Multi-Engine Machine Translation with an Open-Source Decoder for Statistical Machine Translation

Yu Chen[1], **Andreas Eisele**[1,2], **Christian Federmann**[2],
Eva Hasler[3], **Michael Jellinghaus**[1], **Silke Theison**[1]
(authors listed in alphabetical order)
1: Saarland University, Saarbrücken, Germany
2: DFKI GmbH, Saarbrücken, Germany
3: University of Cologne, Germany

Abstract

We describe an architecture that allows to combine statistical machine translation (SMT) with rule-based machine translation (RBMT) in a multi-engine setup. We use a variant of standard SMT technology to align translations from one or more RBMT systems with the source text. We incorporate phrases extracted from these alignments into the phrase table of the SMT system and use the open-source decoder Moses to find good combinations of phrases from SMT training data with the phrases derived from RBMT. First experiments based on this hybrid architecture achieve promising results.

1 Introduction

Recent work on statistical machine translation has led to significant progress in coverage and quality of translation technology, but so far, most of this work focuses on translation into English, where relatively simple morphological structure and abundance of monolingual training data helped to compensate for the relative lack of linguistic sophistication of the underlying models. As SMT systems are trained on massive amounts of data, they are typically quite good at capturing implicit knowledge contained in co-occurrence statistics, which can serve as a shallow replacement for the world knowledge that would be required for the resolution of ambiguities and the insertion of information that happens to be missing in the source text but is required to generate well-formed text in the target language.

Already before, decades of work went into the implementation of MT systems (typically rule-based) for frequently used language pairs[1], and these systems quite often contain a wealth of linguistic knowledge about the languages involved, such as fairly complete mechanisms for morphological and syntactic analysis and generation, as well as a large number of bilingual lexical entries spanning many application domains.

It is an interesting challenge to combine the different types of knowledge into integrated systems that could then exploit both explicit linguistic knowledge contained in the rules of one or several conventional MT system(s) and implicit knowledge that can be extracted from large amounts of text.

The recently started EuroMatrix[2] project will explore this integration of rule-based and statistical knowledge sources, and one of the approaches to be investigated is the combination of existing rule-based MT systems into a multi-engine architecture. The work described in this paper is one of the first incarnations of such a multi-engine architecture within the project, and a careful analysis of the results will guide us in the choice of further steps within the project.

2 Architectures for multi-engine MT

Combinations of MT systems into multi-engine architectures have a long tradition, starting perhaps with (Frederking and Nirenburg, 1994). Multi-engine systems can be roughly divided into simple

[1]See (Hutchins et al., 2006) for a list of commercial MT systems

[2]See http://www.euromatrix.net

Proceedings of the Second Workshop on Statistical Machine Translation, pages 193–196,
Prague, June 2007.

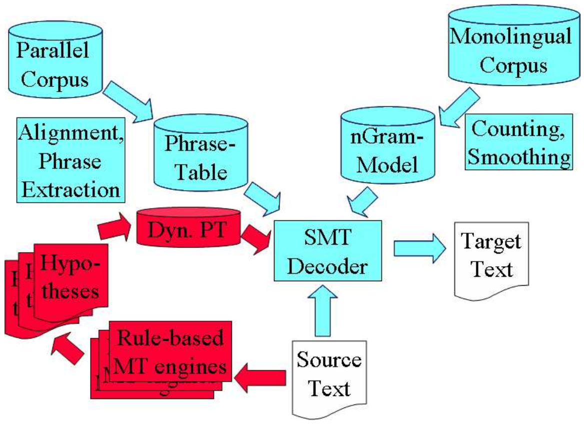

Figure 1: Architecture for multi-engine MT driven by a SMT decoder

architectures that try to select the best output from a number of systems, but leave the individual hypotheses as is (Tidhar and Küssner, 2000; Akiba et al., 2001; Callison-Burch and Flournoy, 2001; Akiba et al., 2002; Nomoto, 2004; Eisele, 2005) and more sophisticated setups that try to recombine the best parts from multiple hypotheses into a new utterance that can be better than the best of the given candidates, as described in (Rayner and Carter, 1997; Hogan and Frederking, 1998; Bangalore et al., 2001; Jayaraman and Lavie, 2005; Matusov et al., 2006; Rosti et al., 2007).

Recombining multiple MT results requires finding the correspondences between alternative renderings of a source-language expression proposed by different MT systems. This is generally not straightforward, as different word order and errors in the output can make it hard to identify the alignment. Still, we assume that a good way to combine the various MT outcomes will need to involve word alignment between the MT output and the given source text, and hence a specialized module for word alignment is a central component of our setup.

Additionally, a recombination system needs a way to pick the best combination of alternative building blocks; and when judging the quality of a particular configuration, both the plausibility of the building blocks as such and their relation to the context need to be taken into account. The required optimization process is very similar to the search in a SMT decoder that looks for naturally sounding combinations of highly probable partial translations. Instead of implementing a special-purpose search procedure from scratch, we transform the information contained in the MT output into a form that is suitable as input for an existing SMT decoder. This has the additional advantage that resources used in standard phrase-based SMT can be flexibly combined with the material extracted from the rule-based MT results; the optimal combination can essentially be reduced to the task of finding good relative weights for the various phrase table entries.

A sketch of the overall architecture is given in Fig. 1, where the blue (light) parts represent the modules and data sets used in purely statistical MT, and the red (dark) parts are the additional modules and data sets derived from the rule-based engines. It should be noted that this is by far not the only way to combine systems. In particular, as this proposed setup gives the last word to the SMT decoder, we risk that linguistically well-formed constructs from one of the rule-based engines will be deteriorated in the final decoding step. Alternative architectures are under exploration and will be described elsewhere.

3 MT systems and other knowledge sources

For the experiments, we used a set of six rule-based MT engines that are partly available via web interfaces and partly installed locally. The web based systems are provided by Google (based on Systran for the relevant language pairs), SDL, and ProMT which all deliver significantly different output. Locally installed systems are OpenLogos, Lucy (a recent offspring of METAL), and translate pro by lingenio (only for German ↔ English). In addition to these engines, we also used the scripts included in the Moses toolkit (Koehn et al., 2006)[3] to generate phrase tables from the training data. We enhanced the phrase tables with information on whether a given pair of phrases can also be derived via a third, intermediate language. We assume that this can be useful to distinguish different degrees of reliability, but due to lack of time for fine-tuning we could not yet show that it indeed helps in increasing the overall quality of the output.

[3]see http://www.statmt.org/moses/

4 Implementation Details

4.1 Alignment of MT output

The input text and the output text of the MT systems was aligned by means of GIZA++ (Och and Ney, 2003), a tool with which statistical models for alignment of parallel texts can be trained. Since training new models on merely short texts does not yield very accurate results, we applied a method where text can be aligned based on existing models that have been trained on the Europarl Corpus (Koehn, 2005) beforehand. This was achieved by using a modified version of GIZA++ that is able to load given models.

The modified version of GIZA++ is embedded into a client-server setup. The user can send two corresponding files to the server, and specify two models for both translation directions from which alignments should be generated. After generating alignments in both directions (by running GIZA++ twice), the system also delivers a combination of these alignments which then serves as input to the following steps described below.

4.2 Phrase tables from MT output

We then concatenated the phrase tables from the SMT baseline system and the phrase tables obtained from the rule-based MT systems and augmented them by additional columns, one for each system used. With this additional information it is clear which of the MT systems a phrase pair stems from, enabling us to assign relative weights to the contributions of the different systems. The optimal weights for the different columns can then be assigned with the help of minimum error rate training (Och, 2003).

5 Results

We compared the hybrid system to a purely statistical baseline system as well as two rule-based systems. The only differences between the baseline system and our hybrid system are the phrase table – the hybrid system includes more lexical entries than the baseline – and the weights obtained from minimum error rate training.

For a statistical system, lexical coverage becomes an obstacle – especially when the bilingual lexical entries are trained on documents from different domains. However, due to the distinct mechanisms used to generate these entries, rule-based systems and statistical systems usually differ in coverage. Our system managed to utilize lexical entries from various sources by integrating the phrase tables derived from rule-based systems into the phrase table trained on a large parallel corpus. Table 1 shows

Systems	Token #
Ref.	2091 (4.21%)
R-I	3886 (7.02%)
R-II	3508 (6.30%)
SMT	3976 (7.91%)
Hybrid	2425 (5.59%)

Table 1: Untranslated tokens (excl. numbers and punctuations) in output for news commentary task (de-en) from different systems

a rough estimation of the number of untranslated words in the respective output of different systems. The estimation was done by counting "words" (i.e. tokens excluding numbers and punctuations) that appear in both the source document and the outputs. Note that, as we are investigating translations from German to English, where the languages share a lot of vocabulary, e.g. named entities such as "USA", there are around 4.21% of words that should stay the same throughout the translation process. In the hybrid system, 5.59% of the words remain unchanged, which is is the lowest percentage among all systems. Our baseline system (SMT in Table 1), not comprising additional phrase tables, was the one to produce the highest number of such untranslated words.

	Baseline	Hybrid
test	18.07	21.39
nc-test	21.17	22.86

Table 2: Performance comparison (BLEU scores) between baseline and hybrid systems, on in-domain (test) and out-of-domain (nc-test) test data

Higher lexical coverage leads to better performance as can be seen in Table 2, which compares BLEU scores of the baseline and hybrid systems, both measured on in-domain and out-of-domain test data. Due to time constraints these numbers reflect results from using a single RBMT system (Lucy); using more systems would potentially further improve results.

6 Outlook

Due to lack of time for fine-tuning the parameters and technical difficulties in the last days before delivery, the results submitted for the shared task do not yet show the full potential of our architecture.

The architecture described here places a strong emphasis on the statistical models and can be seen as a variant of SMT where lexical information from rule-based engines is used to increase lexical coverage. We are currently also exploring setups where statistical alignments are fed into a rule-based system, which has the advantage that well-formed syntactic structures generated via linguistic rules cannot be broken apart by the SMT components. But as rule-based systems typically lack mechanisms for ruling out implausible results, they cannot easily cope with errors that creep into the lexicon due to misalignments and similar problems.

7 Acknowledgements

This research has been supported by the European Commission in the FP6-IST project EuroMatrix. We also want to thank Teresa Herrmann for helping us with the Lucy system.

References

Yasuhiro Akiba, Kenji Imamura, and Eiichiro Sumita. 2001. Using multiple edit distances to automatically rank machine translation output. In *Proceedings of MT Summit VIII*, Santiago de Compostela, Spain.

Yasuhiro Akiba, Taro Watanabe, and Eiichiro Sumita. 2002. Using language and translation models to select the best among outputs from multiple mt systems. In *COLING*.

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *ASRU*, Italy.

Chris Callison-Burch and Raymond S. Flournoy. 2001. A program for automatically selecting the best output from multiple machine translation engines. In *Proc. of MT Summit VIII*, Santiago de Compostela, Spain.

Andreas Eisele. 2005. First steps towards multi-engine machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, June.

Robert E. Frederking and Sergei Nirenburg. 1994. Three heads are better than one. In *ANLP*, pages 95–100.

Christopher Hogan and Robert E. Frederking. 1998. An evaluation of the multi-engine MT architecture. In *Proceedings of AMTA*, pages 113–123.

John Hutchins, Walter Hartmann, and Etsuo Ito. 2006. IAMT compendium of translation software. Twelfth Edition, January.

Shyamsundar Jayaraman and Alon Lavie. 2005. Multi-engine machine translation guided by explicit word matching. In *Proc. of EAMT*, Budapest, Hungary.

P. Koehn, M. Federico, W. Shen, N. Bertoldi, O. Bojar, C. Callison-Burch, B. Cowan, C. Dyer, H. Hoang, R. Zens, A. Constantin, C. C. Moran, and E. Herbst. 2006. Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding. Final Report of the 2006 JHU Summer Workshop.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the MT Summit*.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *In Proc. EACL*, pages 33–40.

Tadashi Nomoto. 2004. Multi-engine machine translation with voted language model. In *Proc. of ACL*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of ACL*, Sapporo, Japan, July.

Manny Rayner and David M. Carter. 1997. Hybrid language processing in the spoken language translator. In *Proc. ICASSP '97*, pages 107–110, Munich, Germany.

Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie J. Dorr. 2007. Combining translations from multiple machine translation systems. In *Proceedings of the Conference on Human Language Technology and North American chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL'2007)*, pages 228–235, Rochester, NY, April 22-27.

Dan Tidhar and Uwe Küssner. 2000. Learning to select a good translation. In *COLING*, pages 843–849.