Towards Deeper MT - A Hybrid System for German

Eleftherios Avramidis, Maja Popović*, Aljoscha Burchardt and Hans Uszkoreit
German Research Center for Artificial Intelligence (DFKI)
Language Technology Lab
firstname.lastname@dfki.de
∗ Humboldt University of Berlin
maja.popovic@hu-berlin.de

Abstract

The idea to improve MT quality by using deep linguistic and knowledge-driven information has frequently been expressed. If the goal is to use deep information for building an MT system, there are two extreme options: (1) to start from a purely knowledge-driven approach (RBMT) and try to arrive at the same recall found in current SMT systems; (2) to start from an SMT system and try to arrive at higher precision by modifying it so that more knowledge drives the translation process.

The system architecture we will describe in this paper starts in the middle of these extreme options. It is a hybrid architecture that we take as a starting point for future experiments and extensions to increase MT quality by more knowledge-driven processing.

1 Introduction

Statistical Machine Translation (SMT) based on comparably shallow features can be considered the most successful paradigm in Machine Translation (MT). The processing pipelines and machine learning architectures have become quite sophisticated and complex and allow for many types of optimisations. SMT systems have a (theoretical) high recall in the sense that they provide output for most input and that the pieces that would constitute a good translation are usually present somewhere in a huge search space during the translation process (e.g. in phrase tables or language models). However, it is very difficult to arrive at high precision, i.e., to automatically choose the right pieces and put together a fluent and accurate translation of a given input. The idea to further improve MT quality by adding deeper (i.e., more linguistic and knowledge-driven) information has thus frequently been expressed.

At the same time, rule-based MT systems that primarily apply such knowledge and that are able to control precision much better are used only in certain nieches today. The reason is that they lack recall: for example, parsing failure or gaps in the lexicon typically lead to a dead-end where the only option is to manually code the missing information, which is too resource intensive especially if one wants to take care of those less frequent items and phenomena in the “long tail”.

If one has the goal to use deep information for building an MT system with the best possible results, there are two extreme options: (1) to start from a purely knowledge-driven approach and try to arrive at the same (theoretical) recall found in current SMT systems; (2) to start from an SMT system and try to arrive at high precision by modifying it so that knowledge drives the search process. Today, it is an open research question what will lead to the best results in the end.

The system architecture we will describe below starts in the middle of both extreme options. It is a hybrid architecture that we take as starting point for future experiments and extensions to increase MT quality by more knowledge-driven processing. This systems has been developed within the QTLeap project[1] where it serves as a “deeper” baseline system as compared to a pure SMT baseline. The goal of the project is to explore different combinations of shallow and deep processing for improving MT quality. The system presented in this paper is the first of a series of system prototypes developed in the project. We therefore refer to it as System 1 in this paper.

[1] http://qtleap.eu/

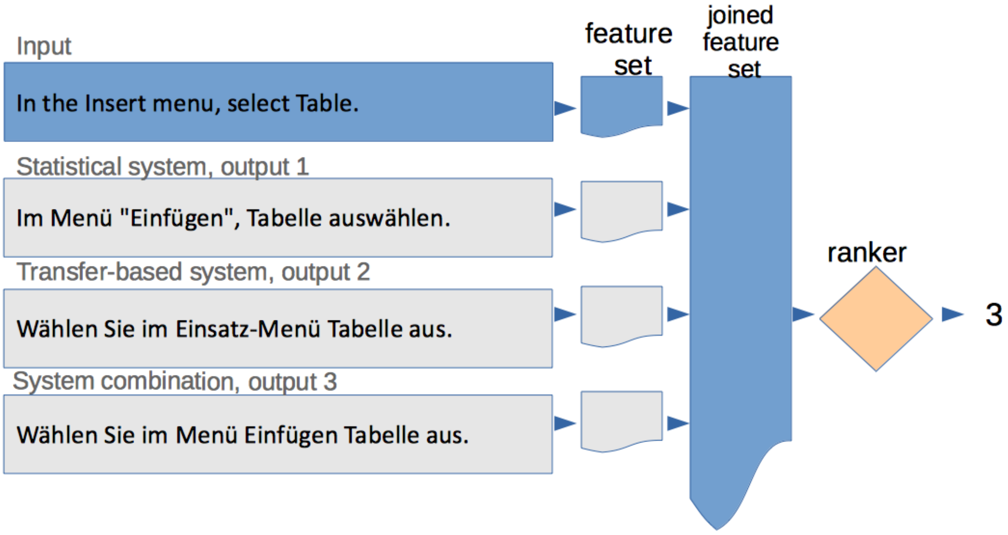
Input
In the Insert menu, select Table.
Statistical system, output 1
Im Menü "Einfügen", Tabelle auswählen.
Transfer-based system, output 2
Wählen Sie im Einsatz-Menü Tabelle aus.
System combination, output 3
Wählen Sie im Menü Einfügen Tabelle aus.
feature set
joined feature set
ranker
3

Figure 1: Architecture of System 1.

2 A Hybrid System Combination for German↔English

The fact that German is relatively well-resourced in comparison to other language pairs has allowed MT researchers to build strong statistical systems with very good performance on a lexical or a local level (Bojar et al., 2014). The German-English MT system we present here (System 1) aims to effectively incorporate deep linguistic processing into existing successful machine translation methods for this language pair.

Since our main goal is to achieve a high-quality system that allows for experimentation, competes with state-of-the-art systems, and can be useful in the projects real use-case scenario (translating user queries and expert answers in a chat-based PC helpdesk scenario), we use a system implementation that takes advantage of deep transfer and also includes a statistical mechanism that enhances performance by keeping the best parts from each employed method. Figure 1 shows the overall hybrid architecture that includes:

- A statistical Moses system,
- the commercial transfer-based system Lucy,
- their serial system combination, and
- an informed selection mechanism (“ranker”).

The components of this hybrid system will be detailed in the sections below.

2.1 Statistical MT system: Moses

Our statistical machine translation component was based on a vanilla phrase-based system built with Moses (Koehn et al., 2007) trained on the following corpora: Europarl ver. 7, News Commentary ver. 9 (Bojar et al., 2014), Commoncrawl (Smith et al., 2013), and MultiUN (Eisele and Chen, 2010) as well as on the following domain corpora: the Document Foundation (Libreoffice Help – 47K sentence pairs, Libreoffice User Interface – 35K parallel entries), the Document Foundation Terminology (690 translated terms), the Document Foundation Website (226 sentence pairs), Chromium browser (6,3K parallel entries), Ubuntu Documentation (6,3K sentence pairs), Ubuntu Saucy (183K parallel entries), and Drupal web-content management (5K parallel entries). Language models of order 5 have been built and interpolated with SRILM (Stolcke, 2002) and KenLM (Heafield, 2011). For German to English, we also experimented with the method of pre-ordering the source side based on the target-side grammar

(Popovic and Ney, 2006). As a tuning set we used the *news-test 2013*. In our architecture, this system on its own also serves as baseline.

2.2 Transfer-based MT system: Lucy

The transfer-based core of System 1 is based on the Lucy system (Alonso and Thurmair, 2003) that includes the results of long linguistic efforts over the last decades and that has successfully been used in previous projects including Euromatrix+ and QTLaunchPad.

The transfer-based approach has shown good results that compete with pure statistical systems, although its focus is on translating according to linguistic structures sets. Translation occurs in three phases, namely analysis, transfer, and generation. All three phases consist of hand-coded linguistic rules which have shown to perform well for capturing the structural and semantic differences between German and other languages. During the analysis phase, a parsing algorithm constructs a tree of the source language using a monolingual lexicon and the included grammar rules. The analysis algorithm reverts to a shallower analysis at the phrasal level in cases when the engine is not able to process the full tree. The analysis tree is subsequently used for the transfer phase, where deep representations of the source are transferred into deep representations of the target language using a bilingual lexicon based on canonical forms and categories. The generation phase creates the target sentence on the lexical level, using inflection and agreement rules between the dependent target language structures. A RestAPI allows the different processing steps and/or intermediate results to be influenced.

Deep features for empirical enhancement Although deep techniques indicate good coverage of a number of linguistic phenomena, each of the three phases may frequently encounter serious robustness issues and/or the inability to fully process a given sentence. Erroneous analysis from early phases may aggregate along the pipeline and cause further sub-optimal choices in later phases, thus severely deteriorating the quality of the produced translation. Preliminary analysis (Federmann and Hunsicker, 2011) has shown that such is the case for source sentences that are ungrammatical in the first place or that have a very shallow syntax with many specialized lexical entries. To tackle these issues, we combine the transfer-based component with our supportive SMT engine in the following two ways:

(a) train a statistical machine translation to automatically post-edit the output of the transfer-based system ("serial combination")

(b) use the post-edited or the SMT output in cases where the transfer-based system exhibits lower performance. This is done through an empirical *selection mechanism* that performs real-time analysis of the produced translations and automatically selects the output that is predicted to be of a better quality (Avramidis, 2011). Figure 1 shows the overall architecture of System 1 for en→de.

2.3 Serial System Combination: Lucy+Moses

For automatic post-editing of the transfer-based system, a serial Transfer+SMT system combination is used, as described in (Simard et al., 2007) The first stage is translation of the source-language part of the training corpus by the transfer-based system. The second stage is training an SMT system with the transfer-based translation output as a source language and the target-language part as a target language. Later, the test set is first translated by the transfer-based system, and the obtained translation is translated by the SMT system. Figure 2 illustrates the architecture for translation direction en→de. Note that the notion of "German*" in the figure is meant to distinguish the input and output of the SMT system. "German*" is the normal output of the transfer-based system.

In this linear system combination, improvement of up to 6 absolute BLEU points has been achieved for both translation directions in several pilot evaluations. Nevertheless, the method on its own could not outperform the SMT system trained on a large parallel corpus. The example in Figure 1 nicely illustrates how the statistical post-editing operates.

While the original SMT output used the right terminology ("Menü Einfügen" – "insert menu"), the instruction (*Im Menü "Einfügen", Tabelle auswählen*) is stylistically not very polite. In contrast, the output of the transfer-based system (*Wählen Sie im Einsatz-Menü Tabelle aus*) is formulated politely, yet mistranslates the menu type. The serial system combination produces a perfect translation. In this particular case, the machine translation (*Wählen Sie im Einfügen Menü Tabelle aus*) is even better than the human reference (*Wählen Sie im Einfügen Menü die Tabelle aus*) as the latter introduces a determiner for "table" that is not justified by the source.

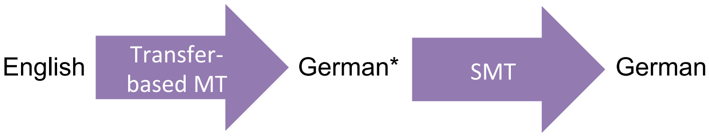

Figure 2: Serial System Combination en→de.

2.4 Parallel System Combination: Selection Mechanism

The selection mechanism is based on encouraging results of previous projects including Euromatrix Plus (Federmann and Hunsicker, 2011), T4ME (Federmann, 2012), QTLaunchPad (Avramidis, 2013; Shah et al., 2013). It has been extended to include several deep features that can only be generated on a sentence level and that would otherwise blatantly increase the complexity of the transfer or decoding algorithm. In System 1, automatic syntactic and dependency analysis is employed on a sentence level, in order to choose the sentence that fulfills the basic quality aspects of the translation: (a) assert the fluency of the generated sentence, by analyzing the quality of its syntax (b) ensure its adequacy, by comparing the structures of the source with the structures of the generated sentence.

All deep features produced are used to build a ranker based on machine learning against training preference labels. Preference labels are part of the training data and indicate which system output for a given source sentence is of optimal quality. Preference labels are generated either by automatic reference-based metrics or derived from human preferences. The ranker is a result of experimenting with various combinations of feature set and machine learning algorithms and choosing the one that performs best on the project corpus. The selection mechanism is based on the "Qualitative" toolkit that was presented in the MT Marathon, as an open-source contribution (Avramidis et al., 2014).

Feature sets We started from feature sets that performed well in previous experiments and we experimented with several extensions and modifications. In particular:

- Basic syntax-based feature set: unknown words, count of tokens, count of alternative parse trees, count of verb phrases, parse log likelihood. Parse was done with Berkeley Parser and features were extracted from both source and target. This feature set has performed well as a metric in WMT11 metrics task.

- Basic feature set + 17 QuEst[2] baseline features: this feature set combines the basic syntax-based feature set described above with the baseline feature set of the QuEst toolkit. This feature set combination obtained the best result in the WMT13 quality estimation task.

- Basic syntax-based feature set with Bit Parser: here we replace the Berkeley parser features on the target side with Bit Parser.

- Advanced syntax-based feature set: this augments the basic set by adding IBM model 1 probabilities, full depth of parse trees, depth of the 'S' node, position of the VP and other verb nodes from the beginning and end of the parent node, count of unpaired brackets and compound suggestions (for German, as indicated by LanguageTool.org).

Machine Learning We tested all suggested feature sets with many machine learning methods, including Support Vector Machines (with both RBF and linear kernel), Logistic Regression, Extra/Decision Trees, k-neighbors, Gaussian Naive Bayes, Linear and Quadratic Discriminant Analysis, Random Forest

[2] http://www.quest.dcs.shef.ac.uk/

and Adaboost ensemble over Decision Trees. The binary classifiers were wrapped into rankers using the "soft pairwise recomposition" to avoid ties between the systems.

The classifiers were trained on MT outputs of all systems that participated in the translation shared tasks of WMT (years 2008-2014). We also experimented on several sources of sentence level preference labels, in particular human ranks, METEOR and F-score. We chose the label type that maximizes (if possible) all automatic scores, including document-level BLEU.

Best combination The optimal systems used:

1. the *basic syntax-based feature set* for English-German, trained with Support Vector Machines against METEOR scores.

2. the *advanced syntax-based feature set* for German-English, trained with Linear Discriminant Analysis against METEOR scores as well.

Table 1 shows the results of the selection mechanism on a test set used in the QTLeap project that consists of 1000 German questions and English answers to be translated in the respective other language.[3]

The table quantifies the contribution of the three systems: Transfer-based, SMT, and the linear Transfer+SMT combination. It is notable that the mechanism in many cases favors transfer-based output, which is an indication that the deep features are active; one would have expected a bias towards SMT for a shallower selection mechanism. However, this first hypothesis needs to be confirmed by further studies.

	Transfer	SMT	Transfer+SMT
de→en questions	45.2%	33.3%	23.8%
en→de answers	42.5 %	16.3%	50.5%

Table 1: Percentages chosen automatically by the selection mechanism from each of the systems. Percentages which sum more than 100% indicate ties. When ties occur, there is a preset order of preference SMT, Transfer, Transfer+SMT.

3 Evaluation

3.1 Automatic Evaluation

Translation results were evaluated using three automatic metrics: BLEU,[4] word-level F-score (wordF) and character-level F-score (charF) using `rgbF.py` (Popovic, 2012). F-scores are calculated on 1-grams, 2-grams, 3-grams and 4-grams and then averaged using the arithmetic mean. The final score is obtained in the usual way and is the harmonic mean of precision and recall. Although BLEU is certainly the most used automatic metric, F-score has been shown to correlate better with human judgments, especially if n-grams are averaged using arithmetic instead of geometric mean. We also calculated character level F-score because the target language is morphologically rich.

As baseline, we used the Moses SMT system described above on its own. Following the evaluation scenario in the project, we evaluate on the translation of questions for the direction German into English and on the answers only for the direction from English to German. Table 2 shows the scores for the baseline (Moses) and contrasts them with the results for System 1.

The results show that System 1 performs comparably to the baseline for translation of questions into English while the translation of answers into German still poses more problems. In addition to the scores discussed above, the translation errors were analyzed using Hjerson (Popović, 2011), an automatic tool for error analysis that provides a categorization into five classes:

- word form (agreement errors, tense, capitalization, part of speech)

[3] The corpus is available for Basque, Bulgarian, Czech, Dutch, English, German, Portuguese and Spanish and can be downloaded from the META-SHARE portal (`http://metashare.metanet4u.eu/`) under the name "QTLeap Corpus".

[4] We used the official BLEU script `mteval-v13a.pl --international-tokenization`.

		questions de→en	answers en→de
Moses	BLEU	43.0	41.7
	wordF	44.6	42.2
	charF	64.9	64.7
System 1	BLEU	43.3	33.0
	wordF	43.8	30.2
	charF	63.4	57.4

Table 2: BLEU scores, word-level and character-level F-scores for Moses baseline and System 1 translation outputs.

- word order
- omission
- addition
- mistranslation (general mistranslations, terminology errors, style, punctuation and any changes to wording)

For each error class, the tool provides raw error counts together with error rates (raw counts normalized over the total number of words in the translation output). Block error counts and block error rates are calculated as well, where the block refers to a group of successive words belonging to the same error class.

The tool is language independent. It requires the translation output and a reference, both in full form and lemmatized. During the evaluation experiments, it has been observed that there are a number of capitalization errors (or inconsistencies between the reference and the translation), such as "OpenOffice" vs. "openOffice", "VLC" vs "vlc", etc. Therefore we subsequently calculated capitalization error rates as difference between word form error rate of true-cased texts and word form error rates of lower-cased texts that are displayed in the table below. The pure morphological errors are those obtained with lower-cased texts. In order to arrive at a fair treatment of the prevalent items in the input such as "File > Save As" or URLs, we have reported block error rates instead of word-level error rates.

The results are presented in Table 3. The error classification results are presented below, in the form of block error rates (lower is better). The error rates read as follows, taking Moses de→enas an example: 12.2% of the word groups in the translation output are mistranslated in comparison to the human reference (i.e. these words are different than the reference words). So, if the system has translated 100 words, ca. 12 (consecutive blocks of) words consist of other words than found in the reference.

		questions de→en	answers en→de
Moses	form	1.2	4.4
	order	6.5	5.7
	omission	4.4	4.6
	addition	2.8	3.7
	mistranslation	12.2	11.9
System 1	form	1.1	4.0
	order	5.6	5.6
	omission	3.4	3.0
	addition	3.6	7.4
	mistranslation	12.8	13.5

Table 3: Class error rates for Moses and System 1 translation outputs.

When going from Moses to System 1, this automatic analysis indicates that the number of morphological errors, reordering errors and omissions goes down slightly while the number of mistranslations (lexical errors) goes up. The most striking difference is the increase in additions when translating into German which almost doubles. The reason for this might be that deeper systems produce structurally different translations that do not match the reference translations. This needs to be analysed in more detail.

3.2 User Evaluation

Finally, we also evaluated the performance of System 1 compared to the Moses baseline in a task-based user evaluation performed by volunteers that will be published in this volume (Del Gaudio et al.). Explained briefly, users were presented a technical question (in German) in a web interface, a German reference answer and answers translated from English by Moses and System 1 in random order. They had to indicate which MT answer is better or if both are the same given these options (where A and B are the two systems, respectively):

i A is a better answer than B

ii B is a better answer than A

iii A and B are equally good answers

iv A and B are equally bad answers

100 question-answer pairs were judged by three volunteers. If we lump ties (i.e., iii and iv) together, the central (averaged) results of the user evaluation are:

- System 1 has been judged better than Moses in 17.3% of cases (i)
- System 1 has been judged better or same as Moses in 75.5 % of cases (i+iii+iv)

Given that, for translation into German, the BLEU score of System 1 is more than 8 points worse than that of Moses, further detailed investigation is needed to interpret these results.

4 Summary and outlook

In this paper, we've described a first experimental systems that combines deep and shallow MT components in different hybrid combinations. The goal is to explore various ways of using "deeper" information for translation between English and German. Evaluation has shown that the hybrid system performs comparably to an SMT baseline for some tasks, yet shows worse performance on others. A small user evaluation has shown promising results. In the future, various experiments and improvements are possible and foreseen, starting from improving the transfer-based system (handling of lexical items such as terminology, MWEs, untranslatables, and robustness of parsing), the serial combination (e.g., improved disambiguation), and moving up to more detailed analysis and testing and improvement of the selection mechanism (e.g., integrating more linguistic information from external parsing).

Acknowledgments

This paper has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 610516 (QTLeap: Quality Translation by Deep Language Engineering Approaches). We are grateful to the anonymous reviewers for their valuable feedback.

References

Juan A. Alonso and Gregor Thurmair. 2003. The comprendium translator system. In *Proceedings of the Ninth Machine Translation Summit*, New Orleans, LA, September.

Eleftherios Avramidis, Lukas Poustka, and Sven Schmeier. 2014. Qualitative: Open source python tool for quality estimation over multiple machine translation outputs. *The Prague Bulletin of Mathematical Linguistics*, 102(1):5–16.

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 Optimising the Division of Labour in Hybrid Machine Translation (ML4HMT-11)*, Barcelona, Spain. Center for Language and Speech Technologies and Applications (TALP), Technical University of Catalonia.

Eleftherios Avramidis. 2013. Sentence-level ranking with quality estimation. *Machine Translation (MT)*, 28(Special issue on Quality Estimation):1–20.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Andreas Eisele and Yu Chen. 2010. MultiUN: A Multilingual Corpus from United Nation Documents. In Daniel Tapias, Mike Rosner, Stelios Piperidis, Jan Odjik, Joseph Mariani, Bente Maegaard, Khalid Choukri, and Nicoletta Calzolari (Conference Chair), editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation. International Conference on Language Resources and Evaluation (LREC-2010), May 19-21, La Valletta, Malta*, pages 2868–2872. European Language Resources Association (ELRA).

Christian Federmann and Sabine Hunsicker. 2011. Stochastic Parse Tree Selection for an Existing RBMT System. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 351–357, Edinburgh, Scotland, July. Association for Computational Linguistics.

Christian Federmann. 2012. Can Machine Learning Algorithms Improve Phrase Selection in Hybrid Machine Translation? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation*, pages 113–118, Avignon, France, April. European Chapter of the Association for Computational Linguistics (EACL).

Kenneth Heafield. 2011. KenLM : Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, number 2009, pages 187–197, Edinburgh, Scotland, July. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Chris Zens, Richard a nd Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Maja Popovic and Hermann Ney. 2006. Pos-based word reorderings for statistical machine translation. In *International Conference on Language Resources and Evaluation*, pages 1278–1283.

Maja Popović. 2011. Hjerson: An Open Source Tool for Automatic Error Classification of Machine Translation Output. *The Prague Bulletin of Mathematical Linguistics*, (96):59–68, October.

Maja Popovic. 2012. rgbf: An open source tool for n-gram based automatic evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:99–108, 10.

Kashif Shah, Eleftherios Avramidis, Ergun Biçici, and Lucia Specia. 2013. QuEst: Design, Implementation and Extensions of a Framework for Machine Translation Quality Estimation. *The Prague Bulletin of Mathematical Linguistics (PBML)*, 100:19–30.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Proceedings of The North American Chapter of the Association for Computational Linguistics Conference (NAACL-07)*, pages 508–515, Rochester, NY, April.

Jason R Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt Cheap Web-Scale Parallel Text from the Common Crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1383, Sofia, Bulgaria, August. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm — an Extensible Language Modeling Toolkit. In *System*, volume 2, pages 901–904. ISCA, September.