Using Log-linear Models for Selecting Best Machine Translation Output

Michael Carl
Institut für Angewandte Informationswissenschaft
Martin Luther Str. 14, 66111 Saarbrücken, Germany
E-mail: carl@iai.uni-sb.de

Abstract

We describe a set of experiments to explore statistical techniques for ranking and selecting the best translations in a graph of translation hypotheses. In a previous paper (Carl, 2007) we have described how the hypotheses graph is generated through shallow mapping and permutation rules . We have given examples of its nodes consisting of vectors representing morpho-syntactic properties of words and phrases. This paper describes a number of methods for elaborating statistical feature functions from some of the vector components. The feature functions are trained off-line on different types of text and their log-linear combination is then used to retrieve the best translation paths in the graph. We compare two language modelling toolkits, the CMU and the SRI toolkit and arrive at three results: 1) word-lemma based feature function models produce better results than token-based models, 2) adding a PoS-tag feature function to the word-lemma model improves the output and 3) weights for lexical translations are suitable if the training material is similar to the texts to be translated

1. Introduction

Feature functions are being increasingly used in recent Machine Translation (MT) approaches to select and rank translation candidates. Developed within a statistical processing framework (Och and Ney, 2002), the log-linear combination of feature functions provide a flexible framework for discriminative modeling that allows to combine disparate and overlapping sources of information in a single model

According to Oepen et al. (2007), "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." Each pair of source sentence **f** and target sentence **e** is represented as a real-valued feature vector *h*. A vector of weights *w* is then trained to optimize some objective function of the training data so as to allow a search procedure to find the target sentence $\hat{e}$ with the highest probability:

$$\hat{e} = \texttt{argmax} \sum_{m}^{M} w_m h_m(\cdot)$$

A number of feature functions have been explored in various system implementations (e.g. Oepen et al. 2007, Liu et al. 2007, Quirk 2007), separating the features roughly into source features, channel features and target features. Source features include probabilities of representations resulting from the SL analysis such as likelihood of the parse tree and dictionary matching. Channel models include SL-to-TL alignment and lexical translation probabilities. Target features refer to probabilities of the generated TL sentence, including, among other things, *n-gram* language models.

In contrast to most purely statistical MT systems[1] , we use rule-based methods to generate an AND/OR graph of translation hypotheses (Carl 2007) and a beam search algorithm to traverse the graph and feature functions to rank the paths, as e.g. in the MISTRAL system (Patry et

[1]A number of recent SMT architectures are described http://iwslt07.itc.it/menu/program.html

al, 2007). We report on two experimental settings, using different types of feature functions and different language modelling toolkits[2] . The experiments rely on four types of target models and a statistical lexical translation model which are described in section 2. All of the language models were trained on the BNC[3] . The BNC is a collection of tagged texts making use of the CLAWS5 tag set which comprises roughly 70 different tags. We also tested a lemma-tag co-occurrence model which was also trained on the BNC. In section 3 we describe additional feature functions, The lexical translation models are trained on an excerpt of the EUROPARL corpus.

The aim of the experimental setting was to figure out which combination of language models and lexical weights can enhance the quality of the translations, and whether the system can be tuned to a particular domain with lexical weights.

2. Comparing different language models

In a previous set of experiments (Carl 2007) we compared several language models trained with the CMU language modelling toolkit[4] . The CMU language modelling toolkit generates *n-gram* language models (LMs) from tokenized texts. The CMU toolkit generates a vocabulary of up to 65535 words which occur most frequently in the training material. It supports open LMs which account for unknown words and closed LMs which assume all tokens to be known in the training material. A LM made up of CLAWS5 tags would be a closed language model since there are less than 70 different tags in this tag set and all tags are likely to occur in the training material.

The closed LMs assume that only items in the training data will occur in the test data, while open LMs save some of the probability mass for (unknown) words in the test data which did not occur in the training set. These words will be mapped on the item UNK.

To find most suited LMs for our application, we have experimented with the following parameters:

1. open token-based LM:

   This language model works on (lower-cased) surface word-forms.

2. closed mixed token-tag LM:

   The vocabulary of this model consists of word tokens. Unknown words are mapped on their CLAWS5 tag.

3. closed mixed lemma-tag LM:

   The vocabulary of this model consists of lemmas. Similar to the closed mixed lemma-tag model, unknown lemmas are mapped on their CLAWS5 tag.

4. orthogonal lemma-tag LM:

   In the orthogonal lemma-tag model we actually computed three LMs:

   1. an *n-gram* CLAWS5 tag model
   2. an *m-gram* lemma model
   3. a co-occurrence weight of the lemmas, tag according to Laplace's law with the following equation:

$$w(lem, tag) = \frac{NL}{NL + C(lem)} * (C(lem, tag) + 1)$$

   Where *NL* is half the number of different tags, *C(lem)* is the number of occurrences of the lemma in the BNC and *C(lem,tag)* is the number of co-occurrences of a lemma and a tag.

Different *n-gram* language models were computed, based on 20K, 100K, 1M 5M and 6M sentences, from the BNC and with:

[2]Due to time constraints, we did not compare our results with Moses (Koehn et.al, 2007), which also provides possibilities of defining feature functions.

[3]The British National Corpus (BNC) consists of more than 100 million words in more than 6 million sentences http://www.natcorp.ox.ac.uk/

[4]The CMU SLM toolkit can be downloaded from http://www.speech.cs.cmu.edu/SLM_info.html

- $n=\{3,4,5\}$ for the lemma and token models
- $n=\{3,4,5,6,7\}$ for the tag models

In a set of experiments (cf. Carl 2007) we figured out that the orthogonal lemma-tag LM had the best performance, using a *3-gram* lemma model from 5M sentences and a *7-gram* tag model also from 5M sentences. These tests were run on a 200 sentences test corpus. There were three reference translations for each test sentence. The sentences were selected (and partially constructed) so that they cover a range of known translation problems including:

1. lexical translation problems: separable prefixes, fixed verb constructions, degree of adjectives and adverbs, lexical ambiguities, and others
2. syntactic translation problems: pronominalization, determination, word order, different complementation, relative clauses, tense/aspect, head switching, prepositions, category change, and others.

3. Testing Conditions

In another experiment we have extended the previous setting with the following features:

1. Using the SRI[5] language modelling toolkit instead of CMU
2. Using an additional test corpus of 200 sentences from the EUROPARL[6] corpus
3. Adding a further feature which takes into account weights of lexical translations

3.2 Performance of CMU and SRI toolkits

While both, CMU and SRI toolkits are open source projects, the reasoning for using SRI was to see whether there is any difference in performance compared to the CMU-toolkit, based on the fact that:

- SRI is referred to by most of the recent (MT) publications which make use of n-gram statistical language modelling
- SRI is extensively maintained, while the development of the CMU-kit seems to have stopped around 10 years ago
- SRI features numerous backoff models and allows for unlimited vocabulary

The SRI package is somewhat easier to use, due to its C++ implementation. However, it also seems to be a bit slower[7] .

We did not have the time to experiment with all the backoff models, mainly sticking to the (generally recommended) Kneser-Ney and Good-Turing Discount strategies. With these settings we did not find a significant difference between the SRI and the CMU toolkit. Notice, that we only experimented with the **orthogonal lemma-tag LM** and an **orthogonal token-tag model** as discussed earlier.

3.2 Test Corpus

The EUROPARL corpus was used for several reasons. First, we wanted to see how our results on the 200 test sentences compare with a set of sentences which have now become a standard for data-driven MT. We therefore extracted a set of 200 test translations from the EUROPARL corpus and run a number of identical system settings on both texts. Second, we wanted to see whether knowledge of a huge number of translations could be profitably used to enhance the quality of the METIS translations. The results of the comparison is given in section 4.

3.3 Training dictionary weights

For testing the adaptability of the system to EUROPARL terminology and text, we extracted a set of 10.000 translations from another slide of the EUROPARL corpus which did not include our 200 test sentences. All sentences had at most 32 words. We did not consider aligned sentences pairs where one language side was

[5]which can be downloaded from http://www.speech.sri.com/projects/srilm/

[6]EUROPARL is available at http://people.csail.mit.edu/koehn/publications/europarl/

[7]Koehn et al (2007) report that loading and decoding times in Moses are much faster than with the SRI tool, due to more compact 8 bit data storage and efficient access to data.

empty. This training set was used to estimate and train weights for our available dictionary entries. A further feature function would then take these weight into account to compute the most likely translations, hence with a broader knowledge.

Weighing of the lexicon is only crucial for ambiguous entries in order to discriminate between different translations for a German SL expression. The weights of all other entries which have one single translation were set to 0.001. For entries with more than one translation option, weights were computed as follows.

For each word in every SL sentence of the 10.000 sentences corpus we looked up the dictionary and checked whether an entry covers a word in the corresponding TL sentence. A hit ***h(g <--> e)*** was assigned for entries where a German word ***g*** matches in the SL side and an English word ***e*** matches in the TL side of an alignment. We count as noise ***n(g <--> e)*** dictionary entries which match a word in the SL but with no realization of the translation ***e*** in the TL side of the alignment. We then sum up hits and noise for all ambiguous entries over all 10.000 reference sentences. Following this, we compute the cumulated hit rate ***H(g)*** for the German SL words, which amounts to summing the hit rate over all translation ambiguities of ***g***.

The weight ***w(g <--> e)*** of a lexicon entry was finally computed as the ratio of the cumulated hit ***H(g)*** of an entry ***g*** divided by the noise of the entry and the number of hits produced by the word g. The weight *w* is thus a number $0 < w <= 1$. It is 1 if an entry has only hits and no other translation option of **g** was seen in the data, ire. if ***H(g)*** equals ***h(g <--> e)***. It is close to 0 if a dictionary entry produces mainly noise according in our data.

4. Evaluation

We started the evaluation experiments with using only one feature function, and then incrementally added further feature functions to see whether and how the system output improves. We started by running a coupe of tests on our first test corpus and on the EUROPARL corpus using an **open token-based** and an **open lemma-based LM**. Second we added various tag language models. The results can be seen in Figures 1 and 2. All figures represent BLEU scores. In Figures 3 and 4 we added the lemmas-tag co-occurrence model (**TTF**) as described above, and a lexical weight function (**Lex**). These results are plotted in Figures 3 and 4.

The 'No Tag LM' graphs (lower lines in Figures 1 to 4) plot the baseline model using only one language model. The left sides of these graphs show performance when using the open token-based LM. The right side shows performance with an open lemma-based LM[8] . The token and lemma models use the same number of 100K, 1M and 2M sentences and *3*, *4*, *5* and *6-grams* and they were generated with the SRI toolkit.

Surprisingly, in Figure 1, for the 200 test sentences, the *3-gram* model based on 2M sentences yields results which are worse than those produced by the *3* and *4-gram* models based on a set of 1M sentences. This is different on the EUROPARL corpus in Figures 2 and 4. The lemma-based models perform consistently better than the token-based models. Best performance is reported with a *3-gram* model based on 2M sentences. For both test sets, the performance is also consistently better when adding a tag model to this baseline. We compared seven tag models, and combined them with the lemma and token models. The tag models were CLAWS5 tags from the BNC using 100K, 1 M and 5M sentences and with $n=\{3,4,5,6\}$. As a tendency, using larger *n* provides in many cases better results than increasing the size of the training corpus. Best results are generated by the *5-gram* models with 100K and 1M sentences. When combining the token /lemma models with the tag models, we tested various weighting distributions. The weight of a feature determines how much this function contributes to the outcome of the ranker. Lower weights would indicate a smaller contribution of that feature function while higher weight would give the feature more importance. We tested approximately 10 different weights for each feature, so that every token-tag and lemma-tag language model combination was tested on roughly 100 different distributions of the feature shares.

[8]The Open lemma-based model is based on lemmatised forms. In contrast to the CMU model the SRI language models used here allow for an unrestricted size of the vocabulary.

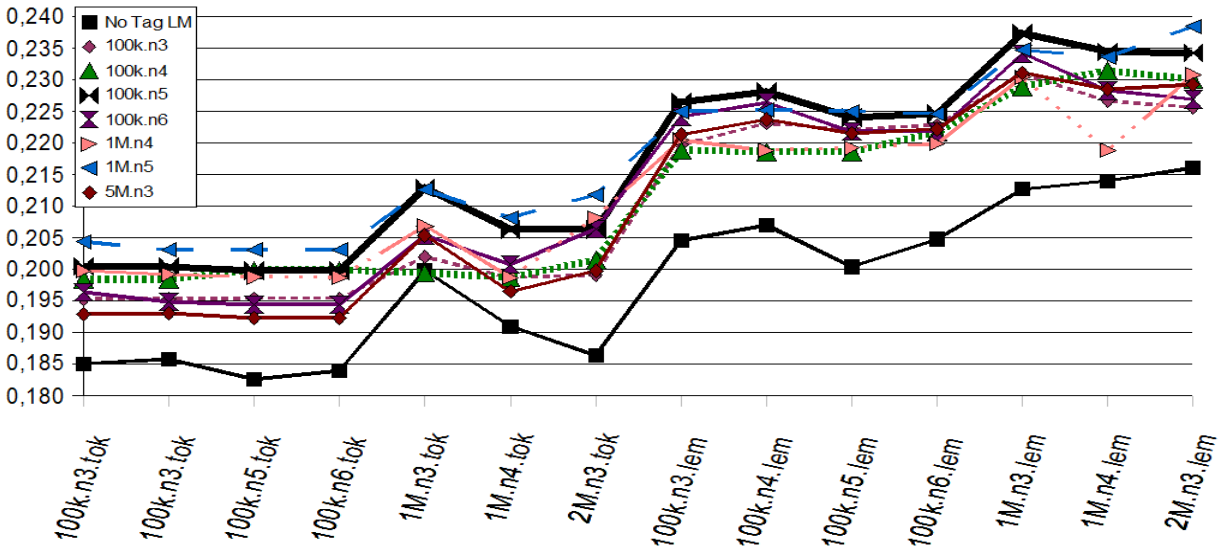

Figure 1: Test set of 200 sentences using various token, lemma and tag language models.

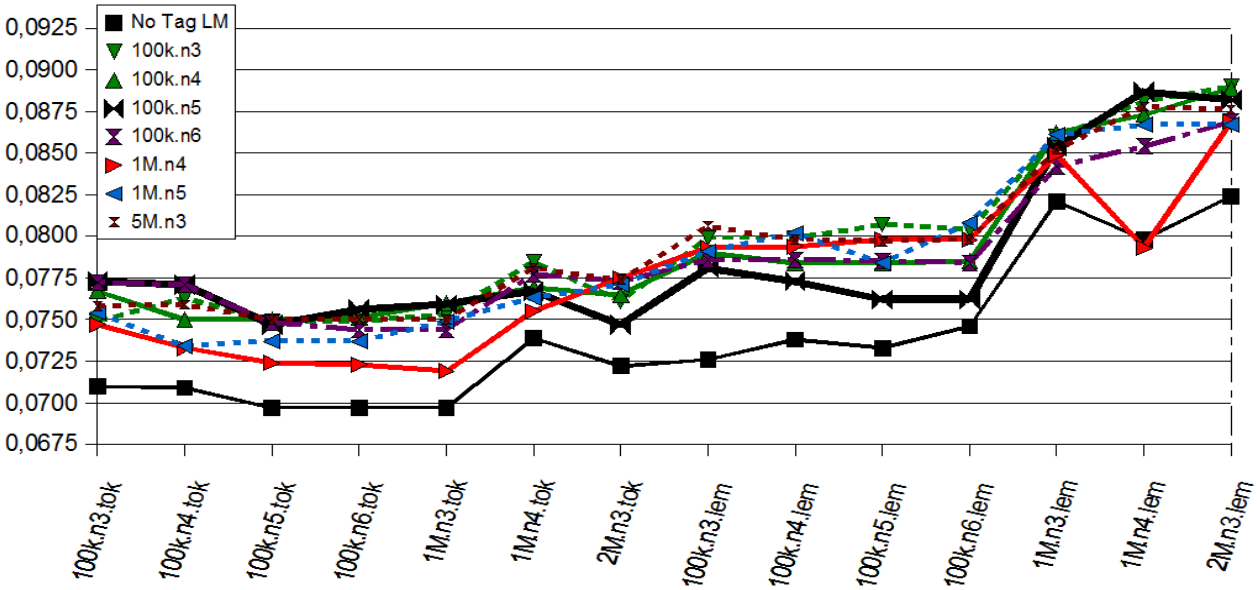

Figure 2: Test set of 200 EUROPARL sentences using various language models.

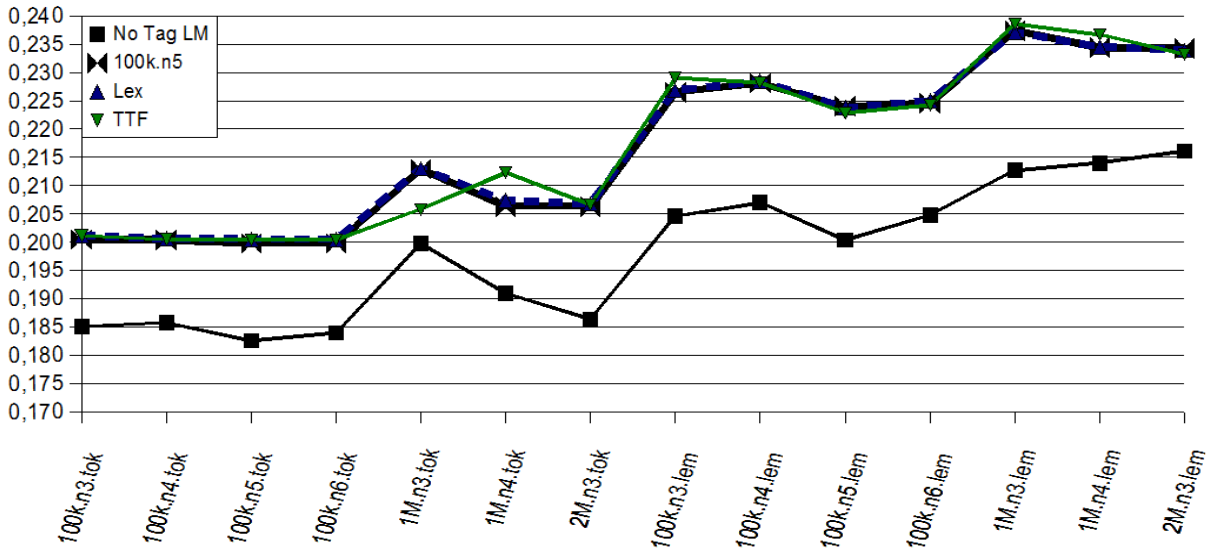

Figure 3: Results for the 200 sentence corpus when adding lexical and token-tag cooccurrence functions.

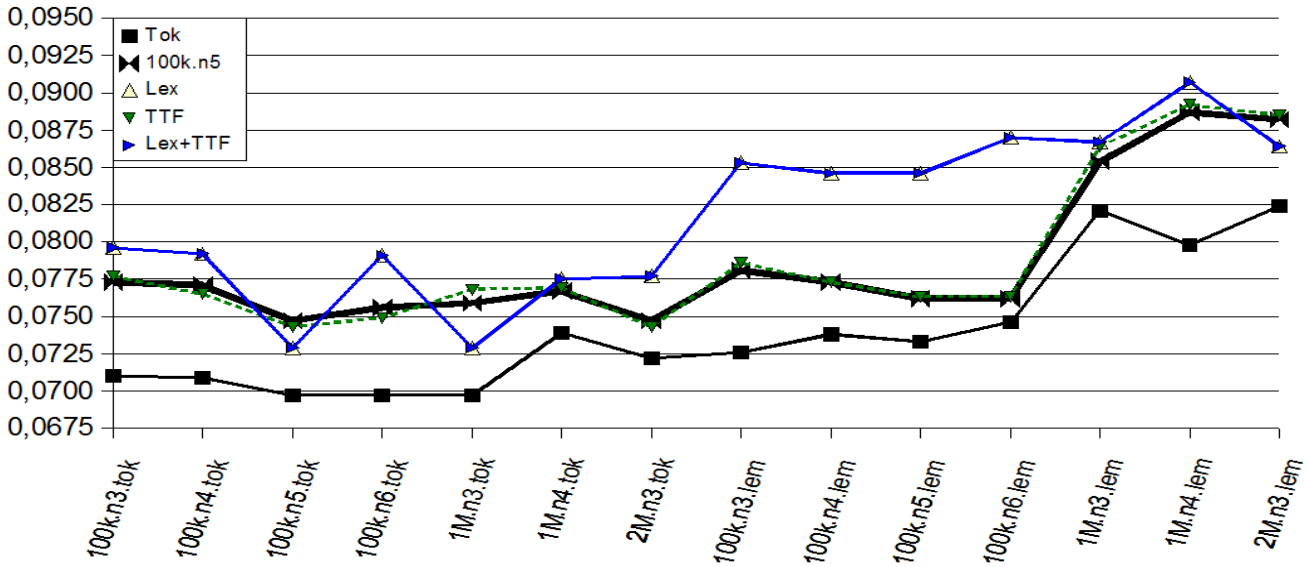

Figure 4: Results for the 200 EUROPARL sentences adding lexical and token-tag cooccurrence functions.

Weights were between *0* and *1.0* and the best results shown in the Figures 1 to 4 use an equal weight of *0.6* for the token, lemma and tag models. Then we added further feature functions to this baseline scenario . Figure 3 repeats the baseline from Figure 1, the 'No Tag LM' and the combination with the 100K, 5-gram tag model. In addition it shows the impact when adding lexical weights and when adding the token-tag co-occurrence function. Here also, we experimented with numerous weighting distributions, but for none of them the overall performance significantly increases. This is different for the EUROPARL test set in Figure 4. Figure 4 also plots the baseline systems from Figure 2, the "No Tag LM" and the combination with the 100K, 5-gram tag model. As can be seen clearly in the graph, when adding the **Lex** function, better results are produced, so that we obtain BLEU values of more than 0.09 on this set. However, here also, the TTF function does not produce any additional improvement.

5. Conclusion

We resume our findings:

1. Lemma-based models produce better results than token-based models. We find that (although not consistently) increasing the size of the training material for lemma models provides better results than increasing the length of the *n-gram* models.
2. Adding a tag model improves the output in any case. Contrary to the findings for the token and lemma models, larger values of *n* (in our case *n=5*) may be an easier way to increase perform than to increase the size of the training set.
3. Adding token-tag co-occurrence statistics as a further feature function does not help.
4. Lexical weights are suitable if the training material is similar to the texts to be translated (i.e. they are from the same domain).

6. References

Allen, Jeffrey and Hogan, Christopher (2000) Toward the Development of a Postediting Module for Raw Machine Translation Output: A Controlled Language Perspective. In *Proceedings of the Third International Workshop on Controlled Language Applications (CLAW00)*. Seattle, Washington: Association for Computational Linguistics, April 29-30, 2000, pp. 62-71.

Carl, Michael (2007) METIS-II: The German to English MT System, In Proceedings of the 11th Machine Translation Summit, Copenhagen, Denmark

Font Llitjós, Ariadna and Carbonell, Jaime G. (2006) Automating Post-Editing to improve MT systems. In *Proc. of the Workshop on Automated Post-Editing at AMTA-2006*, Cambridge, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic, June 2007

Knight, Kevin and Ishwar Chander (1994) Automated Post-editing of Documents, In *Proceedings of the 12th National Conference on Artificial Intelligence*, Seattle, Washington, 779-784.

Patry, Alexandre; Philippe Langlais, Frédéric Béchet (2007), MISTRAL: A Lattice Translation System for IWSLT 2007, In *Proceedings of International Workshop on Spoken Language Translation*, Trento, Italy,
available at: http://iwslt07.itc.it/menu/program.html

Phaholphinyo, Sitthaa, Modhiran, Teerapong, Kritsuthikul, Nattapol and Supnithi, Thepchai (2005) A Practical of Memory-based Approach for Improving Accuracy of MT. In *MT Summit X*. Phuket Island, Thailand.

Povlsen, C. & A. Bech: Ape (2001) Reducing the Monkey Business in Post-Editing by Automating the Task intelligently. In: *MT Summit VIII, Machine Translation in the Information Age*, Santiago de Compostela, Spain, p. 283-286.

Oepen, Stephan and Velldal, Erik and Lonning, Jan Tore and Meurer, Paul and Rosen, Victoria (2007) Towards Hybrid Quality-Oriented Machine Translation -- On

Linguistics and Probabilities in MT, In Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)

Och, Franz J. (2002) Minimum Error Rate Training in Statistical Machine Translation, In Proceeding of ACL,

Liu, Zhanyi and Wang, Haifeng and Wu Hua (2007) Example-Based Machine Translation Based on TSC and Statistical Generation, In Machine Translation (20).

Quirk, Christopher and Menezes, Arul (2007) Dependency Treelet Translation: The convergence of statistical and example-based machine translation?, In Machine Translation (20).