Reversible Stochastic Attribute-Value Grammars

Daniël de Kok
University of Groningen
d.j.a.de.kok@rug.nl

Barbara Plank
University of Groningen
b.plank@rug.nl

Gertjan van Noord
University of Groningen
g.j.m.van.noord@rug.nl

Abstract

An attractive property of attribute-value grammars is their reversibility. Attribute-value grammars are usually coupled with separate statistical components for parse selection and fluency ranking. We propose reversible stochastic attribute-value grammars, in which a *single* statistical model is employed both for parse selection and fluency ranking.

1 Introduction

Reversible grammars were introduced as early as 1975 by Martin Kay (1975). In the eighties, the popularity of attribute-value grammars (AVG) was in part motivated by their inherent reversible nature. Later, AVG were enriched with a statistical component (Abney, 1997): stochastic AVG (SAVG). Training a SAVG is feasible if a stochastic model is assumed which is conditioned on the input sentences (Johnson et al., 1999). Various parsers based on this approach now exist for various languages (Toutanova et al., 2002; Riezler et al., 2002; van Noord and Malouf, 2005; Miyao and Tsujii, 2005; Clark and Curran, 2004; Forst, 2007). SAVG can be applied for generation to select the most fluent realization from the set of possible realizations (Velldal et al., 2004). In this case, the stochastic model is conditioned on the input logical forms. Such generators exist for various languages as well (Velldal and Oepen, 2006; Nakanishi and Miyao, 2005; Cahill et al., 2007; de Kok and van Noord, 2010).

If an AVG is applied both to parsing and generation, two distinct stochastic components are required, one for parsing, and one for generation. To some extent this is reasonable, because some features are only relevant in a certain direction. For instance, features that represent aspects of the surface word order are important for generation, but irrelevant for parsing. Similarly, features which describe aspects of the logical form are important for parsing, but irrelevant for generation. Yet, there are also many features that are relevant in both directions. For instance, for Dutch, a very effective feature signals a direct object NP in fronted position in main clauses. If a main clause is parsed which starts with a NP, the disambiguation component will favor a subject reading of that NP. In generation, the fluency component will favor subject fronting over object fronting. Clearly, such shared preferences are not accidental.

In this paper we propose reversible SAVG in which a *single* stochastic component is applied both in parsing and generation. We provide experimental evidence that such reversible SAVG achieve similar performance as their directional counterparts. A single, reversible model is to be preferred over two distinct models because it explains why preferences in a disambiguation component and a fluency component, such as the preference for subject fronting over object fronting, are shared. A single, reversible model is furthermore of practical interest for its simplicity, compactness, and maintainability. As an important additional advantage, reversible models are applicable for tasks which combine aspects of parsing and generation, such as word-graph parsing and paraphrasing. In situations where only a small amount of training data is available for parsing or generation, *cross-pollination* improves the perfor-

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 194–199,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

mance of a model. If preferences are shared between parsing and generation, it follows that a generator could benefit from parsing data and vice versa. We present experimental results indicating that in such a bootstrap scenario a reversible model achieves better performance.

2 Reversible SAVG

As Abney (1997) shows, we cannot use relatively simple techniques such as relative frequencies to obtain a model for estimating derivation probabilities in attribute-value grammars. As an alternative, he proposes a maximum entropy model, where the probability of a derivation d is defined as:

$$p(d) = \frac{1}{Z} exp \sum_i \lambda_i f_i(d) \tag{1}$$

$f_i(d)$ is the frequency of feature f_i in derivation d. A weight λ_i is associated with each feature f_i. In (1), Z is a normalizer which is defined as follows, where Ω is the set of derivations defined by the grammar:

$$Z = \sum_{d' \in \Omega} exp \sum_i \lambda_i f_i(d') \tag{2}$$

Training this model requires access to *all* derivations Ω allowed by the grammar, which makes it hard to implement the model in practice.

Johnson et al. (1999) alleviate this problem by proposing a model which conditions on the input sentence s: $p(d|s)$. Since the number of derivations for a given sentence s is usually finite, the calculation of the normalizer is much more practical. Conversely, in generation the model is conditioned on the input logical form l, $p(d|l)$ (Velldal et al., 2004). In such directional stochastic attribute-value grammars, the probability of a derivation d given an input x (a sentence or a logical form) is defined as:

$$p(d|x) = \frac{1}{Z(x)} exp \sum_i \lambda_i f_i(x,d) \tag{3}$$

with $Z(x)$ as ($\Omega(x)$ are all derivations for input x):

$$Z(x) = \sum_{d' \in \Omega(x)} exp \sum_i \lambda_i f_i(x,d') \tag{4}$$

Consequently, the constraint put on feature values during training only refers to derivations with the same input. If X is the set of inputs (for parsing, all sentences in the treebank; for generation, all logical forms), then we have:

$$E_p(f_i) - E_{\tilde{p}}(f_i) = 0 \equiv \tag{5}$$
$$\sum_{x \in X} \sum_{d \in \Omega(x)} \tilde{p}(x)p(d|x)f_i(x,d) - \tilde{p}(x,d)f_i(x,d) = 0$$

Here we assume a uniform distribution for $\tilde{p}(x)$. Let $j(d)$ be a function which returns 0 if the derivation d is inconsistent with the treebank, and 1 in case the derivation is correct. $\tilde{p}(x,d)$ is now defined in such a way that it is 0 for incorrect derivations, and uniform for correct derivations for a given input:

$$\tilde{p}(x,d) = \tilde{p}(x) \frac{j(d)}{\Sigma_{d' \in \Omega(x)} j(d')} \tag{6}$$

Directional SAVG make parsing and generation practically feasible, but require separate models for parse disambiguation and fluency ranking.

Since parsing and generation both create derivations that are in agreement with the constraints implied by the input, a single model can accompany the attribute-value grammar. Such a model estimates the probability of a derivation d given a set of constraints c, $p(d|c)$. We use conditional maximum entropy models to estimate $p(d|c)$:

$$p(d|c) = \frac{1}{Z(c)} exp \sum_i \lambda_i f_i(c,d) \tag{7}$$

$$Z(c) = \sum_{d' \in \Omega(c)} exp \sum_i \lambda_i f_i(c,d') \tag{8}$$

We derive a reversible model by training on data for parse disambiguation and fluency ranking simultaneously. In contrast to directional models, we impose the two constraints per feature given in figure 1: one on the feature value with respect to the sentences S in the parse disambiguation treebank and the other on the feature value with respect to logical forms L in the fluency ranking treebank. As a result of the constraints on training defined in figure 1, the feature weights in the reversible model distinguish, at the same time, good parses from bad parses as well as good realizations from bad realizations.

3 Experimental setup and evaluation

To evaluate reversible SAVG, we conduct experiments in the context of the Alpino system for Dutch.

$$\sum_{s \in S} \sum_{d \in \Omega(s)} \tilde{p}(s) p(d|c=s) f_i(s,d) - \tilde{p}(c=s,d) f_i(s,d) = 0$$

$$\sum_{l \in L} \sum_{d \in \Omega(l)} \tilde{p}(l) p(d|c=l) f_i(l,d) - \tilde{p}(c=l,d) f_i(l,d) = 0$$

Figure 1: Constraints imposed on feature values for training reversible models $p(d|c)$.

Alpino provides a wide-coverage grammar, lexicon and parser (van Noord, 2006). Recently, a sentence realizer has been added that uses the same grammar and lexicon (de Kok and van Noord, 2010).

In the experiments, the cdbl part of the Alpino Treebank (van der Beek et al., 2002) is used as training data (7,154 sentences). The WR-P-P-H part (2,267 sentences) of the LASSY corpus (van Noord et al., 2010), which consists of text from the Trouw 2001 newspaper, is used for testing.

3.1 Features

The features that we use in the experiment are the same features which are available in the Alpino parser and generator. In the following section, these features are described in some detail.

Word adjacency. Two word adjacency features are used as auxiliary distributions (Johnson and Riezler, 2000). The first feature is the probability of the sentence according to a word trigram model. The second feature is the probability of the sentence according to a tag trigram model that uses the part-of-speech tags assigned by the Alpino system. In both models, linear interpolation smoothing for unknown trigrams, and Laplacian smoothing for unknown words and tags is applied. The trigram models have been trained on the Twente Nieuws Corpus corpus (approximately 110 million words), excluding the Trouw 2001 corpus. In conventional parsing tasks, the value of the word trigram model is the same for all derivations of a given input sentence.

Lexical frames. Lexical analysis is applied during parsing to find all possible subcategorization frames for the tokens in the input sentence. Since some frames occur more frequently in good parses than others, we use feature templates that record the frames that were used in a parse. An example of such a feature is: "'to play' serves as an intransitive verb". We also use an auxiliary distribution of word and frame combinations that was trained on a large corpus of automatically annotated sentences (436 million words). The values of lexical frame features are constant for all derivations in sentence realization, unless the frame is not specified in the logical form.

Dependency relations. There are also feature templates which describe aspects of the dependency structure. For each dependency, three types of dependency features are extracted. Examples of such features are "a pronoun is used as the subject of a verb", "the pronoun 'she' is used as the subject of a verb", "the noun 'beer' is used as the object of the verb 'drink'". In addition, features are used which implement auxiliary distributions for selectional preferences, as described in Van Noord (2007). In conventional realization tasks, the values of these features are constant for all derivations for a given input representation.

Syntactic features. Syntactic features include features which record the application of each grammar rule, as well as features which record the application of a rule in the context of another rule. An example of the latter is 'rule 167 is used to construct the second daughter of a derivation constructed by rule 233'. In addition, there are features describing more complex syntactic patterns such as: fronting of subjects and other noun phrases, orderings in the middle field, long-distance dependencies, and parallelism of conjuncts in coordination.

3.2 Parse disambiguation

Earlier we assumed that a treebank is a set of correct derivations. In practice, however, a treebank only contains an abstraction of such derivations (in our case sentences with corresponding dependency structures), thus abstracting away from syntactic details needed in a parse disambiguation model. As in Osborne (2000), the derivations for the parse disambiguation model are created by parsing the training corpus. In the current setting, up to at most 3000 derivations are created for every sentence. These derivations are then compared to the gold standard dependency structure to judge the quality of the parses. For a given sentence, the parses with the highest concept accuracy (van Noord, 2006) are considered correct, the rest is treated as incorrect.

3.3 Fluency ranking

For fluency ranking we also need access to full derivations. To ensure that the system is able to generate from the dependency structures in the treebank, we parse the corresponding sentence, and select the parse with the dependency structure that corresponds most closely to the dependency structure in the treebank. The resulting dependency structures are fed into the Alpino chart generator to construct derivations for each dependency structure. The derivations for which the corresponding sentences are closest to the original sentence in the treebank are marked correct. Due to a limit on generation time, some longer sentences and corresponding dependency structures were excluded from the data. As a result, the average sentence length was 15.7 tokens, with a maximum of 26 tokens. To compare a realization to the correct sentence, we use the General Text Matcher (GTM) method (Melamed et al., 2003; Cahill, 2009).

3.4 Training the models

Models are trained by taking an informative sample of $\Omega(c)$ for each c in the training data (Osborne, 2000). This sample consists of at most 100 randomly selected derivations. Frequency-based feature selection is applied (Ratnaparkhi, 1999). A feature *f partitions* $\Omega(c)$, if there are derivations d and d' in $\Omega(c)$ such that $f(c,d) \neq f(c,d')$. A feature is used if it partitions the informative sample of $\Omega(c)$ for at least two c. Table 1 lists the resulting characteristics of the training data for each model.

	Features	Inputs	Derivations
Generation	1727	3688	141808
Parse	25299	7133	376420
Reversible	25578	10811	518228

Table 1: Size of the training data for each model

We estimate the parameters of the conditional maximum entropy models using TinyEst,[1] with a Gaussian (ℓ_2) prior distribution ($\mu = 0$, $\sigma^2 = 1000$) to reduce overfitting (Chen and Rosenfeld, 1999).

4 Results

4.1 Parse disambiguation

Table 2 shows the results for parse disambiguation. The table also provides lower and upper bounds: the baseline model selects an arbitrary parse per sentence; the oracle chooses the best available parse. Figure 2 shows the learning curves for the directional parsing model and the reversible model.

Model	**CA (%)**	**f-score (%)**
Baseline	75.88	76.28
Oracle	94.86	95.09
Parse model	90.93	91.28
Reversible	90.87	91.21

Table 2: Concept Accuracy scores and f-scores in terms of named dependency relations for the parsing-specific model versus the reversible model.

The results show that the general, reversible, model comes very close to the accuracy obtained by the dedicated, parsing specific, model. Indeed, the tiny difference is not statistically significant. We compute statistical significance using the *Approximate Randomization Test* (Noreen, 1989).

4.2 Fluency ranking

Table 3 compares the reversible model with a directional fluency ranking model. Figure 3 shows the learning curves for the directional generation model and the reversible model. The reversible model achieves similar performance as the directional model (the difference is not significant).

To show that a reversible model can actually profit from mutually shared features, we report on an experiment where only a small amount of generation

[1] `http://github.com/danieldk/tinyest`

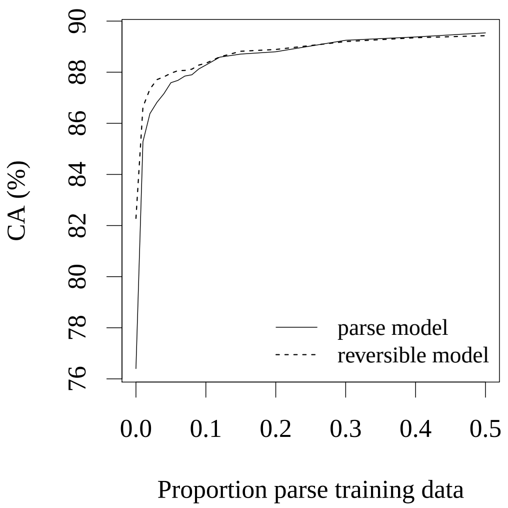

Figure 2: Learning curve for directional and reversible models for parsing. The reversible model uses all training data for generation.

Model	GTM
Random	55.72
Oracle	86.63
Fluency	71.82
Reversible	71.69

Table 3: General Text Matcher scores for fluency ranking using various models.

training data is available. In this experiment, we manually annotated 234 dependency structures from the cdbl part of the Alpino Treebank, by adding correct realizations. In many instances, there is more than one fluent realization. We then used this data to train a directional fluency ranking model and a reversible model. The results for this experiment are shown in Table 4. Since the reversible model outperforms the directional model we conclude that indeed fluency ranking benefits from parse disambiguation data.

Model	GTM
Fluency	70.54
Reversible	71.20

Table 4: Fluency ranking using a small amount of annotated fluency ranking training data (difference is significant at $p < 0.05$).

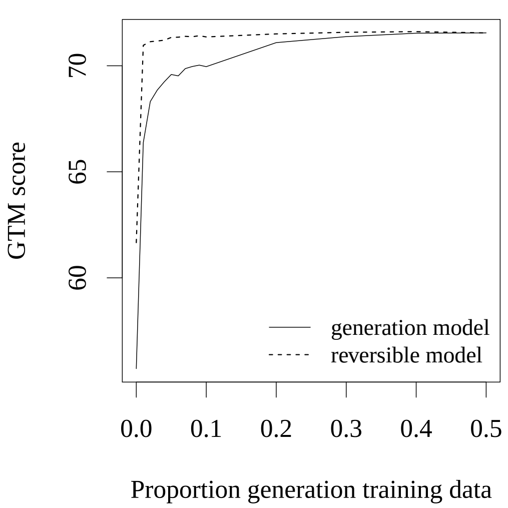

Figure 3: Learning curves for directional and reversible models for generation. The reversible models uses all training data for parsing.

5 Conclusion

We proposed reversible SAVG as an alternative to directional SAVG, based on the observation that syntactic preferences are shared between parse disambiguation and fluency ranking. This framework is not purely of theoretical interest, since the experiments show that reversible models achieve accuracies that are similar to those of directional models. Moreover, we showed that a fluency ranking model trained on a small data set can be improved by complementing it with parse disambiguation data.

The integration of knowledge from parse disambiguation and fluency ranking could be beneficial for tasks which combine aspects of parsing and generation, such as word-graph parsing or paraphrasing.

References

Steven Abney. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23(4):597–618.

Aoife Cahill, Martin Forst, and Christian Rohrer. 2007. Stochastic realisation ranking for a free word order language. In *ENLG '07: Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 17–24, Morristown, NJ, USA.

Aoife Cahill. 2009. Correlating human and automatic evaluation of a german surface realiser. In *Proceedings of the ACL-IJCNLP 2009 Conference - Short Papers*, pages 97–100.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University, Pittsburg.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting of the ACL*, pages 103–110, Morristown, NJ, USA.

Daniël de Kok and Gertjan van Noord. 2010. A sentence generator for Dutch. In *Proceedings of the 20th Computational Linguistics in the Netherlands conference (CLIN)*.

Martin Forst. 2007. Filling statistics with linguistics: property design for the disambiguation of german lfg parses. In *DeepLP '07: Proceedings of the Workshop on Deep Linguistic Processing*, pages 17–24, Morristown, NJ, USA.

Mark Johnson and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the 1st Meeting of the NAACL*, pages 154–161, Seattle, Washington.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the ACL*.

Martin Kay. 1975. Syntactic processing and functional sentence perspective. In *TINLAP '75: Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, pages 12–15, Morristown, NJ, USA.

I. Dan Melamed, Ryan Green, and Joseph Turian. 2003. Precision and recall of machine translation. In *HLT-NAACL*.

Yusuke Miyao and Jun'ichi Tsujii. 2005. Probabilistic disambiguation models for wide-coverage hpsg parsing. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 83–90, Morristown, NJ, USA.

Hiroko Nakanishi and Yusuke Miyao. 2005. Probabilistic models for disambiguation of an hpsg-based chart generator. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*, pages 93–102.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience.

Miles Osborne. 2000. Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the 18th conference on Computational linguistics (COLING)*, pages 586–592.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1):151–175.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 271–278, Morristown, NJ, USA.

Kristina Toutanova, Christopher D. Manning, Stuart M. Shieber, Dan Flickinger, and Stephan Oepen. 2002. Parse disambiguation for a rich hpsg grammar. In *First Workshop on Treebanks and Linguistic Theories (TLT)*, pages 253–263, Sozopol.

Leonoor van der Beek, Gosse Bouma, Robert Malouf, and Gertjan van Noord. 2002. The Alpino dependency treebank. In *Computational Linguistics in the Netherlands (CLIN)*.

Gertjan van Noord and Robert Malouf. 2005. Wide coverage parsing with stochastic attribute value grammars. Draft available from the authors. A preliminary version of this paper was published in the Proceedings of the IJCNLP workshop Beyond Shallow Analyses, Hainan China, 2004.

Gertjan van Noord, Ineke Schuurman, and Gosse Bouma. 2010. Lassy syntactische annotatie, revision 19053.

Gertjan van Noord. 2006. **A**t **L**ast **P**arsing **I**s **N**ow **O**perational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven.

Gertjan van Noord. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the International Workshop on Parsing Technology (IWPT)*, ACL 2007 Workshop, pages 1–10, Prague.

Erik Velldal and Stephan Oepen. 2006. Statistical ranking in tactical generation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 517–525, Sydney, Australia, July. ACL.

Erik Velldal, Stephan Oepen, and Dan Flickinger. 2004. Paraphrasing treebanks for stochastic realization ranking. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories (TLT)*, pages 149–160.