Learning Performance of a Machine Translation System: a Statistical and Computational Analysis

Marco Turchi **Tijl De Bie** **Nello Cristianini**
Dept. of Engineering Mathematics
University of Bristol,
Bristol, BS8 1TR, UK
{Marco.Turchi, Tijl.DeBie}@bristol.ac.uk
nello@support-vector.net

Abstract

We present an extensive experimental study of a Statistical Machine Translation system, Moses (Koehn et al., 2007), from the point of view of its learning capabilities. Very accurate learning curves are obtained, by using high-performance computing, and extrapolations are provided of the projected performance of the system under different conditions. We provide a discussion of learning curves, and we suggest that: 1) the representation power of the system is not currently a limitation to its performance, 2) the inference of its models from finite sets of i.i.d. data is responsible for current performance limitations, 3) it is unlikely that increasing dataset sizes will result in significant improvements (at least in traditional i.i.d. setting), 4) it is unlikely that novel statistical estimation methods will result in significant improvements. The current performance wall is mostly a consequence of Zipf's law, and this should be taken into account when designing a statistical machine translation system. A few possible research directions are discussed as a result of this investigation, most notably the integration of linguistic rules into the model inference phase, and the development of active learning procedures.

1 Introduction and Background

The performance of every learning system is the result of (at least) two combined effects: the representation power of the hypothesis class, determining how well the system can approximate the target behaviour; and statistical effects, determining how well the system can approximate the best element of the hypothesis class, based on finite and noisy training information. The two effects interact, with richer classes being better approximators of the target behaviour but requiring more training data to reliably identify the best hypothesis. The resulting trade-off, equally well known in statistics and in machine learning, can be expressed in terms of bias variance, capacity-control, or model selection. Various theories on learning curves have been proposed to deal with it, where a learning curve is a plot describing performance as a function of some parameters, typically training set size.

In the context of Statistical Machine Translation (SMT), where large bilingual corpora are used to train adaptive software to translate text, this task is further complicated by the peculiar distribution underlying the data, where the probability of encountering new words or expressions never vanishes. If we want to understand the potential and limitations of the current technology, we need to understand the interplay between these two factors affecting performance. In an age where the creation of intelligent behaviour is increasingly data driven, this is a question of great importance to all of Artificial Intelligence.

These observations lead us to an analysis of learning curves in machine translation, and to a number of related questions, including an analysis of the flexibility of the representation class used, an analysis of the stability of the models with respect to perturbations of the parameters, and an analysis of the computational resources needed to train these systems.

Using the open source package Moses (Koehn et

Proceedings of the Third Workshop on Statistical Machine Translation, pages 35–43,
Columbus, June 2008.

al., 2007) and the Spanish-English Europarl corpus (Koehn, 2005) we have performed a complete investigation of the influence of training set size on the quality of translations and on the cost of training; the influence of several design choices; the role of data sizes in training various components of the system. We use this data to inform a discussion about learning curves. An analysis of learning curves has previously been proposed by (Al-Onaizan et al., 1999). Recent advances in software, data availability and computing power have enabled us to undertake the present study, where very accurate curves are obtained on a large corpus.

Since our goal was to obtain high accuracy learning curves, that can be trusted both for comparing different system settings, and to extrapolate performance under unseen conditions, we conducted a large-scale series of tests, to reduce uncertainty in the estimations and to obtain the strongest possible signals. This was only possible, to the degree of accuracy needed by our analysis, by the extensive use of a high performance computer cluster over several weeks of computation.

One of our key findings is that the current performance is not limited by the representation power of the hypothesis class, but rather by model estimation from data. And that increasing of the size of the dataset is not likely to bridge that gap (at least not for realistic amounts in the i.i.d. setting), nor is the development of new parameter estimation principles. The main limitation seems to be a direct consequence of Zipf's law, and the introduction of constraints from linguistics seems to be an unavoidable step, to help the system in the identification of the optimal models without resorting to massive increases in training data, which would also result in significantly higher training times, and model sizes.

2 Statistical Machine Translation

What is the best function class to map Spanish documents into English documents? This is a question of linguistic nature, and has been the subject of a long debate. The de-facto answer came during the 1990's from the research community on Statistical Machine Translation, who made use of statistical tools based on a noisy channel model originally developed for speech recognition (Brown et al., 1994; Och and Weber, 1998; R.Zens et al., 2002; Och and Ney, 2001; Koehn et al., 2003). A Markovian language model, based on phrases rather than words, coupled with a phrase-to-phrase translation table are at the heart of most modern systems. Translating a text amounts to computing the most likely translation based on the available model parameters. Inferring the parameters of these models from bilingual corpora is a matter of statistics. By model inference we mean the task of extracting all tables, parameters and functions, from the corpus, that will be used to translate.

How far can this representation take us towards the target of achieving human-quality translations? Are the current limitations due to the approximation error of this representation, or to lack of sufficient training data? How much space for improvement is there, given new data or new statistical estimation methods or given different models with different complexities?

We investigate both the approximation and the estimation components of the error in machine translation systems. After analysing the two contributions, we focus on the role of various design choices in determining the statistical part of the error. We investigate learning curves, measuring both the role of the training set and the optimization set size, as well as the importance of accuracy in the numeric parameters.

We also address the trade-off between accuracy and computational cost. We perform a complete analysis of Moses as a learning system, assessing the various contributions to its performance and where improvements are more likely, and assessing computational and statistical aspects of the system.

A general discussion of learning curves in Moses-like systems and an extrapolation of performance are provided, showing that the estimation gap is unlikely to be closed by adding more data in realistic amounts.

3 Experimental Setup

We have performed a large number of detailed experiments. In this paper we report just a few, leaving the complete account of our benchmarking to a full journal version (Turchi et al., In preparation). Three experiments allow us to assess the most promising directions of research, from a machine learning point of view.

1. Learning curve showing translation performance as a function of training set size, where translation is performed on unseen sentences. The curves, describing the statistical part of the performance, are seen to grow very slowly with training set size.

2. Learning curve showing translation performance as a function of training set size, where translation is performed on known sentences. This was done to verify that the hypothesis class is indeed capable of representing high quality translations in the idealized case when all the necessary phrases have been observed in training phase. By limiting phrase length to 7 words, and using test sentences mostly longer than 20 words, we have ensured that this was a genuine task of decoding. We observed that translation in these idealized conditions is worse than human translation, but much better than machine translation of unseen sentences.

3. Plot of performance of a model when the numeric parameters are corrupted by an increasing amount of noise. This was done to simulate the effect of inaccurate parameter estimation algorithms (due either to imprecise objective functions, or to lack of sufficient statistics from the corpus). We were surprised to observe that accurate estimation of these parameters accounts for at most 10% of the final score. It is the actual list of phrases that forms the bulk of the knowledge in the system.

We conclude that the availability of the right models in the system would allow the system to have a much higher performance, but these models will not come from increased datasets or estimation procedures. Instead, they will come from the results of either the introduction of linguistic knowledge, or the introduction of query algorithms, themselves resulting necessarily from confidence estimation methods. Hence these appear to be the two most pressing questions in this research area.

3.1 Software

Moses (Koehn et al., 2007) is a complete translation toolkit for academic purposes. It provides all the components needed to create a machine translation system from one language to another. It contains different modules to preprocess data, train the language models and the translation models. These models can be tuned using minimum error rate training (Och, 2003). Moses uses standard external tools for some of these tasks, such as GIZA++ (Och and Ney, 2003) for word alignments and SRILM (Stolcke, 2002) for language modeling. Notice that Moses is a very sophisticated system, capable of learning translation tables, language models and decoding parameters from data. We analyse the contribution of each component to the overall score.

Given a parallel training corpus, Moses preprocesses it removing long sentences, lowercasing and tokenizing sentences. These sentences are used to train the language and translation models. This phase requires several steps as aligning words, computing the lexical translation, extracting phrases, scoring the phrases and creating the reordering model. When the models have been created, the development set is used to run the minimum error rate training algorithm to optimize their weights. We refer to that step as the optimization step in the rest of the paper. Test set is used to evaluate the quality of models on the data. The translated sentences are embedded in a sgm format, such that the quality of the translation can be evaluated using the most common machine translation scores. Moses provides BLEU (K.Papineni et al., 2001) and NIST (Doddington, 2002), but Meteor (Banerjee and Lavie, 2005; Lavie and Agarwal, 2007) and TER (Snover et al., 2006) can easily be used instead. NIST is used in this paper as evaluation score after we observed its high correlation to the other scores on the corpus (Turchi et al., In preparation).

All experiments have been run using the default parameter configuration of Moses. It means that Giza++ has used IBM model 1, 2, 3, and 4 with number of iterations for model 1 equal to 5, model 2 equal to 0, model 3 and 4 equal to 3; SRILM has used n-gram order equal to 3 and the Kneser-Ney smoothing algorithm; Mert has been run fixing to 100 the number of nbest target sentence for each develop sentence, and it stops when none of the weights changed more than 1e-05 or the nbest list does not change.

The training, development and test set sentences are tokenized and lowercased. The maximum number of tokens for each sentence in the training pair has been set to 50, whilst no limit is applied to the development or test set. TMs were limited to a phrase-length of 7 words and LMs were limited to 3.

3.2 Data

The Europarl Release v3 Spanish-English corpus has been used for the experiments. All the pairs of sentences are extracted from the proceedings of the European Parliament.

This dataset is made of three sets of pairs of sentences. Each of them has a different role: *training*, *development* and *test* set. The training set contains 1,259,914 pairs, while there are 2,000 pairs for development and test sets.

This work contains several experiments on different types and sizes of data set. To be consistent and to avoid anomalies due to overfitting or particular data combinations, each set of pairs of sentences have been randomly sampled. The number of pairs is fixed and a software selects them randomly from the whole original training, development or test set using a uniform distribution (bootstrap). Redundancy of pairs is allowed inside each subset.

3.3 Hardware

All the experiments have been run on a cluster machine, http://www.acrc.bris.ac.uk/acrc/hpc.htm. It includes 96 nodes each with two dual-core opteron processors, 8 GB of RAM memory per node (2 GB per core); 4 thick nodes each with four dual-core opteron processors, 32 GB of RAM memory per node (4 GB per core); ClearSpeed accelerator boards on the thick nodes; SilverStorm Infiniband high-speed connectivity throughout for parallel code message passing; General Parallel File System (GPFS) providing data access from all the nodes; storage - 11 terabytes. Each experiment has been run using one core and allocating 4Gb of RAM.

4 Experiments

4.1 Experiment 1: role of training set size on performance on new sentences

In this section we analyse how performance is affected by training set size, by creating learning curves (NIST score vs training set size).

We have created subsets of the complete corpus by sub-sampling sentences from a uniform distribution, with replacement. We have created 10 random subsets for each of the 20 chosen sizes, where each size represents 5%, 10%, etc of the complete corpus. For each subset a new instance of the SMT system has been created, for a total of 200 models. These have been optimized using a fixed size development set (of 2,000 sentences, not included in any other phase of the experiment). Two hundred experiments have then been run on an independent test set (of 2,000 sentences, also not included in any other phase of the experiment). This allowed us to calculate the mean and variance of NIST scores. This has been done for the models with and without the optimization step, hence producing the learning curves with error bars plotted in Figure 1, representing translation performance versus training set size, in the two cases.

The growth of the learning curve follows a typical pattern, growing fast at first, then slowing down (traditional learning curves are power laws, in theoretical models). In this case it appears to be growing even slower than a power law, which would be a surprise under traditional statistical learning theory models. In any case, the addition of massive amounts of data from the same distribution will result into smaller improvements in the performance. The small error bars that we have obtained also allow us to neatly observe the benefits of the optimization phase, which are small but clearly significant.

4.2 Experiment 2: role of training set size on performance on known sentences

The performance of a learning system depends both on the statistical estimation issues discussed in the previous subsection, and on functional approximation issues: how well can the function class reproduce the desired behaviour? In order to measure this quantity, we have performed an experiment much like the one described above, with one key differ-

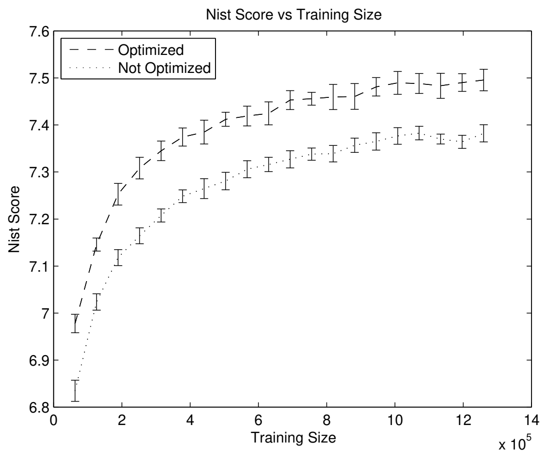

Figure 1: "Not Optimized" has been obtained using a fixed test set and no optimization phase. "Optimized" using a fixed test set and the optimization phase.

ence: the test set was selected randomly from the training set (after cleaning phase). In this way we are guaranteed that the system has seen all the necessary information in training phase, and we can assess its limitations in these very ideal conditions. We are aware this condition is extremely idealized and it will never happen in real life, but we wanted to have an upper bound on the performance achievable by this architecture if access to ideal data was not an issue. We also made sure that the performance on translating training sentences was not due to simple memorization of the entire sentence, verifying that the vast majority of the sentences were not present in the translation table (where the maximal phrase size was 7), not even in reduced form. Under these favourable conditions, the system obtained a NIST score of around 11, against a score of about 7.5 on unseen sentences. This suggests that the phrase-based Markov-chain representation is sufficiently rich to obtain a high score, if the necessary information is contained in the translation and language models.

For each model to be tested on known sentences, we have sampled ten subsets of 2,000 sentences each from the training set.

The "Optimized, Test on Training Set" learning curve, see figure 2, represents a possible upper bound on the best performance of this SMT system, since it has been computed in favourable conditions. It does suggest that this hypothesis class has the power of approximating the target behaviour more accurately than we could think based on performance on unseen sentences. If the right information has been seen, the system can reconstruct the sentences rather accurately. The NIST score computed using the reference sentences as target sentences is around 15, we identify the relative curve as "Human Translation". At this point, it seems likely that the process with which we learn the necessary tables representing the knowledge of the system is responsible for the performance limitations.

The gap between the "Optimized, Test on Training Set" and the "Optimized" curves is even more interesting if related to the slow growth rate in the previous learning curve: although the system can represent internally a good model of translation, it seems unlikely that this will ever be inferred by increasing the size of training datasets in realistic amounts.

The training step results in various forms of knowledge: translation table, language model and parameters from the optimization. The internal models learnt by the system are essentially lists of phrases, with probabilities associated to them. Which of these components is mostly responsible for performance limitations?

4.3 Experiment 3: effect on performance of increasing noise levels in parameters

Much research has focused on devising improved principles for the statistical estimation of the parameters in language and translation models. The introduction of discriminative graphical models has marked a departure from traditional maximum likelihood estimation principles, and various approaches have been proposed.

The question is: how much information is contained in the fine grain structure of the probabilities estimated by the model? Is the performance improving with more data because certain parameters are estimated better, or just because the lists are growing? In the second case, it is likely that more sophisticated statistical algorithms to improve the estimation of probabilities will have limited impact.

In order to simulate the effect of inaccurate estimation of the numeric parameters, we have added increasing amount of noise to them. This can either represent the effect of insufficient statistics in estimating them, or the use of imperfect parameter esti-

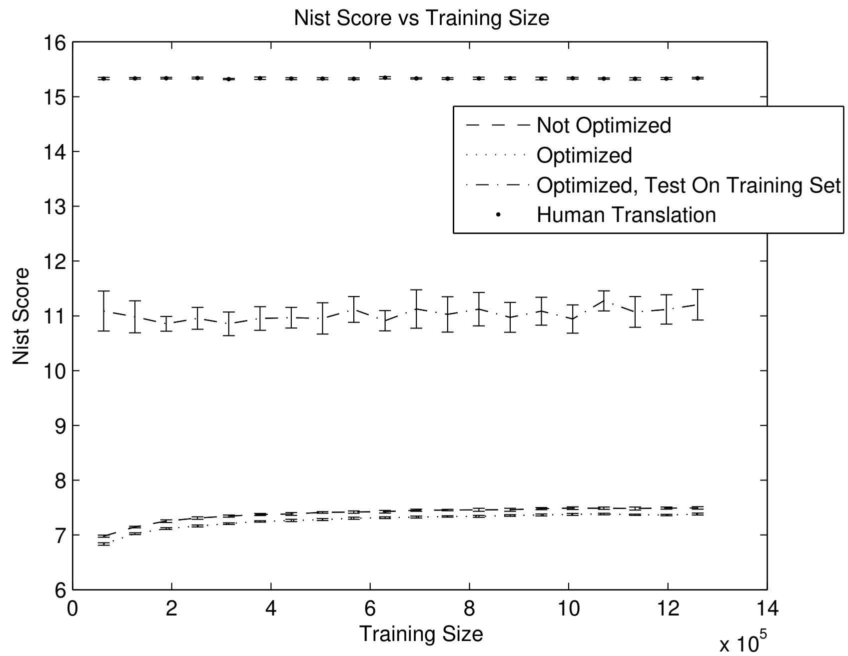

Figure 2: Four learning curves have been compared. "Not Optimized" has been obtained using a fixed test set and no optimization phase. "Optimized" using a fixed test set and the optimization phase. "Optimized Test On Training Set" a test set selected by the training set for each training set size and the optimization phase. "Human Translation" has been obtained by computing NIST using the reference English sentence of the test set as target sentences.

mation biases. We have corrupted the parameters in the language and translation models, by adding increasing levels of noise to them, and measured the effect of this on performance.

One model trained with 62,995 pairs of sentences has been chosen from the experiments in Section 4.1. A percentage of noise has been added to each probability in the language model, including conditional probability and back off, translation model, bidirectional translation probabilities and lexicalized weighting. Given a probability p and a percentage of noise, pn, a value has been randomly selected from the interval [-x,+x], where $x = p * pn$, and added to p. If this quantity is bigger than one it has been approximated to one. Different values of percentage have been used. For each value of pn, five experiment have been run. The optimization step has not been run.

We see from Figure 3 that the performance does not seem to depend crucially on the fine structure of the parameter vectors, and that even a large addition of noise (100%) produces a 10% decline in NIST score. This suggests that it is the list itself, rather

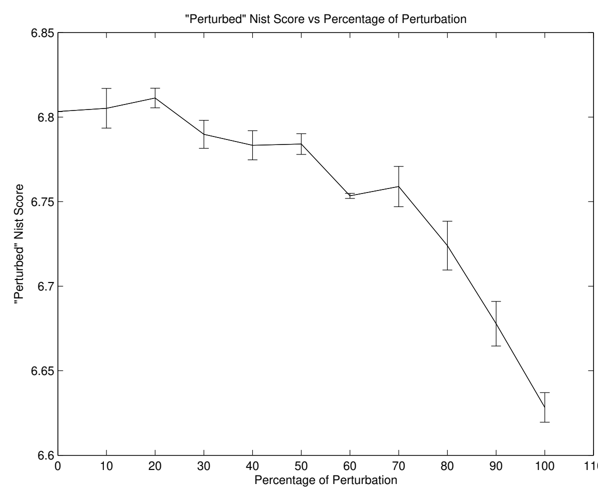

Figure 3: Each probability of the language and translation models has been perturbed adding a percentage of noise. This learning curve reports the not optimized NIST score versus the percentage of perturbation applied. These results have been obtained using a fixed training set size equal to 62,995 pairs of sentences.

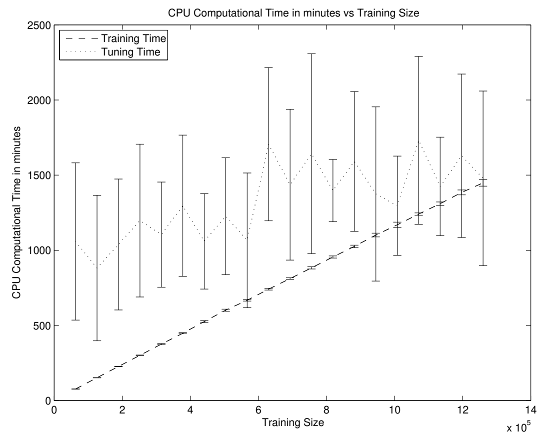

Figure 4: Training and tuning user time vs training set size. Time quantities are expressed in minutes.

than the probabilities in it, that controls the performance. Different estimation methods can produce different parameters, but this does not seem to matter very much. The creation of a more complete list of words, however, seems to be the key to improve the score. Combined with the previous findings, this would mean that neither more data nor better statistics will bridge the performance gap. The solution might have to be found elsewhere, and in our Discussion section we outline a few possible avenues.

5 Computational Cost

The computational cost of models creation and development-phase has been measured during the creation of the learning curves. Despite its efficiency in terms of data usage, the development phase has a high cost in computational terms, if compared with the cost of creating the complete language and translation models.

For each experiment, the user CPU time is computed as the sum of the user time of the main process and the user time of the children.

These quantities are collected for training, development, testing and evaluation phases. In figure 4, training and tuning user times are plotted as a function of the training set size. It is evident that increasing the training size causes an increase in training time in a roughly linear fashion.

It is hard to find a similar relationship for the tuning time of the development phase. In fact, the tuning time is strictly connected with the optimization algorithm and the sentences in the development set. We can also see in figure 4 that even a small development set size can require a large amount of tuning time. Each point of the tuning time curve has a big variance. The tuning phase involves translating the development set many times and hence its cost depends very weakly on the training set size, since a large training set leads to larger tables and these lead to slightly longer test times.

6 Discussion

The impressive capability of current machine translation systems is not only a testament to an incredibly productive and creative research community, but can also be seen as a paradigm for other Artificial Intelligence tasks. Data driven approaches to all main areas of AI currently deliver the state of the art performance, from summarization to speech recognition to machine vision to information retrieval. And statistical learning technology is central to all approaches to data driven AI.

Understanding how sophisticated behaviour can be learnt from data is hence not just a concern for machine learning, or to individual applied communities, such as Statistical Machine Translation, but rather a general concern for modern Artificial Intelligence. The analysis of learning curves, and the identification of the various limitations to performance is a crucial part of the machine learning method, and one where statistics and algorithmics interact closely.

In the case of Statistical Machine Translation, the analysis of Moses suggests that the current bottleneck is the lack of sufficient data, not the function class used for the representation of translation systems. The clear gap between performance on training and testing set, together with the rate of the learning curves, suggests that improvements may be possible but not by adding more data in i.i.d. way as done now. The perturbation analysis suggests that improved statistical principles are unlikely to make a big difference either.

Since it is unlikely that sufficient data will be available by simply sampling a distribution, one needs to address a few possible ways to transfer large amounts of knowledge into the system. All of them lead to open problems either in machine learning or in machine translation, most of them having been already identified by their respective communities as important questions. They are actively being worked on.

The gap between performances on training and on test sets is typically affected by model selection choices, ultimately controlling the trade off between overfitting and underfitting. In these experiments the system used phrases of length 7 or less. Changing this parameter might reflect on the gap and this is the focus of our current work.

A research programme naturally follows from our analysis. The first obvious approach is an effort to identify or produce datasets on demand (active learning, where the learning system can request translations of specific sentences, to satisfy its information needs). This is a classical machine learning question, that however comes with the need for further theoretical work, since it breaks the traditional i.i.d. assumptions on the origin of data. Furthermore, it would also require an effective way to do confidence estimation on translations, as traditional active learning approaches are effectively based on the identification (or generation) of instances where there is low confidence in the output (Blatz et al., 2004; Ueffing and Ney, 2004; Ueffing and Ney, 2005b; Ueffing and Ney, 2005a).

The second natural direction involves the introduction of significant domain knowledge in the form of linguistic rules, so to dramatically reduce the amount of data needed to essentially reconstruct them by using statistics. These rules could take the form of generation of artificial training data, based on existing training data, or a posteriori expansion of translation and language tables. Any way to enforce linguistic constraints will result in a reduced need for data, and ultimately in more complete models, given the same amount of data (Koehn and Hoang, 2007).

Obviously, it is always possible that the identification of radically different representations of language might introduce totally different constraints on both approximation and estimation error, and this might be worth considering.

What is not likely to work. It does not seem that the introduction of more data will change the situation significantly, as long as the data is sampled i.i.d. from the same distribution. It also does not seem that more flexible versions of Markov models would be likely to change the situation. Finally, it does not seem that new and different methods to estimate probabilities would make much of a difference. Our perturbation studies show that significant amounts of noise in the parameters result into very small variations in the performance. Note also that the current algorithm is not even working on refining the probability estimates, as the rate of growth of the tables suggests that new n-grams are constantly appearing, reducing the proportion of time spent refining probabilities of old n-grams.

It does seem that the control of the performance relies on the length of the translation and language tables. Ways are needed to make these tables grow much faster as a function of training set size; they can either involve active selection of documents to translate, or the incorporation of linguistic rules to expand the tables without using extra data.

It is important to note that many approaches suggested above are avenues currently being actively pursued, and this analysis might be useful to decide which one of them should be given priority.

7 Conclusions

We have started a series of extensive experimental evaluations of performance of Moses, using high performance computing, with the goal of understanding the system from a machine learning point of view, and use this information to identify weaknesses of the system that can lead to improvements. We have performed many more experiments that cannot be reported in this workshop paper, and will be published in a longer report (Turchi et al., In preparation). In general, our goal is to extrapolate the performance of the system under many conditions, to be able to decide which directions of research are most likely to deliver improvements in performance.

Acknowledgments

Marco Turchi is supported by the EU Project SMART. The authors thank Callum Wright, Bristol HPC Systems Administrator, and Moses mailing list.

References

Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F. J. Och, D. Purdy, N. A. Smith, and D. Yarowsky. 1999. Statistical machine translation: Final report. Technical report, Johns Hopkins University 1999 Summer Workshop on Language Engineering, Center for Speech and Language Processing.

S. Banerjee and A. Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics.

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. 2004. Confidence estimation for machine translation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 315, Morristown, NJ, USA. Association for Computational Linguistics.

P. F. Brown, S. Della Pietra, V.t J. Della Pietra, and R. L. Mercer. 1994. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

G. 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*, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

P. Koehn and H. Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *the Annual Meeting of the Association for Computational Linguistics, demonstration session.*

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X*, pages 79–86.

K.Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL '02*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

A. Lavie and A. Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *ACL '07: Proceedings of 45th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

F. J. Och and H. Ney. 2001. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL '02*, pages 295–302, Morristown, NJ, USA. Association for Computational Linguistics.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F. J. Och and H. Weber. 1998. Improving statistical natural language translation with categories and rules. In *COLING-ACL*, pages 985–989.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL '03*, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.

R.Zens, F. J.Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *KI '02: Proceedings of the 25th Annual German Conference on AI*, pages 18–32, London, UK. Springer-Verlag.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231. Association for Machine Translation in the Americas.

A. Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *Intl. Conf. on Spoken Language Processing*.

M. Turchi, T. De Bie, and N. Cristianini. In preparation. Learning analysis of a machine translation system.

N. Ueffing and H. Ney. 2004. Bayes decision rules and confidence measures for statistical machine translation. In *EsTAL-2004*, pages 70–81.

N. Ueffing and H. Ney. 2005a. Application of word-level confidence measures in interactive statistical machine translation. In *EAMT-2005*, pages 262–270.

N. Ueffing and H. Ney. 2005b. Word-level confidence estimation for machine translation using phrase-based translation models. In *Proceedings of HLT '05*, pages 763–770, Morristown, NJ, USA. Association for Computational Linguistics.