Maximum Entropy Models for Realization Ranking

Erik Velldal[♣] **and Stephan Oepen**[♣♠]

[♣] Department of Linguistics and Scandinavian Studies, University of Oslo (Norway)
[♠] Center for the Study of Language and Information, Stanford (USA)

`erik.velldal@iln.uio.no` | `oe@csli.stanford.edu`

Abstract

In this paper we describe and evaluate different statistical models for the task of *realization ranking*, i.e. the problem of discriminating between competing surface realizations generated for a given input semantics. Three models are trained and tested; an n-gram language model, a discriminative maximum entropy model using structural features, and a combination of these two. Our realization component forms part of a larger, hybrid MT system.

1 Introduction

The topic of this paper is the use of statistical models for *realization ranking*, i.e. the problem of choosing among multiple paraphrases that are generated for a given meaning representation. The particular system considered in this project is the generator component of the Norwegian-to-English machine translation system LOGON (Oepen et al., 2004). While the core of the LOGON system follows a symbolic or rule-based approach, its deep linguistic analysis is augmented with statistical methods for ambiguity management. The focus of this paper is the isolated subproblem of ranking, and ultimately selecting, the final target realizations produced by the generator component.

Velldal, Oepen, & Flickinger (2004) introduced a notion of *symmetric treebanks* that can be used for training statistical models for realization ranking in a manner similar to earlier work on statistical models for parse selection. The utility of a small initial prototype of a symmetric treebank was tested by training models for realization ranking using a limited feature set. The preliminary results of Velldal et al. (2004) suggest that a discriminative model trained on tiny amounts of data can compete favorably on the realization ranking task when compared to a n-gram language model trained on a large text corpus. In the current paper we train and evaluate rankers on an expanded treebank and using a richer inventory of feature types. Three models are described: a traditional surface-oriented n-gram model, a maximum entropy model using structural features, and a combination of these two. We evaluate the different models, as well as the utility of individual feature types, by comparing exact match accuracy and averaged per-sentence BLEU scores (Papineni, Roukos, Ward, & Zhu, 2002).

The paper is organized as follows. In Section 2 we briefly review our notion of symmetric treebanks and the properties of the data set used for our experiments. Section 3 describes the three models that we train, including the feature types of the maximum entropy (MaxEnt) models. An evaluation of the performance of the different models is presented in Section 4, before we go on to discuss the results and sketch directions for ongoing work in Section 5.

2 Background and Data

The LOGON system has an architecture based on semantic transfer that uses meaning representations based on Minimal Recursion Semantics (MRS; Copestake, Flickinger, Malouf, Riehemann, & Sag, 1995). Operating from such input representations, the lexically-driven chart generator of the Linguistic Knowledge Builder system (LKB; Carroll, Copestake, Flickinger, & Poznanski, 1999) then generates target language realizations in accordance with the LinGO English Resource Grammar (ERG; Flickinger, 2002).[1] As is long established, there are usually many ways to express a given meaning in natural language, some more effective or natural-sounding than others. Table 1 shows some examples of alternate outputs when generating from a single (underspecified) input semantics using the ERG: while a linguistic precision grammar goes a long way towards guaran-

[1] Both the LKB and ERG, as well as large parts of the LOGON machine translation system itself are part of the open-source DELPH-IN repository; see '`http://www.delph-in.net/`' for background.

remember that dogs must be on a leash .
remember dogs must be on a leash .
on a leash remember that dogs must be .
on a leash remember dogs must be .
a leash remember that dogs must be on .
a leash remember dogs must be on .
dogs remember must be on a leash .

Table 1: Example sets of generator outputs using the LinGO ERG. Unless the input semantics is specified for aspects of information structure (e.g. requresting foregrounding of a specific entity), paraphrases will include all grammatically legitimate topicalizations. Other sources of generator ambiguity include, for example, the optionality of complementizers and relative pronouns, permutation of (intersective) modifiers, as well as lexical and orthographic alternations.

teeing grammaticality of all realizations (to the level of providing the so-called *that* filter on subject extraction, for example), clearly some outputs are far more fluent than others. For the ambiguous items in the test data that we consider in this paper we get close to 73 realizations on average, where the maximium is 5712 candidates for a single input MRS (this maximum, however, is specific to our data set and could well be larger). The number of per-item readings is expected to further increase as the coverage of the MT system as a whole is broadened and as the system is extended to generate from packed, ambiguous transfer outputs. It is therefore necessary to have a scalable method for selecting the final target realizations.

2.1 Symmetric Treebanks

This section briefly describes the data sets that we use for evaluating the different statistical rankers and also for training the MaxEnt models.

In order to select a preferred surface realization we want a conditional model that gives us the probability of a string given its semantics. It is worth noting that the problem of realization ranking in many ways can be seen as 'inversely similar' to the problem of *parse selection*, i.e. choosing the best analysis for a given string. Our work on constructing models for realization draws heavily on the previous work on parse disambiguation in relation to the HPSG Redwoods[2] treebank, as reported by Oepen et al. (2002).

Stochastic models for parse selection are typically trained on a treebank consisting of strings paired with their optimal analyses. When training the discriminative models (described in Section 3.2) for realization selection we use a treebank where this optimality relation is taken to be *bidirectional* in the sense that the original string is also treated as an optimal realization of the corresponding semantic analysis (i.e. 'meaning'). For each input, the Redwoods treebank provides a full HPSG analysis that also includes the semantic component. This means that we can use the semantics associated with each preferred analysis to generate all paraphrases for each item. Velldal et al. (2004) proposed a notion of *symmetric treebanks* defined as the combination of (a) a set of pairings of surface forms and associated semantics, combined with (b) the sets of alternative analyses for each surface form and (c) sets of alternate realizations of the semantics. The preferred or optimal realizations are automatically labeled by matching the *yields* of the generated trees against the original strings in the parse treebank.

Some core metrics of our experimental material are summarized in Table 2: the data set (dubbed Rondane) is comprised of a little over one thousand sentences of instructional, native-English text taken from on-line guides to tourism in Norway (the application domain of the LOGON machine translation system). The raw text, ERG parse trees, and hand-selected MRS meaning representations are part of the publicly available Redwoods treebank, and we used the re-generation and alignment technique sketched above to obtain a symmetric treebank for our purposes. For our realization ranking experiments, we excluded the items that had no or just a single, unambiguous generator output, arriving at a total of 864 $\langle$*meaning*, *surface*$\rangle$ pairs for training and evaluation. Table 2 also provides the baseline statistics for guessing the preferred realization by chance: using the same measure of exact match accuracy as applied in Section 4, the random choice baseline for the Rondane generation treebank is at just above 18%.

3 Models for Realization Ranking

In this section we describe the three different rankers that we apply for the task of choosing among the target sentences produced by the generator. The first model that we present is a traditional n-gram language model. We then

[2] See 'http://www.delph-in.net/redwoods/' for more information about the Redwoods project.

Aggregate	**items**	**words**	**readings**	**baseline**
	$\sharp$	ϕ	ϕ	ϕ
$100 \leq$ *readings*	87	20.5	580.8	0.42
$50 \leq$ *readings* < 100	61	17.3	73.0	1.44
$10 \leq$ *readings* < 50	269	15.1	22.5	5.61
$5 <$ *readings* < 10	172	11.1	6.9	15.66
$1 <$ *readings* < 5	275	8.8	2.8	40.9
Total	**864**	**13.0**	**72.9**	**18.03**

Table 2: Some core metrics for the symmetric treebank data used in our initial experiments, broken down by degrees of ambiguity in generation. The columns are, from left to right, the subdivision of the data according to the number of realizations, total number of items scored (excluding items with only one realization), average string length, and average structural ambiguity. The rightmost column shows a random choice baseline, i.e. the probability of selecting the preferred realization by chance.

go on to look at the two *maximum entropy* or *log-linear* models that we train using structural features from the symmetric treebank described in the previous section.

3.1 A Language Model Ranker

The first statistical model that we apply for ranking the generator outputs is an n-gram language model.[3] This approach is in many ways similar to those presented by, among others, Langkilde & Knight (1998) and White (2004) and quite generally still appears predominant in the realization ranking literature. The model is trained on an unannotated version of the British National Corpus (BNC), containing roughly 100 million words. As the realizations in our symmetric treebank also include punctuations, these are also treated as separate tokens by the language model (in addition to sentence boundary markers). We then rank the realizations by computing their negative log-probabilities with respect to the model. In other words, the score of a string with k tokens, $score(w_1^k)$, is computed as $-\ln p_n(w_1^k) = -\sum_{i=1}^{k} \ln p_n(w_i|w_{i-n}, \ldots, w_{i-1})$. After training and testing several language models for varying values of n, we ended up using an 4-gram model (backing-off for unobserved n-grams) for the results reported here.

3.2 A Maximum Entropy Ranker

Log-linear models provide a very flexible framework that has been widely used for a range of tasks in NLP, including parse selection (see e.g. Johnson, Geman, Canon, Chi, & Riezler, 1999; Malouf & Noord, 2004) and reranking for machine translation (see e.g. Och et al., 2004). A model is specified by a set of real-valued *feature functions* that describe properties of the data, and an associated set of *learned weights* that determine the contribution of each feature. Given a set of d such features, each realization r is represented as a feature vector $f(r) \in \Re^d$, and a vector of weights $\lambda \in \Re^d$ is then fitted to optimize the likelihood of the training data. A conditional log-linear model for the probability of a realization r given the semantics s, has the general parametric form

$$(1) \quad p_\lambda(r|s) = \frac{1}{Z_\lambda(s)}\, q(r|s) \exp\left(\sum_{i=1}^{d} \lambda_i f_i(r)\right)$$

where Z_λ is a normalization term defined as

$$(2) \quad Z_\lambda(s) = \sum_{r' \in Y(s)} q(r'|s) \exp\left(\sum_{i=1}^{d} \lambda_i f_i(r')\right)$$

and $Y(s)$ gives the set of all possible realizations of s. The so-called reference or default distribution q is often only implicit since in maximum entropy estimation this is just the constant function $\frac{1}{|Y(s)|}$ (for a given s). One can, however, also replace this uniform distribution by some other reference distribution to incorporate prior knowledge in the model. This approach is also known as *maximum entropy / minimum divergence* (MEMD) modeling, and we will return to this more general framework below.

The estimation[4] of the λ-parameters seek to maximize the (log of) a penalized likelihood

[3] When training the language models we used the freely available CMU-SLM Toolkit.

[4] We use the `estimate` open-source package (Malouf, 2002) for training, using its *limited-memory variable metric* as the optimization method.

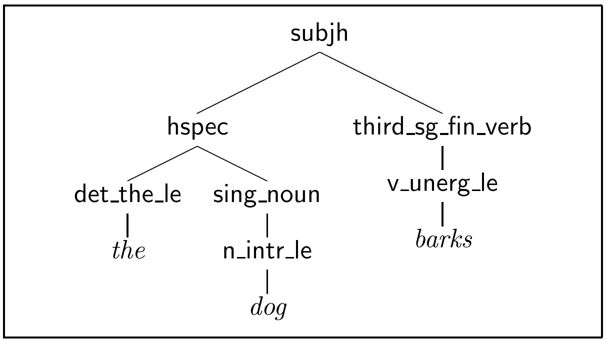

Figure 1: Sample HPSG derivation tree for the input *the dog barks*. Phrasal nodes are labeled with identifiers of grammar rules, and (pre-terminal) lexical nodes with class names for types of lexical entries.

#	sample features
1	⟨0 subjh hspec third_sg_fin_verb⟩
1	⟨1 △ subjh hspec third_sg_fin_verb⟩
1	⟨0 hspec det_the_le sing_noun⟩
1	⟨1 subjh hspec det_the_le sing_noun⟩
1	⟨2 △ subjh hspec det_the_le sing_noun⟩
2	⟨0 subjh third_sg_fin_verb⟩
2	⟨0 hspec sing_noun⟩
3	⟨1 n_intr_le *dog*⟩
3	⟨2 det_the_le n_intr_le *dog*⟩
3	⟨3 ◁ det_the_le n_intr_le *dog*⟩
4	⟨1 n_intr_le⟩
4	⟨2 det_the_le n_intr_le⟩
4	⟨3 ◁ det_the_le n_intr_le⟩

Table 3: Example structural features extracted from the derivation tree in Figure 1 The first column numbers the feature template corresponding to each example; in the examples, the first integer value is a parameter to feature templates, i.e. the depth of grandparenting (types 1 and 2) or n-gram size (types 3 and 4). The special symbols △ and ◁ denote the root of the tree and left periphery of the yield, respectively.

function as in

$$\text{(3)} \qquad \hat{\lambda} = \arg\max_{\lambda} \log L(\lambda) - \frac{\sum_{i=1}^{d} \lambda_i^2}{2\sigma^2}$$

where $L(\lambda)$ is the 'conditionalized' likelihood of the training data (as described by Johnson et al., 1999), computed as $L(\lambda) = \prod_{i=1}^{N} p_\lambda(r_i|s_i)$. The second term of the likelihood function in Equation (3) is a penalty term that is commonly used for reducing the tendency of log-linear models to over-fit, especially when training on sparse data using many features (Chen & Rosenfeld, 1999; Johnson et al., 1999; Malouf & Noord, 2004). More specifically it defines a zero-mean Gaussian prior on the feature weights which effectively leads to less extreme values. Note that, maximizing the likelihood of the training data is equivalent to minimizing the relative entropy (aka KL-diveregence) between the model and the reference distribution $\mathrm{D}(p_\lambda||q)$ on the one hand, and between the empirical distribution and the model $\mathrm{D}(\tilde{p}\,||p_\lambda)$ on the other.

Given a MaxEnt model p_λ, the scores used for ranking the candidate realizations can be computed simply as $score(r) = \sum_i \lambda_i f_i(r)$ since we are only interested in the rank order.

3.3 Maximum Entropy Features

The first MaxEnt model that we trained uses structural features defined over HPSG derivation trees as summarized in Table 3. For the purpose of parse selection, Toutanova, Manning, Shieber, Flickinger, & Oepen (2002) and Toutanova & Manning (2002) train a discriminative log-linear model on the Redwoods parse treebank, using features defined over *derivation trees* with non-terminals representing the *construction types* and *lexical types* of the HPSG grammar. The basic feature set of our MaxEnt realization ranker is defined in the same way (corresponding to the PCFG-S model of Toutanova & Manning, 2002), each feature capturing a sub-tree from the derivation limited to depth one. Table 3 shows example features in our MaxEnt models, where the feature template # 1 corresponds to local derivation sub-trees. To reduce the effects of data sparseness, feature type # 2 in Table 3 provides a back-off to derivation sub-trees, where the sequence of daughters is reduced to just the head daughter. Conversely, to facilitate sampling of larger contexts than just sub-trees of depth one, feature template # 1 allows optional grandparenting, including the upwards chain of dominating nodes in some features. In our experiments, we found that grandparenting of up to two dominating nodes gave the best balance of enlarged context vs. data sparseness.

In addition to these dominance-oriented features taken from the derivation trees of each realization, our models also include more surface-oriented features, viz. n-grams of lexical types with or without lexicalization. Feature type # 3 in Table 3 defines n-grams of variable size, where (in a loose analogy to part of speech tagging) sequences of lexical types capture syntactic category assignments. Feature templates # 3 and # 4 only differ with regard to lexicalization, as the former includes the surface token associated with the rightmost element of each n-gram (loosely corresponding to the emission probabilities in an HMM tagger). Unless otherwise noted, we used a maximum n-gram size of two in the experiments reported here, again due to its empirically determined best overall performance. When instantiating all feature templates as described above our models contain close to 65000 features.

3.4 A Combined Model

The second MaxEnt model is a combination of the two models described in Section 3.1 and 3.2 above; in addition to the set of structural feature types it includes as a separate feature the sentence scores computed by the n-gram language model. In other words, the value of the $d + 1$'th feature is the log-probability of the string as given by the n-gram model p_n, i.e. $f_{d+1}(r) = \ln\ p_n(y(r))$, where $y(r)$ is the yield of r and $n = 4$ as before.

Johnson & Riezler (2000) show an interesting equivalence between using log-probabilities as features and using a geometric mixture of the same probabilities for the default distribution q of Equation 1 (where the λ-parameters of the features would correspond to their weights in the mixture). This means that a special case of the simple combined model we present here would be a MEMD model where the uniform distribution q is replaced by the language model p_n. If $\lambda_{d+1} = 1$ then $\exp(f_{d+1}\lambda_{d+1}) = p_n$ and we would effectively have a MEMD model as described above with $q = p_n$.

Both of the log-linear models described in this section were trained and tested through 10-fold cross validation on the Rondane data set summarized above, and we empirically determined a suitable value for σ^2 (the variance parameter in the prior of Equation 3) which is here set to 10000 and 1000 for the models with and without the LM-feature respectively.

4 Evaluation

We here present an evaluation of the different models based on exact match accuracy and the BLEU string similarity metric (Papineni et al., 2002). The exact match measure simply counts the number of times that the model assigns the highest probability to a string that exactly matches the corresponding 'gold' or reference sentence (i.e. a sentence that is marked as preferred in the symmetric treebank). This score is discounted appropriately if several realizations are given the top rank by the model. Although the simple measure of exact match accuracy offers a very intuitive and transparent view, it is also in some respects too harsh as an evaluation measure in our setting. Since often more than one of the candidate realizations will be a suitable rendering of the input semantics, it would seem unfair (and potentially even uninformative) to only give credit in the case of an exact match, on an all or nothing basis. In addition to the exact match scores we therefore also include the BLEU metric, which is here used as a string similarity measure.

The well-established BLEU measure computes a weighted average of the n-gram precision of the selected realization with respect to the reference. The precision is computed for all $1 \leq n \leq N$, with $N = 4$, and the final score has a constant range in $[0, 1]$. Note that we here report averaged *sentence-level* BLEU scores. Furthermore, the particular implementation of BLEU in use in the LOGON system contains modifications that are similar in spirit to those found in NEVA as defined by Forsbom (2003). In order to make the measure well defined for cases where the length c of the candidate sentence is less than N or when there are no matching n-grams of size N, we set N to be the highest number $n \leq N_{max}$ for which a match exists (if any), where $N_{max} = \min(c, 4)$.

4.1 Experimental Results

The baseline accuracy that would be expected if we were to randomly pick a candidate sentence for each item is 18.03%. All of the three models we have tested outperform this random choice baseline by a good margin with respect to exact match accuracy; the n-gram model achieves 48.46%, the MaxEnt model 61.58%, while the combined model performs better than any of its component models and checks out with an exact match accuracy of 65.63%. We see the same relative ordering with respect to the similarity-based metric: the structural MaxEnt model outperforms the surface-oriented language model, while the combined model in turn outperforms both of these. Figures 2 and 3 show the accuracy and BLEU scores, respectively, where the data items are aggregated into bins according to the level of ambiguity, i.e. number of paraphrases. In all aggregates we find that the relative order of rankers is the same for the two performance measures, although BLEU appears to be somewhat more forgiving with the n-gram model. As expected we also see a degradation

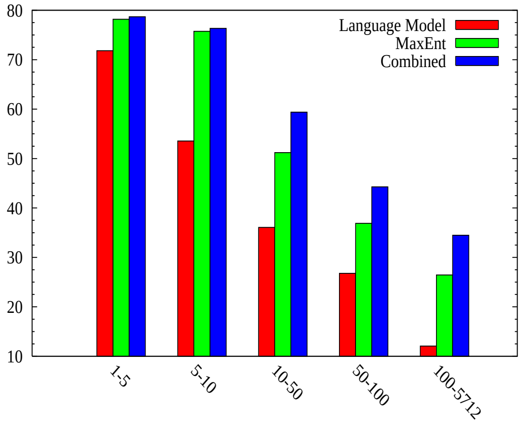

Figure 2: Exact match accuracy scores for the different models binned with respect to number of distinct realizations.

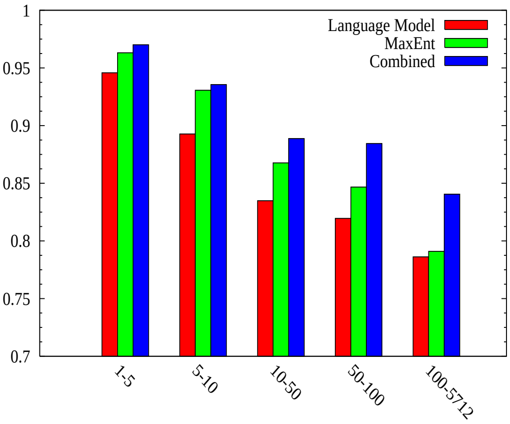

Figure 3: Averaged sentence-level BLEU scores for the different models binned with respect to number of distinct realizations.

of performance as the number of realizations increases.

The good performance of the 'separate' MaxEnt model as compared to the language model seems to give encouraging evidence for the utility of including structural, syntactic features in a model for stochastic realization ranking—especially when considering that the n-gram model is trained on the entire BNC, while the basic MaxEnt model is trained and tested by ten-fold cross-validation on the still relatively limited set of 864 paraphrased items in the Rondane treebank. However, the relative performance of these models probably also says a lot about the importance of using training data that is attuned to the domain of application. With respect to the language model, roughly 15% of the word forms in the data set are out-of-vocabulary items and the vast majority of these are Norwegian proper names (often denoting hiking destinations and such). We have yet to try applying the language model using a similar cross-validation scheme for interpolating additional n-gram models trained on the Rondane treebank. Analogously, in order to test the cross-domain performance of the structural models, we also plan to construct paraphrased treebanks for other parts of the Redwoods data.

When compared to our initial experiments on smaller data sets (Velldal et al., 2004), we find that the MaxEnt models, using structural features, clearly benefit from increased amounts of training data, even though the degree of ambiguity (i.e. number of paraphrases) per item has also increased. The perfomance of the n-gram model seems to degrade more sharply with respect to the number of per-item paraphrases, and the general tendency reported by Velldal et al. (2004) is amplified: discriminative realization ranking with access to structural properties, when trained on small amounts of domain-specific data, appears to outperform the traditional method of selecting among competing strings purely by means of n-gram language models.

5 Summary and Further Directions

For the relatively coherent LOGON domain at least, a small training set of some 900 automatically 'annotated' paraphrases combined with a discriminative model adapted from earlier parse selection work improves substantially over a language model trained on all of the BNC. Our results suggest that this use of domain-specific treebanks—and the underlying assumption of relative 'naturalness' of the original, corpus-attested realizations—provide a good handle on ranking generator outputs, and that structural, linguistic information as is available to the log-linear model is of central importance for this task. Table 4 summarizes the results obtained both with the models developed in this paper and the models used in Velldal et al. (2004). We see that adding additional structural features and n-grams over fine-grained pre-terminal types enabled a substantial improvement in the performance of the MaxEnt ranker.

When we highlighted the similarities to parse selection, there were also some important differences that were glossed over. In relation to parsing, distinct system outputs typically have distinct semantics and it seems more reasonable to only count one or a few as correct and the

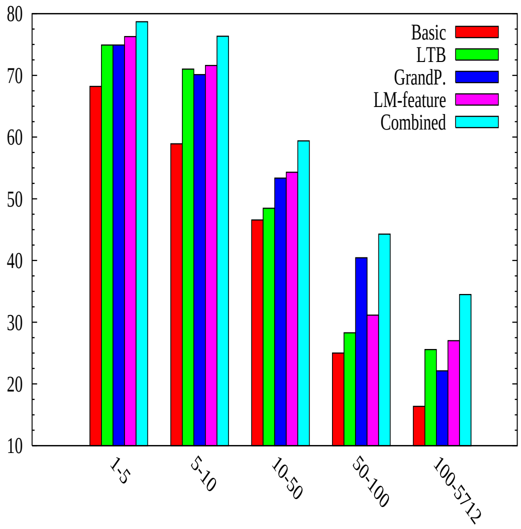

Figure 4: Exact match accuracy scores for different configurations of MaxEnt features, viz. (a) the basic MaxEnt model used by Velldal et al. (2004) ('Basic'); (b) adding n-gram features defined over lexical types in the derivation trees ('LTB'); (c) adding grandparenting features ('GrandP.'); (d) adding the language model feature ('LM-feature'); and, finally, (e) the combined model using all feature types ('Combined'). The data items are binned with respect to number of distinct realizations.

model configuration	**exact**	**BLEU**
language model of (Velldal et al., 2004)	48.46	0.878
basic model of (Velldal et al., 2004)	51.36	0.897
basic plus lexical type bi-grams	58.05	0.898
basic blus grandparenting	59.83	0.906
basic plus both of the above	61.58	0.903
basic plus language model	60.71	0.915
basic plus all of the above	65.63	0.920

Table 4: Performance summaries of best-performing realization rankers in various configurations, when compared to the original set-up of Velldal et al. (2004). While the exact match accuracy yields a broader spread of results, using BLEU as a string similarity measure confirms the overall trend of increased ranker performance when adding more structural features.

others as plain wrong. In realization ranking, on the other hand, it is perhaps more meaningful to think of a graded continuum of more or less natural verbalizations (given an input semantics). All outputs of the LKB realizer are semantically equivalent and guaranteed to be well-formed with respect to the underlying grammar. This means that the kind of properties we aim at capturing with the discriminative model are soft constraints that govern the degree of 'correctness' among competing paraphrases. Osborne (2000) and Malouf & Noord (2004) describe an approach to parse disambiguation using maximum entropy models where the empirical distribution that defines the constraints for the model are not based on frequency counts from a corpus but rather some measure of similarity towards the reference. Defining such a preference weighting of the candidate paraphrases (e.g. by using BLEU or other string-similarity measures typically used for evaluation) prior to training might be a well-suited approach also for building models for realization ranking. In initial experiments, however, we were unable to improve ranker performance over the results reported here when training our MaxEnt model against a graded distribution, although we have not yet obtained conclusive results for this set-up.

More practically, the way our realization rankers actually get deployed in the LOGON system is by means of *selective unpacking* from the packed generator forest: Carroll & Oepen (2005) present the unpacking procedure in full detail, but quite obviously there is a tradeoff between the ability to prune competing but dis-preferred realizations early, on the one hand, and improved realization ranking accuracy obtained from feature templates that take into account structural properties of larger constituents, on the other hand.

Acknowledgements

This paper has benefited in many ways from feedback given by members of the LOGON research team at Oslo University, specifically Francis Bond and Dan Flickinger. Comments given to a preliminary report on the underlying research by the audience of the 2004 Workshop on Treebanks and Linguistic Theories and the suggestions by anonymous reviewers for MT Summit 2005 have helped improve the presentation.

References

Carroll, J., Copestake, A., Flickinger, D., & Poznanski, V. (1999). An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation* (pp. 86–95). Toulouse, France.

Carroll, J., & Oepen, S. (2005). High efficiency realization for a wide-coverage unification grammar. In R. D. and (Ed.), *Proceedings of the 2nd International Joint Confer-*

ence on Natural Language Processing. Jeju, Republic of Korea.

Chen, S. F., & Rosenfeld, R. (1999). *A Gaussian prior for smoothing maximum entropy models* (Tech. Rep.). Carnegie Mellon University. (Technical Report CMUCS-CS-99-108)

Copestake, A., Flickinger, D., Malouf, R., Riehemann, S., & Sag, I. (1995). Translation using minimal recursion semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation.* Leuven, Belgium.

Flickinger, D. (2002). On building a more efficient grammar by exploiting types. In S. Oepen, D. Flickinger, J. Tsujii, , & H. Uszkoreit (Eds.), *Collaborative language engineering: A case study in efficient grammar-based processing* (pp. 1–17). CSLI Press.

Forsbom, E. (2003). Training a super model look-alike: Featuring edit distance, n-gram occurrence, and one reference translation. In *Proceedings of the Workshop on Machine Translation Evaluation. Towards Systemizing MT Evaluation.* New Orleans, LO.

Johnson, M., Geman, S., Canon, S., Chi, Z., & Riezler, S. (1999). Estimators for stochastic 'unification-based' grammars. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics* (pp. 535–541). College Park, MD.

Johnson, M., & Riezler, S. (2000). Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the 1st Conference of the North American Chapter of the ACL.* Seattle, WA.

Langkilde, I., & Knight, K. (1998). The practical value of n-grams in generation. In *International natural language generation workshop.*

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning.* Taipei, Taiwan.

Malouf, R., & Noord, G. van. (2004). Wide coverage parsing with stochastic attribute value grammars. In *Proceedings of the IJCNLP workshop Beyond Shallow Analysis.* Hainan, China.

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V., Jin, Z., & Radev, D. (2004). A smorgasbord of features for statistical machine translation. In *Proceedings of the 5th Conference of the North American Chapter of the ACL.* Boston.

Oepen, S., Dyvik, H., Lønning, J. T., Velldal, E., Beermann, D., Carroll, J., Flickinger, D., Hellan, L., Johannessen, J. B., Meurer, P., Nordgård, T., & Rosén, V. (2004). Som å kapp-ete med trollet? Towards MRS-based Norwegian – English Machine Translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation* (pp. 11 – 20). Baltimore, MD.

Oepen, S., Toutanova, K., Shieber, S., Manning, C., Flickinger, D., & Brants, T. (2002). The LinGO Redwoods treebank. Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics.* Taipei, Taiwan.

Osborne, M. (2000). Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the 18th International Conference on Computational Linguistics.* Saarbrücken, Germany.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu. A method for automatic evaluation of Machine Translation. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics* (pp. 311–318). Philadelphia, PA.

Toutanova, K., & Manning, C. D. (2002). Feature selection for a rich HPSG grammar using decision trees. In *Proceedings of the 6th Conference on Natural Language Learning.* Taipei, Taiwan.

Toutanova, K., Manning, C. D., Shieber, S. M., Flickinger, D., & Oepen, S. (2002). Parse disambiguation for a rich hpsg grammar. In *First workshop on treebanks and linguistic theories.* Sozopol, Bulgaria.

Velldal, E., Oepen, S., & Flickinger, D. (2004). Paraphrasing treebanks for stochastic realization ranking. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories.* Tübingen, Germany.

White, M. (2004). Reining in CCG chart realization. In *Proceedings of the 3rd International Conference on Natural Language Generation.* Hampshire, UK.