. deviation
Training	9,933	10.67	3.46
Test	2,295	10.51	3.55

The parameters of a 3-gram model were estimated with the software tool described in (Rosenfeld, 1995) [3]. We used the linear interpolation smooth technique supported by this tool. The out-of-vocabulary words were grouped in the same class and were used in the computation of the perplexity. The test set perplexity with this model was 180.4.

The values of expression (3) were computed from the tagged and non-tagged part of the training corpus. In order to avoid null values, the unseen events were labeled with a special symbol w' which did not appear in the vocabulary, in such a way that $\Pr(w'|c) \neq 0$, $\forall c \in C$, where C was the set of categories. That is, all the categories could generate the unseen event. This probability took a very small value (several orders of magnitude less than $\min_{w \in V, c \in C} \Pr(w|c)$, where V was the vocabulary of the training corpus), and different values of this probability did not change the results.

The parameters of an initial ergodic SCFG were estimated with each one of the estimation methods mentioned in Section 3. This SCFG had $3,374$ rules, composed from 45 terminal symbols (the number of POStags) and 14 nonterminal symbols (the number of syntactic labels). The probabilities were randomly generated and three different seeds were tested, but only one of them is reported given that the results were very similar. The training corpus was the labeled part of the described corpus. The perplexity of the labeled part of the test set for different estimation algorithms can be seen in Table 3.

Table 3: Perplexity of the labeled part of the test set with the SCFG estimated with the methods mentioned in Section 3.

VS	*k*VS	IOb	VSb
21.56	20.65	13.14	21.84

Once we had estimated the parameters of the defined model, we applied expression (1) by using the LRI algorithm and the VLRI algorithm in expression (4). The test set perplexity that was obtained in function of α for different estimation algorithms (VS, *k*VS, IOb and VSb) can be seen in Fig. 1.

In the best case, the proposed language model obtained more than a 30% improvement over results obtained by the 3-gram language model (see Table 4). This result was obtained when the SCFG estimated with the IOb algorithm was used. The SCFGs estimated with other algorithms also obtained important improvements compared to the 3-gram. In addition, it can be observed that both the LRI algorithm and the VLRI algorithm obtained good results.

Table 4: Best test perplexity for different SCFG estimation algorithms, and the percentage of improvement with respect to the 3-gram model.

	VS	*k*VS	IOb	VSb
LRI	133.6	130.3	124.6	136.3
% improv.	25.9%	27.8%	30.9%	24.5%
VLRI	143.4	137.2	132.4	149.7
% improv.	20.5%	23.0%	26.6%	17.0%

An important aspect to note is that the weight of the grammatical part was approximately 50%, which means that this part provided important information to the language model.

6 Conclusions

A new language model has been introduced. This new language model is defined as a linear combination of an n-gram which represents relations between words, and a stochastic

[3] Release 2.04 is available at http://svr-www.eng.cam.ac.uk/∼ prc14/toolkit.html.

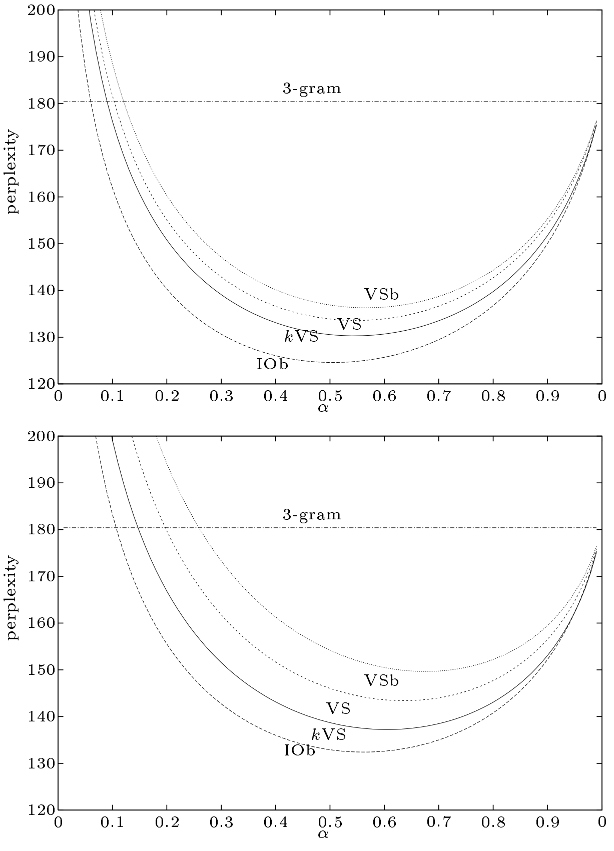

Figure 1: Test set perplexity obtained with the proposed language models in function of gamma. Different curves correspond to SCFGs estimated with different algorithms. The upper graphic corresponds to the results obtained when the LRI algorithm was used in the language models, and the lower graphic corresponds to the results obtained with the VLRI algorithm.

grammatical model which is used to represent the global relation between syntactic structures. The stochastic grammatical model is composed of a category-based SCFG and a probabilistic model of word distribution in the categories. Several algorithms have been described to estimate the parameters of the model from a the sample. In addition, efficient algorithms for solving the problem of the interpretation with this model have been presented.

The proposed model has been tested on the part of Wall Street Journal processed in the Penn Treebank project, and the results obtained improved by more than 30% the test set perplexity over results obtained by a simple 3-gram model.

References

F. Amaya, J.M. Benedí, and J.A. Sánchez. 1999. Learning of stochastic context-free grammars from bracketed corpora by means of reestimation algorithms. In M.I. Torres and A. Sanfeliu, editors, *Proc. VIII Spanish Symposium on Pattern Recognition and Image Analysis*, pages 119–126, Bilbao, España, May. AERFAI.

L.R. Bahl, F. Jelinek, and R.L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190.

J.R. Bellegarda. 1998. A multispan language modeling framework for large vocabulary speech recognition. *IEEE Trans. Speech and Audio Processing*, 6(5):456–476.

C. Chelba and F. Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proc. COLING*, Montreal, Canada. University of Montreal.

J. Gilet and W. Ward. 1998. A language model combining trigrams and stochastic context-free grammars. In *In 5th International Conference on Spoken Language Processing*, pages 2319–2322, Sidney, Australia.

F. Jelinek and J.D. Lafferty. 1991. Computation of the probability of initial substring generation by stochastic context-free grammars. *Computational Linguistics*, 17(3):315–323.

F. Jelinek. 1998. *Statistical Methods for Speech Recognition.* MIT Press.

K. Lari and S.J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer, Speech and Language*, 4:35–56.

M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330.

H. Ney. 1992. Stochastic grammars and pattern recognition. In P. Laface and R. De Mori, editors, *Speech Recognition and Understanding. Recent Advances*, pages 319–344. Springer-Verlag.

F. Pereira and Y. Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135. University of Delaware.

R. Rosenfeld. 1995. The cmu statistical language modeling toolkit and its use in the 1994 arpa csr evaluation. In *ARPA Spoken Language Technology Workshop*, Austin, Texas, USA.

M. Siu and M. Ostendorf. 2000. Variable n-grams and extensions for conversational speech language modeling. *IEEE Trans. on Speech and Audio Processing*, 8(1):63–75.

J.A. Sánchez and J.M. Benedí. 1997. Computation of the probability of the best derivation of an initial substring from a stochastic context-free grammar. In A. Sanfeliu, J.J. Villanueva, and J. Vitrià, editors, *Proc. VII Spanish Symposium on Pattern Recognition and Image Analysis*, pages 181–186, Barcelona, España, April. AERFAI.

J.A. Sánchez and J.M. Benedí. 1998. Estimation of the probability distributions of stochastic context-free grammars from the *k*-best derivations. In *In 5th International Conference on Spoken Language Processing*, pages 2495–2498, Sidney, Australia.

J.A. Sánchez and J.M. Benedí. 1999. Learning of stochastic context-free grammars by means of estimation algorithms. In *Proc. EUROSPEECH'99*, volume 4, pages 1799–1802, Budapest, Hungary.

J.A. Sánchez, J.M. Benedí, and F. Casacuberta. 1996. Comparison between the inside-outside algorithm and the viterbi algorithm for stochastic context-free grammars. In P. Perner, P. Wang, and A. Rosenfeld, editors, *Advances in Structural and Syntactical Pattern Recognition*, pages 50–59. Springer-Verlag.