# Towards Hybrid Quality-Oriented Machine Translation

## — On Linguistics and Probabilities in MT —

**Stephan Oepen**[♣♠]**, Erik Velldal**[♣]**, Jan Tore Lønning**[♣]**,**
**Paul Meurer**[♡]**, Victoria Rosén**[◇]**, and Dan Flickinger**[♠]

[♣] Department of Informatics, University of Oslo, Norway
[♠] Center for the Study of Language and Information, Stanford University, USA
[♡] Centre of Culture, Language and Information Technology, University of Bergen, Norway
[◇] Department of Linguistics, University of Bergen, Norway

### Abstract

We present a hybrid MT architecture, combining state-of-the-art linguistic processing with advanced stochastic techniques. Grounded in a theoretical reflection on the division of labor between rule-based and probabilistic elements in the MT task, we summarize per-component approaches to ranking, including empirical results when evaluated in isolation. Combining component-internal scores and a number of additional sources of (probabilistic) information, we explore discriminative re-ranking of $n$-best lists of candidate translations through an eclectic combination of knowledge sources, and provide evaluation results for various configurations.

## 1 Background—Motivation

Machine Translation is back in fashion, with data-driven approaches and specifically Statistical MT (SMT) as the predominant paradigm—both in terms of scientific interest and evaluation results in MT competitions. But (fully-automated) machine translation remains a hard—if not ultimately impossible—challenge. The task encompasses not only all strata of linguistic description—phonology to discourse—but in the general case requires potentially unlimited knowledge about the actual world and situated language use (Kay, 1980, 1997). Although the majority of commercial MT systems still have large sets of hand-crafted rules at their core (often using techniques first invented in the 1960s and 1970s), MT research in the once mainstream linguistic tradition has become the privilege of a small, faithful minority.

Like a growing number of colleagues, we question the long-term value of *purely* statistical (or data-driven) approaches, both practically and scientifically. Large (parallel) training corpora remain scarce for most languages, and word- and phrase-level alignment continue to be active research topics. Assuming sufficient training material, statistical translation quality still leaves much to be desired; and probabilistic NLP experience in general suggests that one must expect 'ceiling' effects on system evolution. Statistical MT research has yet to find a satisfactory role for linguistic analysis; on its own, it does not further our understanding of language.

Progress on combining rule-based and data-driven approaches to MT will depend on a sustained stream of state-of-the-art, MT-oriented linguistics research. The Norwegian LOGON initiative capitalizes on linguistic precision for high-quality translation and, accordingly, puts scalable, general-purpose linguistic resources—complemented with advanced stochastic components—at its core. Despite frequent cycles of overly high hopes and subsequent disillusionment, MT in our view is the type of application that may demand knowledge-heavy, 'deep' approaches to NLP for its ultimate, long-term success. Much like Riezler & Maxwell III (2006) and Llitjós & Vogel (2007)—being faithful minority members ourselves—we approach a hybrid MT architecture with a semantic transfer backbone as our vantage point. Plurality of approaches to grammatical description, reusability of component parts, and the interplay of linguistic and stochastic processes are among the strong points of the LOGON system.

In the following, we provide a brief overview of the LOGON architecture (§ 2) and a bit of theoretical reflection on the role of probability theory

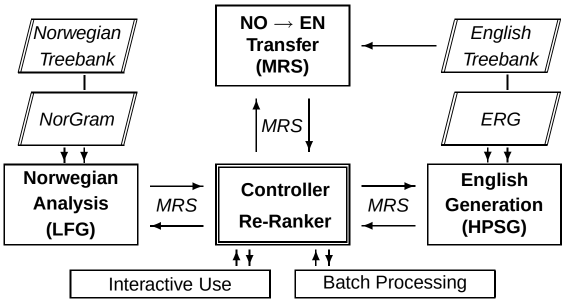

Figure 1: Schematic system architecture: the central controller brokers intermediate representations among the three processing components, accumulating candidate translations and, ultimately re-ranking the $n$-best list.

in finding optimal translations (§ 3). Sections § 4 through § 6 review component-internal ranking in the LOGON pipeline. Finally, § 7 outlines our approach to end-to-end re-ranking, including empirical results for various setups. We conclude with reflections on accomplishments so far and ongoing work in § 8.

## 2 LOGON—Hybrid Deep MT

The LOGON consortium—the Norwegian universities of Oslo (coordinator), Bergen, and Trondheim—has assembled a 'deep' MT prototype over the past four years, expending around fifteen person years on its core translation system. The LOGON pipeline comprises grammar-based parsing, transfer of underspecified Minimal Recursion Semantics (MRS; Copestake, Flickinger, Pollard, & Sag, 2005), and full tactical generation (aka realization). NorGram, the analysis grammar, is couched in the LFG framework and has been continuously developed at the University of Bergen since 1999. Conversely, the generation grammar, ERG (Flickinger, 2000), builds on the HPSG theory of grammar, and has been under development at CSLI Stanford since around 1993. While both analysis and generation deploy general-purpose linguistic resources and processing tools, LOGON had to develop its MRS transfer formalism and Norwegian – English (NoEn) transfer grammar from scratch. The transfer engine—unification-based, resource-sensitive rewriting of MRS terms—constitutes a new generic tool (that is already used for other language pairs and even non-MT tasks), but most of the NoEn transfer grammar is specific to the LOGON language pair and application. Figure 1 shows a schematic view of the LOGON architecture; Oepen et al. (2004) provide a more detailed overview of the LOGON approach.

| set | # | words | coverage | strings |
|---|---|---|---|---|
| $\mathbf{JH_d}$ | 2146 | 12.6 | 64.8 | 266 |
| $\mathbf{JH_t}$ | 182 | 11.7 | 63.2 | 114.6 |

Table 1: LOGON development and held-out corpora (for the *Jotunheimen* segment). Average string length and end-to-end coverage on the two sets are comparable, but the average number of candidate translations is higher on the development data.

In a nutshell, the role of the rule-based components in LOGON is to delineate the space of grammatically and semantically coherent translations, while the ranking of competing hypotheses and ultimately the selection of the best candidate(s) is viewed as a probabilistic task. Parsing, transfer, and realization each produce, on average, a few hundred candidate outputs for one input. Hence, exhausting the complete fan-out combinatorics can be prohibitively expensive, and typically we limit the number of hypotheses passed downstream to a relatively small $n$-best list. For all results reported presently, the fan-out branching factor was limited to a maximum of five output candidates from parsing and (within each branch) transfer; because there is no further downstream processing after generation, we can afford more candidate realizations per input MRS—for a total of up to $5 \times 5 \times 50 = 1250$ distinct fan-out outcomes. However, it is quite common for distinct fan-out paths to arrive at equivalent outputs, for example where the same modifier attachment ambiguity may be present in the source and target language.

Both our linguistic resources, search algorithms, and statistical models draw from contemporary, state-of-the art techniques and ongoing research in larger, non-MT communities. In this regard, the LOGON demonstrator provides a novel blending of approaches, where the majority of its component parts and linguistic resources have independent value (and often are used in parallel in other research efforts and applications).

The consortium circumscribed its domain and ambitions by virtue of a reference corpus of around 50,000 words of running text, six published tourism booklets on back-country activities

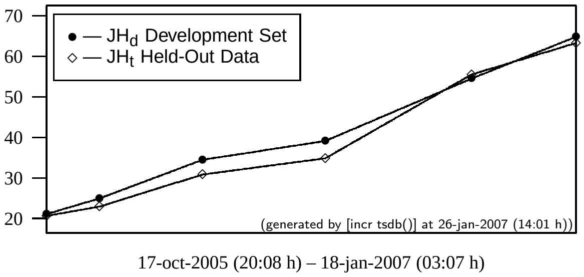

Figure 2: Evolution of end-to-end coverage over time: percentage of *Jotunheimen* inputs with at least one translation.

in Norway. In addition to one original translation, we contracted up to two additional reference translations; about ten per cent of the parallel corpus was held out for evaluation. Table 1 summarizes core metrics of the training and test sections of the *Jotunheimen* booklets, the largest segment and the one for which three reference translations are available. For model training and evaluation, about 670 of the Norwegian inputs and all (∼6,000) English references were manually treebanked (see below).

Aiming primarily to gauge the utility of its 'pure' setup (rather than for a complete MT solution) at the current stage, the consortium did not 'diffuse' its linguistic backbone with additional robustness measures. Accordingly, the overall error rate is the product of per-component errors, and gradually building up end-to-end coverage—specifically harmonizing semantics for a wide variety of constructions cross-linguistically—was a major part of system development. Figure 2 depicts the evolution of end-to-end coverage in the past year and a half. Upon completion of active development, system performance on held-out data was determined retroactively (for earlier versions). In terms of end-to-end coverage at least, it is reassuring to observe that there are few differences between system behavior on development vs. held-out data: for this domain and genre, the final LOGON demonstrator translates about two thirds of its inputs.

## 3 Some Theoretical Reflections

Given our transfer system, where each of the three steps fan out, there are several possibilities for adding a stochastic component. What should be maximized, and how?

The first possibility is to rank the different components sequentially, one at a time. First rank the results of parsing and choose the topmost candidate, call it $F_1$. Then consider all the results of invoking transfer on $F_1$, and choose the one ranked highest, $E_1$. And finally choose the highest ranked realization $e_1$ of $E_1$. We will refer to this output as the *first translation*, corresponding to the top branch in Figure 3.

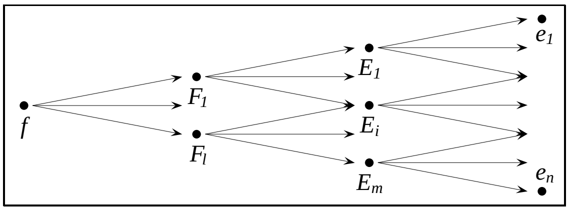

Figure 3: Abstract fan-out tree: each processing component operates non-deterministically, and distinct inputs can, in principle, give rise to equivalent outputs.

The second possibility is to try to find the *most likely path* through the fan-out tree, i.e. try to maximize:

$$\arg\max_{i,j,k} P(e_k|E_j)P(E_j|F_i)P(F_i|f)$$

The two approaches do not always yield the same result. Take as an example a sentence $f$ with two different analyses, $F_1$ and $F_2$, where the main difference between the two is that a particular word is ambiguous between a noun reading in $F_1$, and a verb reading in $F_2$. If the noun has many alternative realizations in the target language while the verb has few, the most likely path might be one that chooses the verb, i.e. passes through $F_2$.

The third possibility for the end-to-end ranking is to try to find the *most likely translation*, i.e.

$$\arg\max_{e} \sum_{F_i} \sum_{E_j} P(e_k|E_j)P(E_j|F_i)P(F_i|f)$$

This might result in a different top-ranked candidate than the most likely path in cases where several different paths result in the same output. Considering PP attachment ambiguities, for example, distinct intermediate semantic representations (pairs of $E_i$s and $F_j$s) can yield the same target string.

Which concept should we try to model? From a theoretical point of view, there are good arguments for choosing what we have called the first translation. It makes sense to try to select the most likely interpretation of what the producer of the source sentence has intended independently of how it gets translated. If one instead selects the most likely path, or the most likely translation, one might select a less likely interpretation of what the speaker had intended.

Our argument for the *first translation* can be illustrated within our earlier example of a word-level noun vs. verb ambiguity in analysis. The many different realizations of the noun in the target language may fall into classes of near synonyms, in which case it does not matter for the quality of the result which synonym is chosen. Even though each of the individual realizations has a low probability, it may be a good translation.

Observe here also that an automatic evaluation measure—measuring the similarities to a set of reference translations, like the BLEU metric (Papineni, Roukos, Ward, & Zhu, 2002)—will favor the view of *most likely translation*. We conjecture, however, that a human evaluation will correspond better to the first translation.

From a theoretical point of view, it seems most correct to go for the first translation. But it presupposes that we choose the correct interpretation of the source sentence, which we cannot expect to always do. In cases where we have chosen an incorrect analysis, this might be revealed by trying to translate it into the target language and consider the result. If all the candidate translations sound bad—or have a very low probability—in the target language, that can be evidence for dispreferring this analysis. Hence information about probabilities from later components in the pipeline may be relevant, not for overwriting analysis probabilities, but for helping in selecting them.

We will in the following first review how LOGON employs component ranking for choosing the first translation, and then consider an end-to-end re-ranking which attempts to find the most probable translation, by directly estimating the posterior translation probability $P(e|f)$.

## 4 Parse Selection

In a sister project to LOGON, the TREPIL project, a toolkit for building parsebanks of LFG analyses is being developed (Rosén, Smedt, & Meurer, 2006). This toolkit, called the LFG Parsebanker, was used to build a treebank for the LOGON development corpus. Parse selection in LOGON uses training data from this treebank; all sentences with full parses with low ambiguity (fewer than 100 readings) were at least partially disambiguated.

| **ambiguity** | **#** | **exact match** | **five-best** |
|---|---|---|---|
| $50-100$ | 16 | 34.4 (17.2) | 56.2 (55.0) |
| $25-49$ | 28 | 30.4 (21.4) | 62.5 (54.3) |
| $10-24$ | 43 | 58.1 (25.3) | 89.5 (73.9) |
| $2-9$ | 53 | 70.8 (35.1) | 96.2 (91.0) |
| **total** | **140** | **53.8 (27.3)** | **84.3 (74.3)** |
| $50-100$ | 16 | 43.7 (17.2) | 81.2 (55.0) |
| $25-49$ | 28 | 50.0 (21.4) | 78.6 (54.3) |
| $10-24$ | 43 | 67.4 (25.3) | 90.7 (73.9) |
| $2-9$ | 53 | 72.6 (35.1) | 100. (91.0) |
| **total** | **140** | **63.2 (27.3)** | **90.7 (74.3)** |

Table 2: Evaluation of parse selection with a model trained with standard feature function templates of the XLE (upper part, as used in LOGON,) and with a discriminant model (lower part, not yet used). Figures are given for the percentage of exact matches and matches among the five top-ranked analyses. Figures in parentheses show a random choice baseline. Both models were trained on seven of nine treebanked texts and evaluated on the two remaining texts.

The parse selection method employed in the LOGON demonstrator uses the stochastic disambiguation scheme and training software developed at PARC (Riezler & Vasserman, 2004). The XLE system provides a set of parameterized feature function templates that must be expanded in accordance with the grammar or the training set at hand. Application of these feature functions to the training data yields feature forests for both the labeled data (the partially disambiguated parse forests) and the unlabeled data (the full parse forests). These feature forests are the input to the statistical estimation algorithm, which generates a property weights file that is used to rank solutions.

One of the challenges in applying the probability model to a given grammar and training set is the choice of appropriate feature functions. We have pursued two approaches for choosing feature functions. In the first approach, we started with a significant subset of the predefined feature function templates and expanded each of them in all possible ways that would result in a non-zero value on at least one parse in the train-

```
{
  prpstn_m[MARG _recommend_v]
  _recommend_v[ARG1 pron, ARG2 _hike_n]
  _a_q[ARG0 _hike_n]
  _around_p[ARG1 _hike_n, ARG2 _source_n]
  implicit_q[ARG0 _source_n]
  poss[ARG1 _waterway_n, ARG2 _source_n]
  def_q[ARG0 _waterway_n]
}
```

Figure 4: Variable-free reduction of the MRS for the utterance ‘We recommend a hike around the waterway’s sources’.

ing set; this could be done automatically. The second approach is motivated by the hypothesis that discriminants, as used in manual annotation (Carter, 1997), represent promising alternative feature functions to the predefined templates. Initial tests (see table 2) show that the discriminant approach (which is not yet used in the LOGON system) scores better than the template-based approach.

## 5 Ranking Transfer Outputs

While MRS formulae are highly structured graphs, Oepen & Lønning (2006) suggest a reduction into a variable-free form that resembles elementary dependency structures. For the ranking of transfer outputs, MRSs are broken down into basic dependency triples, whose probabilities are estimated by adaptation of standard $n$-gram sequence modeling techniques. The actual training is done using the freely available CMU SLM toolkit (Clarkson & Rosenfeld, 1997).

Based on a training set of some 8,500 in-domain MRSs, viz. the treebanked version of the English translations of the (full) LOGON development corpus, our target language ‘semantic model’ is defined as a smoothed tri-gram model over the reduction of MRSs into dependency triples. Figure 4 shows an example structure, corresponding to a total of ten triples, including $\langle$_around_p, ARG1, _hike_n$\rangle$. The ‘vocabulary’ of the model comprises some 4,400 distinct semantic predicates and role labels, for a total number of around 51,000 distinct triples. Similarly, post-transfer English MRSs are broken down into segments of dependency triples and ranked according to the perplexity scores assigned by the semantic model.

We lack a transfer-level ‘treebank’ to evaluate MRS ranking in isolation, but in lieu of such data, we can contrast end-to-end system performance on the JH$_t$ test set. When passing an unranked, random selection of five transfer outputs downstream, the success rate in generation drops to 82.7 per cent (down from 86.5 per cent in ranked, five-best mode). Restricting the comparison to the 109 items that translate in both configurations, our BLEU score over the *first* translation drops from 37.41 to 30.29.[1]

## 6 Realization Ranking

*Realization ranking* is the term we use for the task of discriminating between multiple surface forms generated for a given input semantics. By adapting methods previously used for parse selection, we are able to use treebank data for training a discriminative log-linear model for the conditional probability of a surface realization given an input MRS. Traditionally, however, the standard approach to tackling this problem of indeterminacy in generation is to use an *$n$-gram language model* (Langkilde & Knight, 1998; White, 2004; inter alios). Candidate strings are then ranked according to their ‘fluency’, indicated by the probabilities assigned by the LM. As a baseline for our discriminative model, we trained a tri-gram language model on an unannotated version of the British National Corpus (BNC), containing roughly 100 million words. As in the case of the MRS ranker, we used the CMU SLM toolkit for training, resulting in a Witten-Bell discounted back-off model.

When evaluated in terms of exact match accuracy on the JH$_d$ development set,[2] the LM ranker achieves $53.2\%$, which is well above the random choice baseline of $28.7\%$. However there are many well-known limitations inherent to the $n$-gram approach, such as its inability to capture long-range dependencies and dependencies between non-contiguous words. More generally, the simple $n$-gram models are purely surface ori-

[1] BLEU measures in all our experiments are calculated using the freely available NIST toolkit (in its version 11b).

[2] Note that, when evaluating realization rankers in isolation, we use a different version of the JH$_d$ data set. The MRSs in the generation treebank are here always underspecified with respect to information structure, such as passivization and topicalization. This means that the level of indeterminacy is somewhat higher than what is typically the case within the LOGON MT setting.

| model | exact match | five-best | WA |
|---|---|---|---|
| **BNC LM** | 53.24 | 78.81 | 0.882 |
| **Log-Linear** | 72.28 | 84.59 | 0.927 |

Table 3: Performance of the realization rankers. *BNC LM* is the $n$-gram ranker trained on the raw text version of the BNC. *Log-Linear* shows 10-fold cross-validated results for the discriminative model trained on a generation treebank, including the LM scores as a separate feature.

ented and thereby fail to capture dependencies that show a structural rather than sequential regularity. All in all, there are good reasons to expect to devise better realization rankers by using models with access to grammatical structure. Velldal, Oepen, & Flickinger (2004) introduced the notion of a *generation treebank*, which facilities the training of discriminative log-linear models for realization ranking in a similar fashion as for parse disambiguation. For further background on log-linear models, see § 7.

Our discriminative realization ranker uses a range of features defined over the derivation trees of the HPSG linguistic sign, recording information about local sub-tree configurations, vertical dominance relations, $n$-grams of lexical types, and more (Velldal & Oepen, 2006). When trained and tested by ten-fold cross-validation on a generation treebank created for the JH$_d$ data set, this model achieves $70.28\%$ exact match accuracy, clearly outperforming the $n$-gram-based LM by a good margin (again, the random choice baseline is $28.7\%$). However, by including the scores of the LM as an additional feature, we are able to further boost accuracy up to $72.28\%$. Table 3 summarizes the results of the two different types of realization rankers. The evaluation also includes exact match accuracy within the five top-ranked candidates, as well as average sentence-level *word accuracy* (WA), which is a string similarity measure based on edit distance.

## 7 End-to-End Re-Ranking

Section § 3 already suggests one consideration in favor of re-ranking the complete list of candidate translations once fan-out is complete: component-internal probabilistic models are fallible. Furthermore, besides analysis-, transfer-, and realization-internal information, there are additional properties of each hypothesized pair $\langle f, e\rangle$ that can be brought to bear in choosing the 'best' translation, for example a measure of how much reordering has occurred among corresponding elements in the source and target language, or the degree of harmony between the string lengths of the source and target.

Log-linear models provide a very flexible framework for discriminative modeling that allows us to combine disparate and overlapping sources of information in a single model without running the risk of making unwarranted independence assumptions. In this section we describe a model that directly estimates the posterior translation probability $P_\lambda(e|f)$, for a given source sentence $f$ and translation $e$. Although the re-ranker we describe here is built on top of a hybrid baseline system, the overall approach is similar to that described by Och & Ney (2002) in the context of SMT.

**Log-Linear Models** A log-linear model is given in terms of (a) a set of *specified features* that describe properties of the data, and (b) an associated set of *learned weights* that determine the contribution of each feature. One advantage of working with a discriminative re-ranking setup is that the model can use global features that the baseline system would not be able to incorporate. The information that the feature functions record can be arbitrarily complex, and a given feature can even itself be a separate statistical model. In the following we first give a brief high-level presentation of conditional log-linear modeling, and then we go on to present the actual feature functions in our setup.

Given a set of $m$ real-valued features, each pair of source sentence $f$ and target sentence $e$ are represented as a feature vector $\Phi(f, e) \in \Re^m$. A vector of weights $\lambda \in \Re^m$ is then fitted to optimize some objective function of the training data. For the experiments reported in this paper the weights are fitted to maximize the conditional (or *pseudo*) likelihood (Johnson, Geman, Canon, Chi, & Riezler, 1999).[3] In other words, for each input source sentence in the training data we seek to maximize

[3] For estimation we use the TADM open-source toolkit (Malouf, 2002), using its *limited-memory variable metric* as the optimization method. As is standard practice, the model is regularized by including a zero-mean Gaussian prior on the feature weights to reduce the risk of overfitting.

the probability of its annotated reference translation relative to the other competing candidates. However, for future work we plan to also experiment with optimizing the scores of a given evaluation metric (e.g. BLEU) directly, following the Minimum Error Rate approach of Och (2003).

The three most fundamental features that are supplied in our log-linear re-ranker correspond to the three ranking modules of the baseline system, as described in Sections § 4, § 5, and § 6 above. In other words, these features record the scores of the parse ranker, the MRS ranker, and the realization ranker, respectively. But our re-ranker also includes several other features that are not part of the baseline model.

**Other Features** Our experiments so far have taken into account another eight properties of the translation process, in some cases observing internal features of individual components, in others aiming to capture global information. The following paragraphs provide an informal overview of these additional features in our log-linear re-ranking model.

LEXICAL PROBABILITIES One additional feature type in the log-linear model corresponds to *lexical translation probabilities*. These are estimated on the basis of a small corpus of Norwegian – English parallel texts, comprising 22,356 pairs of aligned sentences.[4] First, GIZA$^{++}$ is used for producing word alignments in both directions, i.e. using both languages as source and target in turn. On the basis of these alignments we then estimate a maximum likelihood translation table, again in both directions.[5] Finally, for each bi-directional sentence pair $\langle e, f\rangle$ and $\langle f, e\rangle$, the corresponding feature in the end-to-end ranker is computed as the length-normalized product of all pairwise word-to-word probabilities.

STRING PROBABILITY Although a part of the (conditional) realization ranker already, we include the string probability (according to the tri-gram language model trained on the BNC) of candidate translations $e_k$ as an independent indicator of output fluency.

DISTORTION Elementary predications (EPs) in our MRS are linked to corresponding surface elements, i.e. sub-string pointers. Surface links are preserved in transfer, such that post-generation, for each EP—or group of EPs, as transfer need not be a one-to-one mapping—there is information about its original vs. its output sub-string span. To gauge reordering among constituents, for both the generator input and output, each EP is compared pairwise to other EPs in the same MRS, and each pair classified with regard to their relative surface positions. Comparing the input and output MRS, we consider corresponding pairs of EP pairs; the distortion metric for a pair of aligned EPs measures their class difference, where for example a change from overlapping to adjacent is penalized mildly, while inverting a precedence relation comes at a higher cost. Finally, the distortion metric for a pair of MRSs is the sum of their per-EP distortion metrics, normalized by the total number of EP pairs.

STRING HARMONY Seeing typological similarity between Norwegian and English, much like for the distortion metric, we assume that there are systematic correspondences at the string level between the source and its translation. To enable the re-ranker to take into account length effects, we include the ratio of word counts, $|e|/|f|$, as a feature in the model.

TRANSFER METRICS Two additional features capture information about the transfer step: the total number of transfer rules that were invoked (as a measure of transfer granularity, e.g. where idiomatic transfer of a larger cluster of EPs contrasts with stepwise transfer of component EPs), as well as the ratio of EP counts, $|E|/|F|$.

SEMANTIC DISTANCE Generation proceeds in two phases: a chart-based bottom-up search enumerates candidate realizations, of which a final semantic compatiblity test selects the one(s) whose MRS is subsumed by the original generator input MRS (Carroll & Oepen, 2005). Given an imperfect input (or error in the generation grammar), it is possible for none of the candidate outputs to fulfill the semantic compatiblity test. In this case, the generator will gradually relax MRS com-

[4] Of these, 9,410 sentences are taken from the LOGON development data, while an additional 12,946 sentences are from the English-Norwegian Parallel Corpus (Oksefjell, 1999).

[5] The ML estimation of the lexical probabilities, as well as the final word alignments produced from the output of GIZA$^{++}$, are carried out using the training scripts provided by Phillip Koehn, and as distributed with the phrase-based SMT module Pharaoh (Koehn, 2004).

parison, going through seven pre-defined levels of semantic mismatch, which we encode as one integer-valued feature in the re-ranking model.

**Training the Model** While batch translating, the LOGON controller records all candidate translations, intermediate semantic representations, and a large number of processing and resource consumption properties in a database, which we call a *profile* (in analogy to software engineering; Oepen et al., 2005). Given the system configuration summarized in Sections § 2 through § 6, we use the JH$_d$ batch profile to train and optimize a log-linear re-ranker. The experimentation infrastructure, here, is essentially the same as in our discriminative realization ranker—the combination of the [incr tsdb()] profiler, the TADM maximum entropy toolkit, and tools for efficient cross-valiation experiments with large data and feature sets (Velldal, 2007).

For training purposes, we mechanically 'annotated' candidate translations by means of the sentence-level NEVA string similarity measure, applied to actual LOGON outputs compared to JH$_d$ reference translations. NEVA is a reformulation of BLEU that avoids many of the problems associated with applying BLEU at the sentence level, and is computed as the arithmetic mean of the raw $n$-gram precision scores (Forsbom, 2003). For each source sentence, we mark the translation(s) with maximum NEVA score (among all candidate outputs for this input) as preferred, thus constructing an empirical distribution where estimation of log-linear model parameters amounts to adjusting conditional probabilities towards higher NEVA scores.

Seeing that the model includes diverse feature types—probabilities, perplexity values, un-normalized log-linear scores, and non-probabilistic quantities—feature values are normalized into a comparable range, using min-max scaling. The hyper-parameters of the model—the TADM convergence threshold and variance of the Gaussian prior—were optimized by ten-fold cross-validation on the training corpus.

**Empirical Results** Table 4 summarizes end-to-end system performance, measured in BLEU scores, for various strategies of selecting among the $n$-best lists obtained from $5 \times 5 \times 50$ fan-out. In all cases, scoring has been reduced to those inputs actually translated by the LOGON system, i.e. $64.8\%$ and $63.2\%$ of the development (JH$_d$) and held-out (JH$_t$) corpora, respectively. As a baseline measure, we used random choice of one output in each context (averaged over twenty iterations), resulting in (estimable) BLEU scores of $34.18$ and $30.84$, respectively.

| set | # | chance | first | LL | top | judge |
|---|---|---|---|---|---|---|
| **JH$_d$** | 1391 | 34.18 | 40.95 | 44.10 | 49.89 | – |
| **JH$_t$** | 115 | 30.84 | 35.67 | 38.92 | 45.74 | 46.32 |

Table 4: BLEU scores for various re-ranking configurations, computed over only those cases actually translated by LOGON (second column). For all configurations, BLEU results on the training corpus are higher by about four points.

As an upper bound on re-ranking efficacy, Table 4 provides two 'oracle' scores: the first, labeled *top*, is obtained from selecting translations with maximal NEVA scores, i.e. using sentence-level NEVA as a proxy for corpus-level BLEU. The second, labeled *judge*, reflects the annotations of a human judge on the JH$_t$ held-out data: considering all available candidates, a native speaker of (American) English and near-native speaker of Norwegian, in each case, picked the translation judged most appropriate (or, in some cases, least awful). Oracle BLEU scores reach $49.89$ and $46.32$, for JH$_d$ and JH$_t$, respectively.

Finally, the column labeled *first* in Table 4 corresponds to the *first translation* concept introduced in § 3 above, and the *LL* column to our log-linear re-ranker (maximizing the *log-likelihood* of the training data). Both clearly improve over the random choice baseline, but the re-ranker outperforms the first translation approach by a large margin—thus returning on the investment of extra fan-out and end-to-end re-ranking. However, at BLEU scores of $44.10$ and $38.92$, respectively, our current re-ranking setup also leaves ample room for further improvements towards the 'oracle' upper bound. We anticipate that fine-tuning the log-linear model, inclusion of additional features, and experimentation with different estimation techniques (see below) will allow us to narrow this differential further.

## 8 Conclusions—Outlook

The future of MT has been (mis-)diagnosed as ‘just around the corner’ since the beginning of time, and there is no basis to expect a breakthrough in fully-automated MT in the foreseeable future. But yet we see progress along the way, specifically in the sustained development of large-scale, general-purpose language technology and its ever tighter integration with refined stochastic techniques.

Among the main results of the Norwegian LOGON initiative is its proof-of-concept demonstrator for quality-oriented, hybrid MT grounded in independently developed computational grammars. The tight coupling of hand-built linguistic resources results in an MT pipeline where, to a very high degree, all candidate translations are (a) related to the source utterance in a systematic—albeit at times unlikely—way and (b) grammatically well-formed. Combining an $n$-best beam search through the space of fan-out combinatorics with stochastic rankers at each step, as well as with discriminative end-to-end re-ranking yields a flexible solution, offering a clear precision vs. efficiency trade-off. For its bounded domain (and limited vocabulary of around 5,000 lexemes), the LOGON system succeeds in translating about two thirds of unseen running text, where BLEU scores and project-internal inspection of results suggest a high degree of output quality. This configuration could, in principle, be an interesting value proposition by itself—as a tool to professional translators, for example. A more systematic, human judgment study of system outputs (for various selection strategies) is currently underway, and we expect results to become available in June this year.

In ongoing work, we aim to further improve re-ranking performance, for example by assessing the relative contribution of individual features, fine-tuning parameter estimation, and including additional properties. Our current maximum likelihood training of the log-linear model is based on a binarized empirical distribution, where for each input we consider the candidate translation(s) with maximum NEVA score(s) as preferred, and all others as dis-preferred. Obviously, however, the degradation in quality among alternate candidates is continuous (rather than absolute), and we have started experimentation with a graded empirical distribution, adapting the approach of Osborne (2000) to the re-ranking task. Finally, in a parallel refinement cycle, we aim to contrast our current (LL) re-ranking model with Minimum Error Rate (MER) training, a method that aims to estimate model parameters to directly optimize BLEU scores (or another quality metric) as its objective function.

Trading coverage for increased output quality may be economic for a range of tasks—say as a complement to other tools in the workbench of a professional translator. Our re-ranking approach, with access to rich intermediate representations, probabilities, and confidence measures, provides a fertile environment for experimentation on *confidence-centric* MT. Applying thresholding techniques on the probability distribution of the re-ranking model, for example, we plan to experimentally determine how much translation quality can be gained by making the candidate selection more restrictive. Alternatively, one can imagine applying yet another model to this task, a classifier deciding on which candidate translations constitute worthy outputs, and which are best suppressed.

The availability of off-the-shelf SMT tools has greatly contributed to re-energized interest and progress in MT in the recent past. We believe that advances in hybrid MT would equally benefit from a repository of general-purpose, easy-to-use linguistic resources. Except for the proprietary XLE, all LOGON results—treebanks, grammars, and software—are available for public download.

## References

Carroll, J., & Oepen, S. (2005). High-efficiency realization for a wide-coverage unification grammar. In R. Dale & K. F. Wong (Eds.), *Proceedings of the 2nd International Joint Conference on Natural Language Processing* (Vol. 3651, pp. 165 – 176). Jeju, Korea: Springer.

Carter, D. (1997). The TreeBanker. A tool for supervised training of parsed corpora. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering.* Madrid, Spain.

Clarkson, P., & Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge Toolkit. In *Proceedings of EuroSpeech.* Rhodes, Greece.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal Recursion Semantics. An introduction.

*Journal of Research on Language and Computation*, *3*(4), 281 – 332.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering, 6 (1)*, 15 – 28.

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, held in conjunction with MT SUMMIT IX.* New Orleans, USA.

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.

Kay, M. (1980). *The proper place of men and machines in translation* (Technical Report # CSL-80-11). Palo Alto, CA: Xerox Palo Alto Research Center.

Kay, M. (1997). It's still the proper place. *Machine Translation*, *12*(1 - 2), 35 – 38.

Koehn, P. (2004). Pharaoh. A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas* (pp. 115 – 124). Washington DC.

Langkilde, I., & Knight, K. (1998). The practical value of n-grams in generation. In *Proceedings of the 9th International Workshop on Natural Language Generation* (pp. 248 – 255). Ontario, Canada.

Llitjós, A. F., & Vogel, S. (2007). A walk on the other side. Adding statistical components to a transfer-based translation system. In *Proceedings of the HLT-NAACL workshop on Syntax and Structure in Statistical Translation.* Rochester, NY.

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

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics* (pp. 160 – 167). Sapporo, Japan.

Och, F. J., & Ney, H. (2002). Discriminative training and Maximum Entropy models for statistical machine translation. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics* (pp. 295 – 302). Philadelphia, PA.

Oepen, S., Dyvik, H., Flickinger, D., Lønning, J. T., Meurer, P., & Rosén, V. (2005). Holistic regression testing for high-quality MT. Some methodological and technological reflections. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation.* Budapest, Hungary.

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.* Baltimore, MD.

Oepen, S., & Lønning, J. T. (2006). Discriminant-based MRS banking. In *Proceedings of the 4th International Conference on Language Resources and Evaluation.* Genoa, Italy.

Oksefjell, S. (1999). A description of the English-Norwegian Parallel Corpus. Compilation and further developments. *International Journal of Corpus Linguistics*, *4*(2), 197 – 219.

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. (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.

Riezler, S., & Maxwell III, J. T. (2006). Grammatical machine translation. In *Proceedings of the Human Language Technology Conference and Annual Meeting of the North American Association for Computational Linguistics* (pp. 248 – 255). New York, NY.

Riezler, S., & Vasserman, A. (2004). Incremental feature selection and $l_1$ regularization for relaxed maximum-entropy modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Barcelona, Spain.

Rosén, V., Smedt, K. D., & Meurer, P. (2006). Towards a toolkit linking treebanking and grammar development. In *Proceedings of the 5th Workshop on Treebanks and Linguistic Theories* (pp. 55 – 66). Prague, Czech Republic.

Velldal, E. (2007). *Stochastic realization ranking.* Doctoral dissertation, University of Oslo, Oslo, Norway. (in preparation)

Velldal, E., & Oepen, S. (2006). Statistical ranking in tactical generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Sydney, Australia.

Velldal, E., Oepen, S., & Flickinger, D. (2004). Paraphrasing treebanks for stochastic realization ranking. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories* (pp. 149 – 160). Tübingen, Germany.

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