Hybrid Machine Translation Using Joint, Binarised Feature Vectors

Christian Federmann
Language Technology Lab
German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
cfedermann@dfki.de

Abstract

We present an approach for Hybrid Machine Translation, based on a Machine-Learning framework. Our method combines output from several source systems. We first define an extensible, total order on translations and use it to estimate a ranking on the sentence level for a given set of systems. We introduce and define the notion of joint, binarised feature vectors. We train an SVM-based classifier and show how its classification results can be used to create hybrid translations. We describe a series of oracle experiments on data sets from the WMT11 translation task in order to find an upper bound regarding the achievable level of translation quality. We also present results from first experiments with an implemented version of our system. Evaluation using NIST and BLEU metrics indicates that the proposed method can outperform its individual source systems. An interesting finding is that our approach allows to leverage good translations from otherwise bad systems as the translation quality estimation is based on sentence-level phenomena rather than corpus-level metrics. We conclude by summarising our findings and by giving an outlook to future work.

1 Introduction

Machine Translation (MT) is an active research area with many different methods and paradigms, each having individual qualities and shortcomings. The total number of approaches is large, their distinct implementation characteristics and set of individual features are largely diverse.

We briefly describe three of these paradigms and provide a quick comparison between them:

1. *Statistical Machine Translation (SMT):* SMT uses large amounts of parallel data to estimate translation probabilities and works on non-linguistic phrases;

2. *Rule-Based Machine Translation (RBMT):* Rule-based systems utilise hand-crafted parsers and grammars that transform a given sentence into a foreign language translation; and

3. *Hybrid Machine Translation (HMT):* Hybrid approaches aim at creating translations from several source systems. The rationale is that the different MT paradigms' individual strengths and shortcomings are often complementary which implies that a clever combination of their translation output would yield an overall better result.

Regardless of the actual methodology implemented in a given machine translation system, the creation of translations usually involves a lot of—often quite heterogeneous—features, ranging from simple scores, parser confidence estimates, or phrase table probabilities to hierarchical parse trees or even full parse forests. This makes it very difficult to derive an *intuitive understanding* of the inner workings of the MT engine in question; hence, it is evident that research on the optimal combination of different MT systems into improved, hybrid machine translation systems is of utmost importance to the field. In this paper, we investigate a Machine-Learning-based (ML) framework for hybrid machine translation.

The remainder of this paper is structured as follows. After having introduced the research topic in this section, we give a brief overview on related work in Section 2 before defining and explaining in detail our Machine-Learning-based framework for hybrid MT in Section 3. We present the basic method in Section 3.1 before discussing its most important components: the total order on translation output is defined in Section 3.2 while the notion of joint, binarised feature vectors for ML is introduced in Section 3.3. In Section 4 we present the suite of experiments we have conducted; their results are discussed in Section 5. We conclude by giving a summary of our findings and an outlook to future research questions in Section 6.

2 Related Work

Hybrid translation methods and system combination approaches have received a lot of research attention over the last decades. There is general consensus in the literature that it is possible to generate hybrid translations from different systems resulting in an improvement over the individual baseline systems. See seminal work from Frederking and Nirenburg (1994), or, e.g., Macherey and Och (2007) and Rosti et al. (2007).

Confusion Networks can be used for combination (Chen et al., 2007; Eisele et al., 2008; Matusov et al., 2006). One of the MT systems is selected as the *backbone* or *skeleton* of the hybrid translation, while other translations are connected via word alignment techniques such as GIZA++ (Och and Ney, 2003). Together, the systems then form a connected graph in which different paths through the network model different translations. An open-source toolkit for system combination implementing this technique is described in Barrault (2010).

As the combination of translation output using phrase-based methods may not preserve the underlying syntactic structure of the translation backbone, there also are methods which perform *Sentence-based Combination*, trying to select the best option out of a set of several black-box translations for a given source text. The concept is similar to *Re-ranking Approaches* in statistical MT. See research described in Avramidis (2011), Gamon et al. (2005), or Rosti et al. (2007).

Finally, there are methods for *Machine-Learning-based Combination*. These usually train classifiers using, e.g., Support Vector Machines (Vapnik, 1995) to determine if a given translation is good or bad. Recent work such as He et al. (2010a) and He et al. (2010b) applies Machine Learning tools to estimate individual translation quality and re-rank a given set of candidate translations on the sentence level. Of course, there also exist various combinations of the aforementioned methods, e.g., Avramidis (2011) or Okita and van Genabith (2011).

3 Methodology

3.1 Classification-Based Hybrid MT

In this section, we describe a Machine-Learning-based framework for hybrid machine translation. Given a set of n translations from several, black-box systems and a development set including reference text, we perform the following processing steps to produce a hybrid translation for some given test set:

1. Generate a system ranking on the development set using some order relation based on quality assessment of the translations with automatic metrics. This can be extended to also include results from manual evaluation;

2. Decompose this system ranking into a set of pairwise comparisons for any two possible pairs of systems A, B. As we do not allow for ties in our comparisons, the two possible values $A > B$, $A < B$ also represent our *Machine-Learning classes* $+1/-1$, respectively;

3. Annotate the translation output with feature values obtained from NLP tools such as, e.g., *language models*,*part-of-speech taggers*, or *parsers*;

4. Create a data set for training an SVM-based classifier that can estimate which of two given systems A, B is considered *better* according to the available feature values;

5. Train an SVM-based classifier model using, e.g., `libSVM`, see Chang and Lin (2011);

Steps 1–5 represent the *training phase* in our Machine-Learning-based framework.

```
for s_id in 1..len(sentences):
  system_wins = {}
  for (A, B) in system_pairs:
    joint_feature_vector = compute_feature_vector(A, B, s_id)
    classification_result = classify(joint_feature_vector)
    if classification_result == "+1":
      system_wins[A].append(B)
    else:
      system_wins[B].append(A)
  compute_best_system(system_wins)
```

Figure 1: Pseudo-code illustrating how an SVM classifier can be used to determine the single best translation using round robin playoff elimination. This operates on the sentence level, `compute_best_system()` finally computes the system with most "wins" over the competing systems. If two systems A, B have scored the same number of wins, the algorithm falls back to the comparison of these two systems. As we do not allow for ties in our system comparisons, the algorithm is guaranteed to terminate and will always return the single best system for a sentence.

We require the availability of a development set and its reference text to allow the definition of the order relation which subsequently defines the instances for the training of our SVM-based classifier. Once trained, we can use the classifier as follows:

6. Apply the resulting classification model onto the candidate translations from the given test set. This will generate pairwise estimates $+1/-1$ for each combination of systems A, B;
7. Perform *round-robin playoff* elimination to find the single best system from the given set of candidate translations on the sentence level;
8. Synthesise the final, hybrid translation output.

Steps $6-8$ represent the *decoding phase* in which the trained classifier is applied to a set of *unseen* translations without any reference text available. By computing pairwise *winners* for each possible pair of systems and each individual sentence of the test set, we determine the single best translation on the sentence level. By combining these, we generate a hybrid translation improving over the single best baseline system.

As our methodology allows to also integrate good parts from otherwise bad systems, we expect the approach to improve over the individual baseline systems—this claim will receive more attention in our experiments described in Section 4.

We need to find solutions for two very important prerequisites in order to successfully implement the proposed combination method:

- We must define a *sufficiently good* order; and
- We need to *convert* this order, using Machine Learning tools, into an equivalent classification model.

We address both issues in the following sections.

3.2 An Extensible, Total Order on Translations

In order to rank the given source translations, we first need to define an *ordering relation* over the set of translation outputs. For this, we apply three renowned MT evaluation metrics which are the *de-facto standards* for automated assessment of MT quality. We consider:

1. The Meteor score, both on the sentence and on the corpus level, see Denkowski and Lavie (2011);
2. The NIST n-gram co-occurence score on the corpus level, see Doddington (2002); and
3. The BLEU score which is the most widely used evaluation metric, see Papineni et al. (2002).

While both BLEU and NIST scores are designed to have a high correlation with results from manual evaluation on the corpus level (denoted by suffix $_C$), the Meteor metric can also be used to meaningfully compare translations down to the level of individual sentences (denoted by suffix $_S$). We make use of this property when defining our order $ord(A, B)$ on translations, as shown in equations 3–7 on Page 9.

Note that the our order $ord(A, B)$ is an *extensible, total order*. It can easily be extended to include, e.g., results from manual evaluation of translation output. In fact, this would be a helpful addition as it would allow to bring in knowledge from domain experts. However, as a manual annotation campaign for n systems is both very time-consuming and expensive, we leave the integration of manual judgements into our ordering relation to future work.

3.3 Joint, Binarised Feature Vectors

As previously mentioned in Section 2, many Machine-Learning-based approaches for system combination use classifiers to estimate the quality or *confidence* in an individual translation output and compare it to other translations afterwards. This means that the feature vector for a given translation A is computed solely on information available in A, not considering any other translation B. Formally, we define $vec_{single}(A) \in \mathbb{R}^n$ as follows:

$$vec_{single}(A) \overset{\text{def}}{=} \begin{pmatrix} f_1(A) \\ f_2(A) \\ \vdots \\ f_n(A) \end{pmatrix} \quad (1)$$

We take a different approach and compute feature vectors for all possible, *pairwise comparisons* of translations A, B, storing *binary feature values* to model if a given feature value $f_x(A)$ for system A is better or worse than the corresponding feature value $f_x(B)$ for the competing system B.

Effectively, this means that we *directly* model the comparison between translations when constructing the set of feature vectors required for training our ML classifier. Equation 2 shows our definition of a *joint, binarised* feature $vec_{joint}(A, B) \in \mathbb{B}^n$:

$$vec_{joint}(A, B) \overset{\text{def}}{=} \begin{pmatrix} f_1(A) > f_1(B) \\ f_2(A) > f_2(B) \\ \vdots \\ f_n(A) > f_n(B) \end{pmatrix} \quad (2)$$

The reason to store binary features values $f_x \in \mathbb{B}$ lies in the fact that these can be handled in a more efficient way during SVM training. Also, previous experiments have shown that using the actual feature values $f_x \in \mathbb{R}$ does not give any additional benefit. Hence, we decided to use binary notation instead. Note that the order in which features for translations A, B are compared does not strictly matter. For the sake of consistency, we decided to compare feature values using simple $A > B$ operations, leaving the actual interpretation of these values or their polarity to the Machine Learning toolkit.

3.4 Feature Set for Training a Binary Classifier

We create the data set for classifier training using a selection of *features*. While there are many possible features which could be added to this feature set, we restricted ourselves to the following choice, leaving changes to future work:

- number of target tokens;
- ratio of target/source tokens;
- number of target parse tree nodes;
- ratio of target/source parse tree nodes;
- number of target parse tree depth;
- ratio of target/source parse tree depth;
- n-gram score for order $n \in \{1, \dots, 5\}$;
- perplexity for order $n \in \{1, \dots, 5\}$.

These features represent a combination of (shallow) parsing and language model scoring and are derived from the set of features that are most often used in the Machine-Learning-based system combination literature (Avramidis, 2011; Gamon et al., 2005; He et al., 2010a; He et al., 2010b; Okita and van Genabith, 2011).

3.5 Hybrid MT Using an SVM Classifier

Given an SVM classification model trained on joint, binary feature vectors as previously described, we can now create hybrid translation output. The basic algorithm is depicted in Figure 1. It estimates the single best translation for each sentence in the test set, based on the $+1/-1$ output of the classifier.

For each sentence, we create a dictionary that stores for some system X the set of systems which were outperformed by X according to the results from our classifier. We consider each possible comparison of systems A, B and compute the corresponding feature vector which is classified by the SVM. Only systems winning at least once in these pairwise comparisons end up as keys in our dictionary. The cardinality of the set of systems outperformed by a system X implicitly represents the number of wins for this system. We determine the single best translation for a sentence by sorting the `system_wins` dictionary so that systems with a larger number of wins come first. There are three cases to consider:

1. If there exists only one top-ranked system, this is selected as the winning translation for the current sentence;

2. If two systems are top-ranked, the winner only depends on the comparison of these two. As we do not allow for ties in our comparisons, this is guaranteed to determine a single winner;

3. If more than two systems are top-ranked, we check if one of the systems outperforms the others. This may not yield a unique winner, in which case we fall back to scoring the top-ranked systems with $ord_{Meteor_C}(A, B)$, using the corpus-level system rankings obtained on the development set to reach a final decision on the best translation for the current sentence.

We investigate the performance of our methodology in the following section.

4 Experiments

4.1 Oracle Experiments

Before working on the fine-tuning of the Machine Learning training method, we want to investigate more closely what improvements can be achieved using our method. Hence, we perform a series of *oracle experiments* in which we simulate a perfect classifier resulting in an optimal selection of hybrid sentences and, thus, an optimal hybrid translation.

4.2 Experimental Setup

We make use of the official results from the WMT11 (Callison-Burch et al., 2011), including translation output, automatic metrics' scores, and also results from human judgement. We focus on systems from two special groups, namely *rule-based systems* and *online translation engines*.

We compute what our method—given a perfect selection on the sentence level—can create given black-box translation output from 1) rule-based MT systems, 2) online translators, or 3) both data sets. We conduct the experiments for all language pairs involving English, German, Spanish, and French, a total of six directions from WMT11's translation task. Results from the oracle experiments are given in Table 3 on Page 10.

4.3 Prototypical Experiments

In order to assess the performance of the proposed approach on real data, we also run experiments on a prototypical implementation of the approach and measure the resulting translation quality. Note that in the data sets used for experimentation individual system names are anonymised as the translations are part of a shared task we participated in to test our method. We train SVM classifiers for two language pairs: Arabic→English and Chinese→English. For the first pair we work on translation output generated by $n = 10$ different systems, for the latter pair there are $n = 15$ systems to consider. The source text originates from the news domain.

4.4 Training Data

As training data, we receive a development set with references as well as a test set without reference. Applying our total order on the given translations, we determine a system ranking on the sentence level. We compute pairwise system comparisons for usage as SVM class labels and annotate each individual translation with parser output and language model scores. We use the Stanford Parser (Green and Manning, 2010; Klein and Manning, 2003; Levy and Manning, 2003) to process the source text and the corresponding translations. For language model scoring, we use the SRILM toolkit (Stolcke, 2002) training a 5-gram language model for English. In this work, we do not consider any source language language models.

4.5 Classifier Training

Figure 3 shows the optimisation grids we obtained during SVM tuning. They show which settings for C and γ result in the best prediction rate. Note how the graphs are similar regarding the *optimal area*. We train our final SVM classifiers with parameters from this area, giving preference to smaller values for both C and γ to reduce computational cost and thus training time.

Still, the overall prediction rate achieved with the aforementioned set of feature vectors is sub-optimal and needs to be improved in future work. We have observed promising improvements with re-scoring of translation output using a shared phrase table and will investigate this further in upcoming work.

5 Evaluation

5.1 Oracle Experiments

The central result from our oracle experiments is that our proposed approach can outperform individual baseline systems for a) each set of systems, and b) every language pair under investigation. Results are presented in Table 3 on Page 10.

The big challenge for future research lies in the definition of a set of features which allows to train a classifier that can estimate our total order relation with a very high probability. The better the final prediction rate of our classification model, the closer we can get to the upper bound of achievable quality.

5.2 Prototypical Experiments

In order to investigate the quality of the described Machine-Learning-based approach for real data, we evaluated the translation quality of our hybrid MT system (referred to as *SVM-combo* in Tables 1 and 2) using BLEU and NIST, both of which are predominantly used in machine translation literature and evaluation campaigns.

Table 1 shows that our hybrid translation output improved over the single best baseline system for language pair Arabic→English, both in terms of NIST score as well as for the BLEU metric. Note that we are listing more than $n = 10$ systems here as not all participating systems from the shared task agreed to be part of the system combination task.

Interestingly, results from language pair Chinese→English are different, as depicted in Table 2. Here, our hybrid translation only achieves $8th$ rank wrt. NIST score and $4th$ rank in terms of BLEU. In summary, it seems we need a refined set of features to improve for this language pair.

Arabic→English		
System	NIST Score	BLEU Score
sys #1	10.1578	0.4300
sys #2	10.0379	0.4251
sys #3	9.8845	0.4179
sys #4	9.8841	0.4132
sys #5	9.8675	0.4109
sys #6	9.8408	0.4070
sys #7	9.7120	0.4012
sys #8	9.6853	0.3996
sys #9	9.6417	0.3982
sys #10	9.4226	0.3799
sys #11	8.5721	0.3160
sys #12	8.1091	0.2746
SVM-combo	1st **10.3584**	1st **0.4523**

Table 1: Translation quality measured using NIST and BLEU scores for language pair Arabic→English. Note how our *SVM-combo* system is able to outperform the individual baseline systems for both metrics.

Chinese→English		
System	NIST Score	BLEU Score
sys #1	**8.6996**	**0.3044**
sys #2	8.4245	0.2927
sys #3	8.1160	0.2813
sys #4	8.0534	0.2795
sys #5	7.9788	0.2587
sys #6	7.8969	0.2545
sys #7	7.7679	0.2518
sys #8	7.6965	0.2369
sys #9	7.6461	0.2489
sys #10	7.5181	0.2265
sys #11	7.4819	0.2580
sys #12	7.4045	0.2276
sys #13	7.2969	0.2472
sys #14	6.8456	0.1957
sys #15	6.2852	0.1497
sys #16	6.1679	0.1867
SVM-combo	8th 7.7636	4th 0.2663

Table 2: Translation quality measured using NIST and BLEU scores for language pair Chinese→English. Here, our *SVM-combo* system only achieves $8th$ rank in terms of NIST score, $4th$ rank according to the BLEU metric.

6 Conclusion

6.1 Summary of Findings

We have described a Machine-Learning-based framework for hybrid MT. We explained how a total order on translation output can be defined and used for feature vector generation. Our method differs from previous work as we consider joint, binarised feature vectors instead of separate feature vectors for each of the available source systems. We proposed an algorithm to make use of a classification model trained on these feature vectors for the creation of hybrid translations.

We reported on a series of oracle experiments in which we simulated optimal translation selection on the sentence level. In summary, these experiments illustrated what translation quality could be achieved using the proposed method. Notably, our approach outperformed the individual baseline systems for all language pairs under investigation.

In our experiments with real data for language pairs Arabic→English and Chinese→English, we observed improvements in terms of both NIST and BLEU scores for the first and mixed results for the latter language pair. We found that our classifier was able to make use of good translation output from systems which performed bad on the corpus level.

6.2 Outlook on Future Work

The total order on translation defined in Section 3 can be extended to include results from manual judgements regarding the quality of translations from candidate systems. It will be interesting to get a better understanding how such a refined order relation impacts the training of the SVM classifier and the quality of the hybrid translation output.

Our classifier can also be changed to generate probabilistic estimates $p \in \mathbb{R}$ instead of absolute class values $+1/-1$. This would give us a parameter to tune the classifier for a better prediction rate on the development set. It remains to be investigated how such a classification scheme would affect the selection of translations on the sentence level and the quality of the final, hybrid output.

While it is clear that the classification model's prediction rate and the quality of the resulting translation output are strongly interrelated, their connection needs to be investigated in more detail to find out how changes of the former would affect and alter the latter.

Finally, we intend to conduct a manual evaluation campaign to investigate in more detail the selection quality of our combination method and to find out if the selected translations are actually the *best* (or at least *good*) translations considering the given source sentence. We look forward to interesting challenges and findings in the future.

Acknowledgments

This work has been funded under the Seventh Framework Programme for Research and Technological Development of the European Commission through the T4ME contract (grant agreement no.: 249119). We are grateful to the anonymous reviewers for their valuable feedback.

References

Eleftherios Avramidis. 2011. DFKI System Combination with Sentence Ranking at ML4HMT-2011. In *Proceedings of the International Workshop on Using Linguistic Information for Hybrid Machine Translation (LIHMT 2011) and of the Shared Task on Applying Machine Learning Techniques to Optimise the Division of Labour in Hybrid Machine Translation (ML4HMT)*, Barcelona, Spain, November.

Loic Barrault. 2010. Many: Open source machine translation system combination. *Prague Bulletin of Mathematical Linguistics, Special Issue on Open Source Tools for Machine Translation*, 1(93):145–155.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Yu Chen, Andreas Eisele, Christian Federmann, Eva Hasler, Michael Jellinghaus, and Silke Theison. 2007. Multi-engine machine translation with an open-source SMT decoder. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 193–196, Prague, Czech Republic, June. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and

Evaluation of Machine Translation Systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, July. Association for Computational Linguistics.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using n-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Andreas Eisele, Christian Federmann, Hervé Saint-Amand, Michael Jellinghaus, Teresa Herrmann, and Yu Chen. 2008. Using Moses to integrate multiple rule-based machine translation engines into a hybrid system. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 179–182, Columbus, Ohio, June. Association for Computational Linguistics.

Robert Frederking and Sergei Nirenburg. 1994. Three Heads are Better Than One. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, ANLC '94, pages 95–100, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level MT Evaluation Without Reference Translations: Beyond Language Modeling. In *Proceedings of the 10th EAMT Conference "Practical applications of machine translation"*, pages 103–111. European Association for Machine Translation, May.

Spence Green and Christopher D. Manning. 2010. Better Arabic Parsing: Baselines, Evaluations, and Analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 394–402, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. 2010a. Bridging SMT and TM with Translation Recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 622–630, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yifan He, Yanjun Ma, Andy Way, and Josef van Genabith. 2010b. Integrating N-best SMT Outputs into a TM System. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 374–382, Stroudsburg, PA, USA. Association for Computational Linguistics.

D. Klein and C. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, volume 1 of *ACL '03*, pages 423–430, Stroudsburg, PA, USA, July. Association for Computational Linguistics.

Roger Levy and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, ACL '03, pages 439–446, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wolfgang Macherey and Franz J. Och. 2007. An Empirical Study on Computing Consensus Translations from Multiple Machine Translation Systems. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 986–995, Prague, Czech Republic, June. Association for Computational Linguistics.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing Consensus Translation from Multiple Machine Translation Systems Using Enhanced Hypotheses Alignment. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–40, Stroudsburg, PA, USA, April. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, March.

Tsuyoshi Okita and Josef van Genabith. 2011. DCU Confusion Network-based System Combination for ML4HMT. In *Proceedings of the International Workshop on Using Linguistic Information for Hybrid Machine Translation (LIHMT 2011) and of the Shared Task on Applying Machine Learning Techniques to Optimise the Division of Labour in Hybrid Machine Translation (ML4HMT)*, Barcelona, Spain, November.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007. Combining Outputs from Multiple Machine Translation Systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 228–235, Rochester, New York, April. Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 257–286, November.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.

$$ord_{BLEU_C}(A,B) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } A_{BLEU_C} > B_{BLEU_C} \\ -1 & \text{if } A_{BLEU_C} < B_{BLEU_C} \\ 0 & \text{else} \end{cases} \tag{3}$$

$$ord_{NIST_C}(A,B) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } A_{NIST_C} > B_{NIST_C} \\ -1 & \text{if } A_{NIST_C} < B_{NIST_C} \\ ord_{BLEU_C}(A,B) & \text{else} \end{cases} \tag{4}$$

$$ord_{Meteor_C}(A,B) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } A_{Meteor_C} > B_{Meteor_C} \\ -1 & \text{if } A_{Meteor_C} < B_{Meteor_C} \\ ord_{NIST_C}(A,B) & \text{else} \end{cases} \tag{5}$$

$$ord_{Meteor_S}(A,B) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } A_{Meteor_S} > B_{Meteor_S} \\ -1 & \text{if } A_{Meteor_S} < B_{Meteor_S} \\ ord_{Meteor_C}(A,B) & \text{else} \end{cases} \tag{6}$$

$$ord(A,B) \overset{\text{def}}{=} ord_{Meteor_S}(A,B) \tag{7}$$

Figure 2: Definition of a total order on translation output. Each pair of translations A, B is compared using a) sentence-based (denoted by suffix $_S$) Meteor scores, and b) corpus-based (denoted by suffix $_C$) Meteor, NIST, and BLEU scores. Meteor is preferred as previous experiments have shown that it has a better correlation with human assessments than both NIST and BLEU. Conflict resolution in case of $A_{BLEU_C} = B_{BLEU_C}$ is described in Section 3.5.

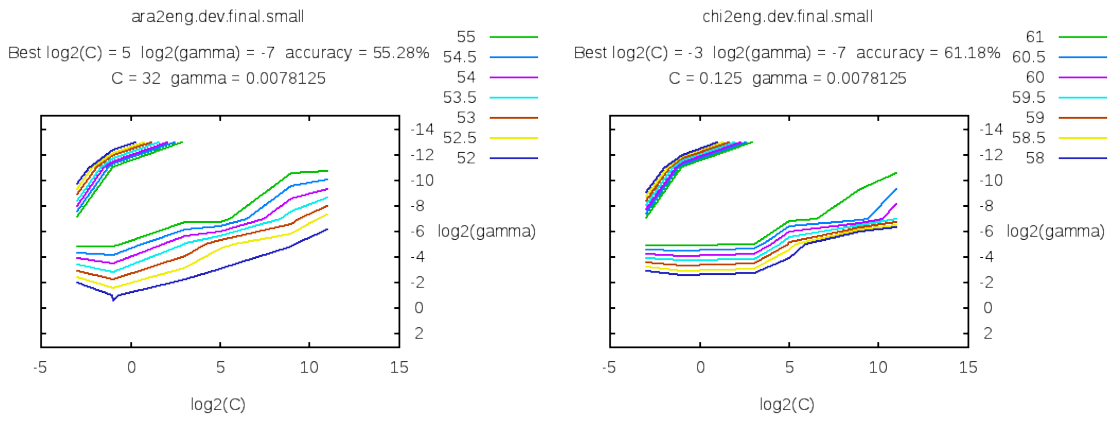

Figure 3: Optimisation grids of SVM parameters C and γ obtained during tuning for language pairs Arabic→English and Chinese→English. Note the similarity of the grids, indicating that our selected set of feature vectors is language independent. Also note the low values for prediction rate, signaling that further research on helpful features needs to be undertaken to achieve hybrid translation quality comparable to the theoretical upper bound as reported in the oracle experiments on WMT11 translation task data.

	RBMT		Online		Combined	
	System	Score	System	Score	System	Score
English→German	rbmt-1	0.35147	online-A	0.36535	rbmt-combo	0.40279
	rbmt-2	0.33472	online-B	0.38436	online-combo	0.40845
	rbmt-3	0.35527			full-combo	0.43591
	rbmt-4	0.34430				
	rbmt-5	0.33935				
German→English	rbmt-1	0.29265	online-A	0.31352	rbmt-combo	0.33519
	rbmt-2	0.28089	online-B	0.31726	online-combo	0.35176
	rbmt-3	0.29429			full-combo	0.36549
	rbmt-4	0.29214				
	rbmt-5	0.28131				
English→Spanish	rbmt-1	0.50936	online-A	0.55937	rbmt-combo	0.55667
	rbmt-2	0.49807	online-B	0.56629	online-combo	0.57535
	rbmt-3	0.51901			full-combo	0.59962
	rbmt-4	0.50729				
	rbmt-5	0.49491				
Spanish→English	rbmt-1	0.34027	online-A	0.34600	rbmt-combo	0.37601
	rbmt-2	0.32835	online-B	0.34099	online-combo	0.37719
	rbmt-3	0.33177			full-combo	0.40510
	rbmt-4	0.33470				
	rbmt-5	0.33166				
English→French	rbmt-1	0.49596	online-A	0.51778	rbmt-combo	0.54125
	rbmt-2	0.47306	online-B	0.55774	online-combo	0.56825
	rbmt-3	0.49819			full-combo	0.59366
	rbmt-4	0.47064				
	rbmt-5	0.48620				
French→English	rbmt-1	0.33422	online-A	0.34434	rbmt-combo	0.37076
	rbmt-2	0.31792	online-B	0.34529	online-combo	0.38272
	rbmt-3	0.32414			full-combo	0.40156
	rbmt-4	0.32564				
	rbmt-5	0.32118				

Table 3: Individual performance in terms of Meteor scores for rule-based and online system submissions in the WMT11 translation task for all language pairs involving English, German, French, and Spanish. We compare these scores to the oracle scores for three data sets: *rbmt*, *online*, and *full* which contains both rbmt and online. Note that, theoretically, each data set can be improved over the single best baseline score, provided system selection performs sufficiently well. The scores reported in this table represent the theoretical upper bound of achievable translation quality for translation output from participating systems of the WMT11 translation tasks.