The QT21 Combined Machine Translation System for English to Latvian

Jan-Thorsten Peter1, Hermann Ney1, Ondřej Bojar2, Ngoc-Quan Pham3, Jan Niehues3, Alex Waibel1, Franck Burlot4, François Yvon4, Mārcis Pinnis5, Valters Šics5, Joost Bastings6, Miguel Rios6, Wilker Aziz6, Philip Williams7, Frédéric Blain8, Lucia Specia8

1RWTH Aachen University, Aachen, Germany
2Charles University, Prague, Czech Republic
3Karlsruhe Institute of Technology, Karlsruhe, Germany
4LIMSI, CNRS, Université Paris Saclay, 91 403 Orsay, France
5Tilde, Riga, Latvia
6University of Amsterdam, Amsterdam, Netherlands
7University of Edinburgh, Edinburgh, UK
8University of Sheffield, Sheffield, UK

Abstract

This paper describes the joint submission of the QT21 projects for the English→Latvian translation task of the EMNLP 2017 Second Conference on Machine Translation (WMT 2017). The submission is a system combination which combines seven different statistical machine translation systems provided by the different groups.

The systems are combined using either RWTH’s system combination approach, or USFD’s consensus-based system-selection approach. The final submission shows an improvement of 0.5 BLEU compared to the best single system on newstest2017.

1 Introduction

Quality Translation 21 (QT21) is a European machine translation research project with the aim of substantially improving statistical and machine learning based translation models for challenging languages and low-resource scenarios.

Members of the QT21 project have jointly built a combined statistical machine translation system, in order to achieve high-quality machine translation from English into Latvian.

Core components of the QT21 combined system for the WMT 2017 shared task for machine translation of news 1 are seven individual English→Latvian translation engines which have been set up by different project partners.

The outputs of all these individual engines are combined using the system combination approach as implemented in Jane, RWTH’s open source statistical machine translation toolkit (Freitag et al., 2014a). The Jane system combination is a mature implementation which previously has been successfully employed in other collaborative projects and for different language pairs (Peter et al., 2016; Freitag et al., 2013, 2014b,c).

As an alternative way of combining our systems, all outputs have been merged as the form of an n-best list and a consensus-based system-selection applied to obtain as best translation hypothesis the candidate that is most similar to the most likely translations amongst those systems.

2 Preprocessing

The training data was pre-processed using a custom language-specific tokenizer and the Moses truecaser (truecase.perl). For tokenisation, we used the Tilde’s regular expression-based tokenizer for Latvian and English that takes into account language-specific characteristics (e.g., abbreviations, contractions, date, time, and numerical expressions, etc.) and non-translatable entities (e.g., phone numbers, e-mail addresses, XML tags, URLs, file paths, various identifiers and codes, etc.). Only the first word in each sentence was truecased.

The data (backtranslation included) is further cleaned using a simple language identifier from Shuyo (2010). We simply removed sentence pairs whose targets cannot be identified by the
tool. The number of sentences being removed is approximately 50000.

3 Translation Systems

Each group contributed one or more systems. In this section the systems are presented in alphabetic order.

3.1 CUNI

The CUNI component of the system was built using Neural Monkey\(^2\) (Helcl and Libovický, 2017), a flexible sequence-to-sequence toolkit implementing primarily the Bahdanau et al. (2015) model but useful also in multi-modal translation and multi-task training.

We used essentially the baseline setup of the system as released for the WMT17 NMT Training Task\(^3\) (Bojar et al., 2017) for an 8GB GPU card. This involves BPE (Sennrich et al., 2016) with 30k merges, maximum sentence length for both source and target limited to 50 (BPE) tokens, no dropout and embeddings (both source and target) of 600, vocabulary shared between encoder and decoder, attention and conditional GRU (Firat and Cho, 2016). We experimented with the RNN size of the encoder and decoder and increased them to 800 instead of 600, at the expense of reducing batch size to 10. The batch size of 30 with this enlarged model would still fit into our GPU card but this run was prematurely interrupted due to a hardware failure and we noticed that it converges slower in terms of sentence pairs (not in terms of wallclock time), so we opted for a more efficient use of the training data by taking the smaller batch.

We trained on 5245514 sentence pairs mixing the genuine parallel data and synthetic data, as described in Section 2. Neural Monkey does not shuffle the corpus, so we shuffled it beforehand and kept the order identical for all training epochs.

The training ran for 15 days on NVIDIA GeForce GTX 1080 and processed 4.7 epochs but the best model (according to BLEU scores on the development set, “devset-b”) was actually reached after 11M sentence pairs (early epoch 3), after 7 days.

Neitherensembling nor beam-search was used for the run, because they were not yet available in Neural Monkey. Instead, the translations were generated using greedy search.

3.2 KIT

The neural machine translation models from KIT are built with the OpenNMT framework (Klein et al., 2017), which is a multi-layer LSTM encoder decoder network. We trained the models with 2.1 million parallel sentence pairs concatenated with 2.8 million pairs from backtranslation provided by University of Edinburgh. The networks have 1024 hidden units for each of 2 LSTM layers for both encoder and decoder. Furthermore, we experiment a number of features with the baseline:

First, we found out that using a context gate to mask activities between the decoder hidden state and the source context vector before producing the distribution at each time step (Tu et al., 2016a) is simple yet beneficial for performance. Second, we strengthen the attentional network with a coverage vector accumulating the previous attentional information, similar to the work of Mi et al. (2016) and Tu et al. (2016b).

Using the two techniques helps improve the BLEU score on the newsdev2017 set by 1.1 (tokenized) BLEU. By using ensembling 3 networks with different configs and rescoring using a model trained with reversed target sentences, we managed to reach 26.96 BLEU score for the development set, which yields 2.8 point of improvement compared to the baseline model. Details about the effect of each technique is described in Pham et al. (2017).

3.3 LIMSI

LIMSI’s input to this system combination consists of two NMT systems, both trained with the NMTPY framework (Caglayan et al., 2017) on bitext, then on synthetic parallel data. All of them were rescored with a Nematus system (Sennrich et al., 2017b). More details about these systems can be found in (Burlot et al., 2017b,a).

The first system, named baseline, is a BPE-to-BPE system. Bilingual sub-word units (Sennrich et al., 2016) were trained on the bitext parallel data with 90k merge operations. All the parameters of the neural network were initialized with Xavier. The system was optimized with Adam, dropout was enabled on source embeddings, encoder states, as well as output layer. The whole training process took approximately 1.5 months.
Table 1: Results of the individual systems for the English→Latvian task. B[LEU] [%] and T[ER] [%] scores are case-sensitive.

<table>
<thead>
<tr>
<th>Individual Systems</th>
<th>newsdev2017/1</th>
<th>newsdev2017/2</th>
<th>newstest2017</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>TER</td>
<td>CTER</td>
</tr>
<tr>
<td>Tilde smt</td>
<td>21.3</td>
<td>59.6</td>
<td>57.8</td>
</tr>
<tr>
<td>CUNI neuralmonkey</td>
<td>18.9</td>
<td>63.2</td>
<td>62.6</td>
</tr>
<tr>
<td>UvA</td>
<td>16.0</td>
<td>70.0</td>
<td>71.9</td>
</tr>
<tr>
<td>KIT</td>
<td>26.8</td>
<td>53.4</td>
<td>49.6</td>
</tr>
<tr>
<td>UEDIN NMT</td>
<td>25.7</td>
<td>55.1</td>
<td>51.6</td>
</tr>
<tr>
<td>UEDIN rescored by KIT</td>
<td>25.9</td>
<td>54.8</td>
<td>52.0</td>
</tr>
<tr>
<td>LIMSI factored</td>
<td>24.3</td>
<td>57.3</td>
<td>53.2</td>
</tr>
<tr>
<td>System Combination</td>
<td>27.4</td>
<td>53.1</td>
<td>50.9</td>
</tr>
</tbody>
</table>

The results shown in Table 1 correspond to an ensemble of our three best models, which produced n-best hypothesis. Finally, these hypothesis were rescored using a Nematus system trained on the same data as the baseline and with similar hyperparameters.

The second system is an experiment with factored NMT, which is part of the NMTPY framework (García-Martínez et al., 2016). The hyper-parameters mentioned above for the baseline also hold for this system. The specific setup we have used consisted in an architecture that enables training towards a dual objective: at each time-step in the output sentence, a normalized word and a PoS-tag are produced. To obtain the first factor vocabulary, all target words have been normalized (Burlot and Yvon, 2017a), i.e. all grammatical information that is redundant wrt. English has been removed from the words. In a nutshell, the normalization system performs a clustering of the morphologically rich language by grouping together words that tend to share the same translation(s) in English. As a result, words are represented by a lemma and a cluster identifier containing the morphological features that have been merged. In our setup, the cluster identifier was systematically split from the lemma. BPE segmentation was thus learnt and applied to lemmas.

Given a lexical unit and a PoS-tag, word forms are retrieved with a dictionary lookup. In the context of morphologically rich languages, deterministic mappings from a lemma and a PoS to a form are very rare. Instead, the dictionary often proposes several word forms corresponding to the same lexical unit and morphological analysis. To address this issue, we let a word-based system select the right word form from the dictionary. To this end, k-best hypothesis from the dictionary were generated, as well as the n-best hypothesis from the factored NMT system, leading to nk-best rescoring.

Our factored NMT system is an ensemble of two best models and rescoring is performed with our single best Nematus model.

3.4 Tilde
The Tilde system is a Moses phrase-based SMT system that was trained on the Tilde MT platform (Vasiljevs et al., 2012). The system was trained using all available parallel data - 1.74 million unique sentence pairs after filtering, and 3 million unique sentence pairs that were acquired by re-translating a random selection of in-domain monolingual sentences with a neural machine translation system (Pinnis et al., 2017). The system has a 5-gram language model that was trained using KenLM (Heafield, 2011) on all available monolingual data (27.83 million unique sentences).

3.5 UEDIN
The University of Edinburgh’s system is an attentional encoder-decoder (Bahdanau et al., 2015), trained using the Nematus toolkit (Sennrich et al., 2017c).

As training data, we used all parallel and synthetic data, which was tokenized, truecased, and filtered as described in Section 2. After filtering, the data was segmented into subword units using byte-pair-encoding (BPE), for which we used 90,000 operations, jointly learned over both sides of the parallel corpora.

We used word embeddings of size 512 and hidden layers of size 1024, with the size of the source and target network vocabularies fixed to the size of the respective BPE vocabularies. In order to reduce the size of the models, the target-side embedding weights were tied with the transpose of...
the output weight matrix (Press and Wolf, 2017). We used a deep transition architecture inspired by the one proposed by Zilly et al. (2016) for language modelling. In experiments conducted during feature development, we found that this gave consistent improvements across multiple language pairs. We also applied layer normalisation (Ba et al., 2016) to all recurrent and feed-forward layers, except for layers that are followed by a softmax. In preliminary experiments, we found that using layer normalisation led to faster convergence and resulted in slightly better performance.

We trained the models with adam (Kingma and Ba, 2015), using a learning rate of 0.0001 and mini-batch size of 80. Training was automatically stopped when the validation cross-entropy failed to reach a new minimum for 10 consecutive savepoints (saving every 10000 updates).

For our final system, we trained eight independent models: four left-to-right and four right-to-left. We used results on newstdev2017 to select one checkpoint from each model. An ensemble of the four left-to-right models was used to generate a 50-best list, which was rescored using the right-to-left models.

For a more detailed description of the system, see Sennrich et al. (2017a).

3.6 UvA: syntactically aware NMT with GCNs

We focus on exploiting structural information on the source side, i.e. in the encoder. We hypothesize that an encoder that incorporates syntax will lead to more informative representations of words, and that these representations, when used as context vectors by the decoder, will lead to an improvement in translation quality. Our model (Bastings et al., 2017) is an attentive encoder-decoder (Bahdanau et al., 2015) where in the encoder side we exploit the power of GCNs (Kipf and Welling, 2016) to induce syntactically-aware representations (Marcheggiani and Titov, 2017). GCNs operate by convolving nodes in a neighbourhood defined by a graph. In our case, a node corresponds to a position in the source sentence which is initially represented by a BiRNN hidden state. We then define a syntactic neighbourhood by following edges in an automatically produced dependency parse. Instead of relying on linear order only (as the BiRNN does), the GCN allows the encoder to ‘teleport’ over parts of the source sentence connecting words that are potentially far apart. The model might not only benefit from this teleporting capability however; also the nature of the relations between words (i.e. dependency relation types and directionality) may be useful, and the GCN exploits this information.

4 System Combination

We conducted experiments with two methods for system combination that only require the translated hypotheses. This allows us choose the contributing systems without any restrictions.

4.1 Confusion Network

System combination produces consensus translations from multiple hypotheses which are obtained from different translation approaches, i.e., the systems described in the previous section. A system combination implementation developed at RWTH Aachen University (Freitag et al., 2014a) is used to combine the outputs of the different engines. The consensus translations outperform the individual hypotheses in terms of translation quality.

The first step in system combination is the generation of confusion networks (CN) from input translation hypotheses. We need pairwise alignments between the input hypotheses, which are obtained from METEOR (Banerjee and Lavie, 2005). The hypotheses are then reordered to match a selected skeleton hypothesis in terms of word ordering. We generate I different CNs, each having one of the input systems as the skeleton hypothesis, and the final lattice is the union of all generated CNs. In Figure 1 an example of a confusion network with I = 4 input translations is depicted. Decoding of a confusion network finds the best path in the network. Each arc is assigned a score of a linear model combination of M different models, which includes word penalty, 3-gram language model trained on the input hypotheses, a binary primary system feature that marks the primary hy-
Table 2: USFD rescoring and combination experiments English→Latvian task. BLEU [%] and TER [%] scores are case-sensitive.

<table>
<thead>
<tr>
<th>Systems</th>
<th>newsdev2017/1</th>
<th>newsdev2017/2</th>
<th>newstest2017</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>TER</td>
<td>cTER</td>
</tr>
<tr>
<td>KIT 20best Blue</td>
<td>25.8</td>
<td>54.5</td>
<td>51.3</td>
</tr>
<tr>
<td>KIT 20best ChrF</td>
<td>25.4</td>
<td>55.0</td>
<td>50.5</td>
</tr>
<tr>
<td>KIT 20best Beer</td>
<td>26.0</td>
<td>54.1</td>
<td>50.2</td>
</tr>
<tr>
<td>LIMSI Factored 12best Bleu</td>
<td>19.7</td>
<td>60.4</td>
<td>55.7</td>
</tr>
<tr>
<td>LIMSI Factored 12best ChrF</td>
<td>19.7</td>
<td>60.3</td>
<td>55.4</td>
</tr>
<tr>
<td>LIMSI factored 12best Beer</td>
<td>19.8</td>
<td>60.3</td>
<td>55.5</td>
</tr>
<tr>
<td>LIMSI factored 100best Bleu</td>
<td>21.5</td>
<td>59.1</td>
<td>55.5</td>
</tr>
<tr>
<td>LIMSI factored 100best ChrF</td>
<td>21.9</td>
<td>58.6</td>
<td>54.3</td>
</tr>
<tr>
<td>LIMSI factored 100best Beer</td>
<td>21.7</td>
<td>58.7</td>
<td>54.3</td>
</tr>
<tr>
<td>Consensus-based System-selection Bleu</td>
<td>19.8</td>
<td>72.5</td>
<td>60.1</td>
</tr>
<tr>
<td>Consensus-based System-selection ChrF</td>
<td>26.5</td>
<td>54.1</td>
<td>49.3</td>
</tr>
<tr>
<td>Consensus-based System-selection Beer</td>
<td>27.1</td>
<td>53.0</td>
<td>49.6</td>
</tr>
<tr>
<td>System Combination</td>
<td>27.4</td>
<td>53.1</td>
<td>50.9</td>
</tr>
<tr>
<td>System Combination + Cons-based Beer</td>
<td>27.4</td>
<td>52.7</td>
<td>50.0</td>
</tr>
</tbody>
</table>

4.2 Consensus-based System Selection

As a secondary solution for system combination, we used USFD’s consensus-based n-best list selection approach (Blain et al., 2017) for system combination by combining each system’s output in the form of a n-best list. Inspired by DeNero et al. (2009)’s work on consensus-based Minimum Bayes Risk (MBR) decoding which compares different types of similarity metrics (BLEU, WER, etc.) under a SMT setup, USFD designed a re-ranking approach to empirically evaluate the effect of consensus on the varying n-best list in NMT.

Given a n-best list, each translation hypothesis is scored against the other MT candidates of the search space towards an automatic metric. In our experiment we considered three automatic metrics amongst the most widely used and which have been shown to be well correlated with human judgments (Bojar et al., 2016): BLEU, BEER (Stanojevic and Simaan, 2014) or CHRF (Popovic, 2015). The entire list of MT candidates is then entirely re-ranked according to the averaged score of each candidate. Different from most re-ranking approaches which make use of additional information usually treated as new model components and combined with the existing ones, we here focus only on the MT candidates. The difference between the consensus-based n-best list selection and an oracle translation is the absence of reference translation: each translation hypothesis is scored against all the other hypotheses used as references while in an oracle translation each translation hypothesis is scored against a single reference. This results in obtaining as best translation hypothesis the candidate that is most similar to the most likely translations.

5 Experimental Evaluation

Since only one development set was provided we split the given development set into two parts: newsdev2017/1 and newsdev2017/2. The first part was used as development set while the second part was our internal test set. The single systems and the system combination are optimized for the newsdev2017/1 set.

The single system scores in Table 2 show that the KIT system is the strongest single system closely followed by the UEDIN NMT system. The rescoring of the UEDIN NMT nbest lists by KIT showed only a small improvement on newstest2017. The system combination of all these systems showed an improvement of 1.1 BLEU on newsdev2017/2 and 0.5 BLEU on official test set, newstest2017.

Table 3 shows a comparison between all systems by scoring the translation output against each other in TER and BLEU. We see that the outputs of the two best performing systems KIT and UEDIN are very close.

6 Morphology Evaluation

In order to get some insight regarding the quality of the morphological correctness of the outputs produced by the systems involved in the combina-
Table 3: Comparison of system outputs against each other, generated by computing BLEU and TER on the system translations for newstest2017. One system in a pair is used as the reference, the other as candidate translation; we report the average over both directions. The USFD system is similar to the “Consensus-based System-selection Beer” in Table 2. The upper-right half lists BLEU [%] scores, the lower-left half TER [%] scores.

<table>
<thead>
<tr>
<th>System</th>
<th>CUNI</th>
<th>KIT</th>
<th>LIMSI</th>
<th>Tilde</th>
<th>UEDIN</th>
<th>UEDIN r.</th>
<th>UvA</th>
<th>USFD</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUNI</td>
<td></td>
<td>38.1</td>
<td>32.4</td>
<td>23.9</td>
<td>37.8</td>
<td>38.2</td>
<td>22.3</td>
<td>40.2</td>
<td>33.3</td>
</tr>
<tr>
<td>KIT</td>
<td>43.8</td>
<td></td>
<td>49.3</td>
<td>29.9</td>
<td>60.1</td>
<td>62.6</td>
<td>28.3</td>
<td>77.0</td>
<td>49.3</td>
</tr>
<tr>
<td>LIMSI</td>
<td>49.8</td>
<td>33.2</td>
<td></td>
<td>26.1</td>
<td>48.6</td>
<td>49.4</td>
<td>27.5</td>
<td>56.7</td>
<td>41.4</td>
</tr>
<tr>
<td>Tilde</td>
<td>57.0</td>
<td>47.4</td>
<td>52.7</td>
<td></td>
<td>30.1</td>
<td>30.2</td>
<td>19.4</td>
<td>31.6</td>
<td>27.3</td>
</tr>
<tr>
<td>UEDIN</td>
<td>45.1</td>
<td>25.6</td>
<td>35.1</td>
<td>48.9</td>
<td></td>
<td>91.1</td>
<td>28.5</td>
<td>76.2</td>
<td>53.2</td>
</tr>
<tr>
<td>UEDIN rescored by KIT</td>
<td>44.5</td>
<td>23.8</td>
<td>34.3</td>
<td>48.4</td>
<td>5.4</td>
<td></td>
<td>28.7</td>
<td>78.4</td>
<td>54.1</td>
</tr>
<tr>
<td>UvA</td>
<td>62.9</td>
<td>56.6</td>
<td>57.5</td>
<td>65.8</td>
<td>57.1</td>
<td>56.7</td>
<td></td>
<td>30.3</td>
<td>26.4</td>
</tr>
<tr>
<td>USFD</td>
<td>42.0</td>
<td>13.9</td>
<td>28.1</td>
<td>45.9</td>
<td>15.8</td>
<td>14.2</td>
<td>54.7</td>
<td></td>
<td>55.8</td>
</tr>
<tr>
<td>Average</td>
<td>49.3</td>
<td>34.9</td>
<td>41.5</td>
<td>52.3</td>
<td>33.3</td>
<td>32.5</td>
<td>58.8</td>
<td>30.7</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Sentence pair evaluation (A-set).

<table>
<thead>
<tr>
<th>System</th>
<th>verbs</th>
<th>pronouns</th>
<th>nouns</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tilde smt</td>
<td>68.8%</td>
<td>70.4%</td>
<td>56.0%</td>
<td>71.8%</td>
</tr>
<tr>
<td>UvA</td>
<td>75.2%</td>
<td>84.2%</td>
<td>46.4%</td>
<td>80.8%</td>
</tr>
<tr>
<td>UEDIN NMT</td>
<td>74.6%</td>
<td>83.6%</td>
<td>57.0%</td>
<td>88.6%</td>
</tr>
<tr>
<td>LIMSI NMT</td>
<td>68.8%</td>
<td>84.6%</td>
<td>64.2%</td>
<td>86.8%</td>
</tr>
<tr>
<td>LIMSI factored</td>
<td>69.6%</td>
<td>82.8%</td>
<td>62.0%</td>
<td>89.0%</td>
</tr>
<tr>
<td>KIT</td>
<td>74.2%</td>
<td>89.0%</td>
<td>56.6%</td>
<td>89.8%</td>
</tr>
</tbody>
</table>

Table 5: Sentence pair evaluation (B-set).

<table>
<thead>
<tr>
<th>System</th>
<th>nouns</th>
<th>adjectives</th>
<th>verbs</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tilde smt</td>
<td>.436</td>
<td>.755</td>
<td>.735</td>
<td>.768</td>
</tr>
<tr>
<td>UvA</td>
<td>.295</td>
<td>.629</td>
<td>.613</td>
<td>.643</td>
</tr>
<tr>
<td>UEDIN</td>
<td>.234</td>
<td>.598</td>
<td>.596</td>
<td>.628</td>
</tr>
<tr>
<td>LIMSI NMT</td>
<td>.255</td>
<td>.616</td>
<td>.610</td>
<td>.644</td>
</tr>
<tr>
<td>LIMSI factored</td>
<td>.233</td>
<td>.587</td>
<td>.582</td>
<td>.612</td>
</tr>
<tr>
<td>KIT</td>
<td>.244</td>
<td>.599</td>
<td>.594</td>
<td>.633</td>
</tr>
</tbody>
</table>

Table 6: Sentence group evaluation with Entropy (C-set).
tion, we ran the evaluation method introduced in (Burlot and Yvon, 2017b). The evaluation of the morphological competence of a machine translation system is performed on an automatically produced test suite. For each source test sentence from a monolingual corpus (the base), one (or several) variant(s) are generated, containing exactly one difference with the base, focusing on a specific target lexeme of the base. These variants differ on a feature that is expressed morphologically in the target, such as the person, number or tense of a verb; or the number or case of a noun or an adjective. This artificial test set is then translated with a machine translation system. The machine translation system is deemed correct if the translations of the base and variant differ in the same way as their respective source. Another setup focuses on a word in the base sentence and produces variants containing antonyms and synonyms of this word. The expected translation is then synonyms and antonyms bearing the same morphological features as the initial word.

There are three types of contrasts implying different sorts of evaluation:

- **A**: We check whether the morphological feature inserted in the source sentence has been translated (e.g. plural number of a noun). Accuracy for all morphological features is averaged over all sentences.

- **B**: We focus on various agreement phenomena by checking whether a given morphological feature is present in both words that need to agree (e.g. case of two nouns). Accuracy is computed here as well.

- **C**: We test the consistency of morphological choices over lexical variation (e.g. synonyms and antonyms all having the same tense) and measure the success based on the average normalized entropy of morphological features in the set of target sentences.

The results for the A-set are shown in Table 4 and reflect the adequacy of an output towards the source, or the quantity of morphological information that has been well conveyed from the source. Certain morphological features indicate rather low contrasts between statistical and neural systems (verb tense and pronoun gender), which shows the relevance of SMT systems in the combination.

Sets B and C are more focused on target monolingual phenomena, such as agreement, and assess the level of fluency of a system output. Here, the observed contrasts between statistical and neural systems are far more obvious: all B-set SMT scores are below 50%, whereas NMT scores are always above. Here again, the superior performance of KIT is noticed, at least for sets A and B. As for the C-set, LIMSI factored, KIT and UEDIN show a comparable high confidence in their morphology predictions across lexical variety.

### 6.1 Consensus-based re-ranking

We report in Table 2 the results of the consensus-based approach for either system re-ranking or system combination.

First, we applied our approach on both KIT and LIMSI-factored outputs. While we never outperform original systems’ performances, we observe that increasing the \( n \)-best size does help with a significant difference between LIMSI’s system 12- or 100-best. One would note that in both cases, consensus-based \( n \)-best list re-ranking with BEER seems to be performing the best amongst all metrics.

Then, we applied our approach at system-level by combining the outputs of all systems described in Section 3. Once again, we observe better performance with BEER compared to the other two metrics, reaching similar results as the system combination based on confusion network. The only noticeable exception being the CT TER score on newstest2017 which is significantly lower compared to the other system combination, most likely the benefit of using character-based metrics.

Finally, we combined both consensus-based selection confusion-based combination and although we observe similar performance to each system individually but a worse CT TER.

### 7 Conclusion

Our combined effort shows again that the combination of different SMT systems results in a better overall system. The final result improved by 0.5 BLEU points. Consensus-based re-ranking showed a performance close to the confusion network approach.

### Acknowledgments

This project has received funding from the European Union’s Horizon 2020 research and
innovation programme under grant agreements № 645452 (QT21).

References


Nakatani Shuyo. 2010. Language detection library for java.


