UdS-DFKI Participation at WMT 2019: 
Low-Resource (en–gu) and Coreference-Aware (en–de) Systems

Cristina España-Bonet
Saarland University
DFKI GmbH
druiter@lsv.uni-saarland.de
{cristinae,Josef.Van_Genabith}@dfki.de

Dana Ruiter
Saarland University
DFKI GmbH

Josef van Genabith
Saarland University
DFKI GmbH

Abstract
This paper describes the UdS-DFKI submission to the WMT2019 news translation task for Gujarati–English (low-resourced pair) and German–English (document-level evaluation). Our systems rely on the on-line extraction of parallel sentences from comparable corpora for the first scenario and on the inclusion of coreference-related information in the training data in the second one.

1 Introduction

This document describes the systems and experiments conducted to participate in the news translation tasks of WMT 2019 for Gujarati–English (gu–en, low-resourced language pair) and German–English (de–en, document-level evaluation). We use different approaches to tackle each setting.

Machine translation (neural, statistical or rule-based), usually operates on a sentence-by-sentence basis. However, when translating a coherent document, surrounding sentences may contain information that needs to be reflected in a local sentence. In our experiments for the document-level task in en2de, we explore how the information beyond sentence level can be made available to a neural machine translation (NMT) system by modifying —tagging— the data in order to include this knowledge. In a similar way, multilingual NMT systems have already been successfully built by only tagging the source data with the knowledge of the target language (Johnson et al., 2017; Ha et al., 2016). With this approach, we incorporate the knowledge that carries coreferences through a text in every sentence. We expect to improve the translation of ambiguous items such as pronouns in English, so we just tackle a specific number of problems and not translation quality in general.

The approach for the low-resource setting is completely different. In this case, we use a neural architecture that allows us to extract parallel data from comparable corpora and filter noise from the available parallel data. The additional data obtained in this way is then used to train SMT models, which we compare to a baseline trained on the available parallel data only to observe the effects of the extraction and filtering.

Below, we describe our coreference-aware system for en2de (Section 2) and our low-resourced approach for en–gu (Section 3). Finally we summarise our findings in Section 4.

2 Coreference-Aware English-to-German System

2.1 Data Preparation

Our system makes use of the annotation of coreference mentions through documents in the source side of the corpus. Documents are annotated with coreference chains using a neural-network-based mention-ranking model as implemented by the Stanford CoreNLP tool (Manning et al., 2014). The tool detects pronominal, nominal and proper names as mentions in a chain. For every mention, CoreNLP extracts its gender (male, female, neutral, unknown), number (singular, plural, unknown), and animacy (animate, inanimate, unknown). This information is not added directly but used to enrich the MT training data by applying a set of heuristics implemented in DocTrans:

- We enrich pronominal mentions with the head of the chain

\footnote{This system achieves a precision of 80% and recall of 70% on the CoNLL 2012 English Test Data (Clark and Manning, 2016).}

\footnote{https://github.com/cristinae/DocTrans/}
Pronoun "I" is not enriched with any coreference information. We clean the head by removing articles and Saxon genitives and we only consider heads with less than 4 tokens in order to avoid enriching a word with a full sentence.

- We enrich nominal mentions including proper names with the gender of the head.
- The head itself is enriched with she/he/it/they depending on its gender and animacy.

The example below shows how we tag the cleaned version of the head of the chain (fish skin) before a pronominal mention (it):

baseline:
I never cook with it.

coref:
I never cook with $<b_{crf}>$ fish skin $<e_{crf}>$ it.

In order to be able to do this processing, we need documents and that limits the amount of corpora we can use. Even though all the corpora made available for the shared task have document boundaries, ParaCrawl, for instance, has a mean of 1.06 sentences per document which makes it useless within our approach.

2.2 Corpus

Monolingual corpora. We use a subset of the NewsCrawl corpus in English and German (years 2014, 2017 and a part of 2018, named as ss-NewsCrawl in Table 1) to calculate word embeddings as explained in Section 2.3. We first use langdetect\textsuperscript{3} to extract only those sentences that are in the desired language and compile the final corpora to have a similar number of subword units (Sennrich et al., 2016a) in both languages and years ($\sim 4.10^6$). The corpus is further cleaned, tokenised, truecased (with Moses scripts\textsuperscript{4}) and BPEd (with subword-nmt\textsuperscript{5}). The vocabulary of the BPE model depends on the system and is detailed in Section 2.3.

Parallel corpora. Due to the restrictions explained in Section 2.1, we use the parallel corpora made available for the shared task in different proportions. Our base system uses CommonCrawl, Europarl, News Commentary and Rapid Corpus. Our large system also uses the ParaCrawl corpus but in a diluted way. The purpose of the dilution is to try to minimise the fact that due to the nature of our system we cannot use single sentences (intrasentence dependencies are already learned by an NMT system) or back-translations (quality is not good enough to extract coreference chains in a source sentence that is an automatic translation).

<table>
<thead>
<tr>
<th></th>
<th># lines</th>
<th>Small</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolingual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssNewsCrawl $en$</td>
<td>176,220,479</td>
<td>x1</td>
<td>x1</td>
</tr>
<tr>
<td>ssNewsCrawl $de$</td>
<td>220,443,585</td>
<td>x1</td>
<td>x1</td>
</tr>
<tr>
<td>Parallel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CommonCrawl</td>
<td>2,394,878</td>
<td>x1</td>
<td>x4</td>
</tr>
<tr>
<td>Europarl</td>
<td>1,775,445</td>
<td>x1</td>
<td>x4</td>
</tr>
<tr>
<td>NewsCommentary</td>
<td>328,059</td>
<td>x4</td>
<td>x16</td>
</tr>
<tr>
<td>Rapid</td>
<td>1,105,651</td>
<td>x1</td>
<td>x4</td>
</tr>
<tr>
<td>ParaCrawlFiltered</td>
<td>12,424,790</td>
<td>x0</td>
<td>x1</td>
</tr>
</tbody>
</table>

Table 1: Number of lines of the monolingual and parallel corpora used in the $en$-$de$ translation systems for the base and large configurations. The second and third columns show the amount of oversampling (or dilution) used in both cases.

Our NMT systems are trained using the transformer architectures implemented in the Marian toolkit (Junczys-Dowmunt et al., 2018). We use two architectures base and big as defined in
Vaswani et al. (2017):

Transformer base. 6-layer encoder–decoder with 8-head self-attention, a 2048-dim hidden feed-forward, and 512-dim word vectors. Growing learning rate from 0 to 0.0003 till update 16,000 (warmup). Decaying learning rate afterwards. Adam optimisation with $\beta_1=0.9$, $\beta_2=0.98$ and $\epsilon=1e-09$. Tied target embeddings.

Transformer big. As Transformer base but with word embeddings with 1024-dim, 4096-dim hidden feed-forward layers, learning rate of 0.0002 with the same warmup and decay, $\beta_2=0.998$.

Using these architectures as basis, we train several models on 4 TITAN X GPUs using an adaptive batch size that differ on:

- Corpus size. Small vs. Large as defined in Table 1

- Vocabulary. Joint en–de BPE with 40K subword units (join) vs. separated vocabularies with 50K subword units each (all the other models).

- Initial word embeddings. Source and target initialisation with monolingual embeddings estimated with word2vec (Mikolov et al., 2013) (Emb) vs. source and target initialisation with bilingual embeddings mapped using vecmap (Artetxe et al., 2017) (EmbMap) vs. no initialisation (all the other models).

- Annotation. No annotation (Baseline) vs. tags with coreference information (all the other models).

- Ensembling. Combinations of the previous models at decoding time.

The terms in parenthesis refer to the models in Table 2. Model names are structured as architectureVocabulary–Annotation–Embeddings–Corpus.

2.4 Results

Table 2 shows the BLEU scores of the different models and ensembles on newstest-2017 (validation) and news-test2018 (test). The first block presents the results of a baseline system without any document-level information; the second block shows the models explored to determine the best configuration; and the third block summarises the ensembling combinations explored in order to choose our primary submission.

The first thing to notice is that in terms of BLEU systems with and without coreference annotations are not significantly different (M01 vs. M08; M02 vs. M09/M11). Since we are modifying only specific aspects of the translation —few words in a document—, we do not obtain large improvements according to automatic evaluation measures, but we expect differences in translation quality according to human evaluators.

The vocabulary turned out to be critical. A system with a joint vocabulary of 40K subword units (M03) is 5-6 BLEU points below its counterpart with 50k units and independent vocabularies (M04). The embeddings are not that decisive. An initialisation of the system using bilingual embeddings slightly improves the results (M07 vs. M05; M10 vs. M09; M12 vs. M11). Using monolingual embeddings implies a very slow training. M06 in

<table>
<thead>
<tr>
<th>Model</th>
<th>news17</th>
<th>news18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M01:trBig-Baseline-Small</td>
<td>25.82</td>
<td>37.62</td>
</tr>
<tr>
<td>M02:trBig-Baseline-Large</td>
<td>27.07</td>
<td>40.38</td>
</tr>
<tr>
<td>Coreference-Aware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M03:trBase-Deep-Small</td>
<td>20.00</td>
<td>29.08</td>
</tr>
<tr>
<td>M04:trBase-Deep-Large</td>
<td>24.74</td>
<td>36.56</td>
</tr>
<tr>
<td>M05:trBase-Deep-Large</td>
<td>26.35</td>
<td>38.74</td>
</tr>
<tr>
<td>M06:trBase-Emb-Large</td>
<td>16.15</td>
<td>22.20</td>
</tr>
<tr>
<td>M07:trBase-EmbMap-Large</td>
<td>26.72</td>
<td>39.12</td>
</tr>
<tr>
<td>M08:trBig-Small</td>
<td>25.85</td>
<td>37.55</td>
</tr>
<tr>
<td>M09:trBig-Large</td>
<td>26.38</td>
<td>38.53</td>
</tr>
<tr>
<td>M10:trBig-EmbMap-Large</td>
<td>26.33</td>
<td>39.12</td>
</tr>
<tr>
<td>M11:trBig-2-Large</td>
<td>27.42</td>
<td>40.07</td>
</tr>
<tr>
<td>M12:trBig-2-EmbMap-Large</td>
<td>27.28</td>
<td>40.28</td>
</tr>
<tr>
<td>Ensembling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M05-M07-M10</td>
<td>27.18</td>
<td>40.92</td>
</tr>
<tr>
<td>M07-M09</td>
<td>27.29</td>
<td>40.10</td>
</tr>
<tr>
<td>M05-M07-M09</td>
<td>27.24</td>
<td>40.56</td>
</tr>
<tr>
<td>M05-M07-M10-M10</td>
<td>27.31</td>
<td>40.98</td>
</tr>
<tr>
<td>M05-M07-M11</td>
<td>27.58</td>
<td>41.58</td>
</tr>
<tr>
<td>M07-M10-M11-M12</td>
<td><strong>27.62</strong></td>
<td><strong>42.82</strong></td>
</tr>
</tbody>
</table>

Table 2: BLEU scores of the models trained for the en2de translation task. The boldfaced ensembled model was submitted as the primary submission; the best performing model with boldfaced BLEU scores was not ready at submission time.
Table 2 is 10 BLEU points below its counterpart with bilingual embeddings (M07), but the training was far from converging even when running for more days.

As expected, increasing the size of the corpus and the number of parameters of the architecture is beneficial for the final translation quality. The former has the only disadvantage of needing more time and computing power. The latter even if achieving around 2 BLEU points of improvement (M04 vs. M05; M08 vs. M09) does not allow us to use document level information during training for part of the data.

An ensemble of different high performing models showed better results than the combination of the last check-points of the best model. Different combinations are reported in Table 2, all of them using a beam search of size 10 which also performed better than the default value of 6. The best ensemble comes from the combination of the four best performing individual models, but unfortunately the two best performing models were not ready at submission time. M11 and M12 are the same as M09 and M10 before convergence and were the ones used in the ensembled translation as our primary submission.

3 English–Gujarati Systems

3.1 Corpus

Monolingual corpora. The monolingual corpora were used mainly as additional data for training word-embeddings in en and gu. For English we use the same NewsCrawl selection as for en–de (ssNewsCrawl). For Gujarati we use the 2018 version of NewsCrawl and CommonCrawl.

To further increase the available data size for training Gujarati embeddings as well as to add similar content to the English word embeddings, we crawled additional Gujarati news pages and, if existent, their English counterparts. This yielded an increase of about 2 M monolingual Gujarati sentences. While crawling for the news articles, articles written during the period from which the test corpus newsdev2019 was created\(^8\) were not included in the creation of these data sets. The number of sentences and tokens extracted from each news outlet is shown in Table 3.

Wikipedia (WP) is a popular source for comparable documents. In order to later extract parallel sentences from it, the WP dumps\(^9\) for English and Gujarati are downloaded. Only the subset of articles that are linked across both languages using Wikipedia’s langlinks are extracted. That is, an article is only taken into account if there is a linked article in the other language. For these purposes, we use WikiTailor (Barrón-Cedeño et al., 2015)\(^10\) to obtain the intersection of articles of both languages. We additionally use the en–gu WP reference which was made available for WMT 2019. The monolingual WP in Gujarati is added to the monolingual data for training the embeddings.

Parallel corpora. We use the concatenation of several parallel corpora available for the en–gu news translation task to train the base model. Firstly, the bible corpus\(^11\) as well as two corpora specially made for WMT2019\(^12\) are used, namely a crawled corpus (WMT19 Crawl) and a localisation corpus extracted from OPUS\(^13\) (WMT Localisation). Lastly, the Translation Quality Estimation (TQE) dataset for Indian languages (Nisarg et al., 2018), which essentially is the concatenation of two corpora by the Indian Languages Corpora Initiative, which focus on the health and tourism domain each. For development, we use the first 999 sentences from the English-Gujarati version of newsdev2019. Further, we report results on the final newsdev2019 corpus.

Pre-processing. All English corpora (excluding the evaluation corpora) undergo the same pre-processing. After being sentence split, the corpora are normalized, tokenized and truecased using standard Moses scripts (Koehn et al., 2007a). A byte-pair-encoding (BPE) (Sennrich et al., 2016b) of 40 \(k\) merge operations trained jointly on en–gu data respectively is applied accordingly. Duplicates are removed and sentences with more than 50 tokens are discarded. In order to enable a multilingual setup, language tokens indicating the designated target language are prepended to each source sentence. As the English–Gujarati setting is bilingual, this reduces to each Gujarati sentence starting with the language token <en>, and each English sentence with <gu>.

Gujarati corpora are normalized and romanized

---

\(^{9}\)Downloaded from https://dumps.wikimedia.org/ on January 2019.

\(^{10}\)https://github.com/cristinae/WikiTailor

\(^{11}\)http://christos-c.com/bible

\(^{12}\)http://www.statmt.org/wmt19/translation-task.html

\(^{13}\)http://opus.nlpl.eu/
Table 3: Size of the corpora used for the \textit{en–gu} models.

<table>
<thead>
<tr>
<th>Corpus Type</th>
<th>Language</th>
<th># Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolingual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssNewsCrawl</td>
<td>en</td>
<td>176,220,479</td>
</tr>
<tr>
<td>CommonCrawl</td>
<td>gu</td>
<td>3,729,406</td>
</tr>
<tr>
<td>NewsCrawl</td>
<td>gu</td>
<td>244,919</td>
</tr>
<tr>
<td>WP Edition</td>
<td>gu</td>
<td>4,280,531</td>
</tr>
<tr>
<td>Crawled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Divya Bhaskar</td>
<td>gu</td>
<td>563,072</td>
</tr>
<tr>
<td>News18 en</td>
<td></td>
<td></td>
</tr>
<tr>
<td>News18 gu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gujarati Samachar</td>
<td>gu</td>
<td>121,349</td>
</tr>
<tr>
<td>Sandesh gu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zeenews en</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zeenews gu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bible en–gu</td>
<td></td>
<td>7,807</td>
</tr>
<tr>
<td>WMT19 Crawl en–gu</td>
<td></td>
<td>10,650</td>
</tr>
<tr>
<td>WMT19 Localisation en–gu</td>
<td></td>
<td>107,637</td>
</tr>
<tr>
<td>TQE en–gu</td>
<td></td>
<td>50,000</td>
</tr>
<tr>
<td>WP Reference en–gu</td>
<td></td>
<td>18,033</td>
</tr>
<tr>
<td>Comparable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WP Comparable en</td>
<td></td>
<td>546,924</td>
</tr>
<tr>
<td>WP Comparable gu</td>
<td></td>
<td>143,120</td>
</tr>
</tbody>
</table>

14 The romanized corpora are then tokenized using Moses. As the romanization is case sensitive, no true-casing is performed. The shared BPE is applied.

15 https://github.com/tmikolov/word2vec

16 https://github.com/artetxem/vecmap

to provide a first model for back-translations as well as to train the final model submitted, the transformer is used in-between to extract additional data from Wikipedia.

The transformer is trained using OpenNMT-py \cite{Klein2017} and is defined as follows: 6-layer encoder-decoder with 8-head self-attention and 2048-dim hidden feed-forward layers. Adam optimization with $\lambda=2$ and $\beta_2=0.998$; noam learning rate decay (as defined in \cite{Vaswani2017}) with 8000 warm-up steps. Labels are smoothed ($\epsilon=0.1$) and a dropout mask ($p=0.1$) is applied. As is common for transformers, position encodings and Xavier parameter initialization \cite{GlorotBengio2010} are used.

3.3 Statistical Machine Translation System

The second family of systems we use in this setting is statistical machine translation (SMT). We expect these systems to perform better when the number of parallel sentences is small. SMT systems are trained using standard freely available software. We estimate a 5-gram or 4-gram language model using interpolated Kneser–Ney discounting with SRILM \cite{Stolcke2002} depending on the language and the size of the monolingual corpus. Word alignment is done with GIZA++ \cite{OchNey2003} and both phrase extraction and decoding are done with the Moses package \cite{Koehn2007b}. The optimisation of the feature weights of the model is done with Minimum Error Rate Training (MERT) \cite{Och2003} against the BLEU \cite{Papineni2002} evaluation metric. Our model considers the language model, direct and inverse phrase probabilities, direct and inverse lexical probabilities, phrase and word penalties, and a lexicalised reordering.

3.4 Results

We train our SMT and NMT in four steps, yielding the following models:

1. SMT\textsubscript{base}: Train an SMT model on the concatenation of all parallel training data listed in Table 3 ($\sim194k$ pairs). This is then used to back-translate $4k$ ($2k$ per language direction) pairs of the monolingual data available.

2. NMT\textsubscript{extract}: Initialize Transformer with the pre-trained word-embeddings. The transformer is used to extract additional data from \textit{en–gu} Wikipedias as well as the crawled
Table 4: BLEU scores achieved on the internal development set and the official newstest2019. Scores on the development set are calculated using multi-bleu on the tokenized outputs, while the results on newstest2019 are those calculated by the WMT matrix. Primary system submissions are in bold.

<table>
<thead>
<tr>
<th></th>
<th>BLEU dev</th>
<th>BLEU nt2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>en2gu</td>
<td>gu2en</td>
</tr>
<tr>
<td>NMT\text{extract}</td>
<td>4.65</td>
<td>10.64</td>
</tr>
<tr>
<td>SMT\text{base}</td>
<td>8.77</td>
<td>12.90</td>
</tr>
<tr>
<td>SMT\text{extract}</td>
<td>9.15</td>
<td>13.08</td>
</tr>
<tr>
<td>SMT\text{all}</td>
<td>8.93</td>
<td>14.08</td>
</tr>
</tbody>
</table>

Zeenews and News18 articles. It is also used to filter the back-translations produced by SMT\text{base} as well as the parallel corpus available. The extraction is performed using the joint NMT learning and extraction framework described in Ruiter et al. (2019). There, we use the margin-based function (Artetxe and Schwenk, 2018) for scoring both word embedding and hidden-state representations. This results in an extracted and filtered corpus of $\sim 275 \, k$ sentences; a slight increase to the original parallel data available to us despite the filtering of less useful pairs.

3. SMT\text{extract}: SMT model, trained on the corpus that resulted from the extraction and filtering performed by NMT\text{extract}.

4. SMT\text{all}: SMT model, trained on both the extracted and filtered corpus by NMT\text{extract}, as well as the parallel data available, resulting in $\sim 475 \, k$ training pairs used.

Due to time constraints we could not apply any system combination technique on the individual systems. However, due to the big gap in performance between SMT and NMT we do not expect significant improvements.

Table 4 shows translation quality as measured by BLEU for both the neural and statistical systems with the different data configurations. The filtering and extraction performed by NMT\text{extract} led to a small increase in BLEU for SMT\text{extract} and SMT\text{all}, indicating that the filtering was based on positive decisions. However, when taking into account that the average number of extracted pairs from WP was steadily around 1.6 $k$ pairs, and comparing them with the 18 $k$ pairs in the en–gu WP reference, it becomes clear that extraction did not obtain high recall. This is most likely due to three difficulties that the system encounters in this setting: i) Not enough comparable data was available to adapt the internal representations (word embeddings and hidden states) to the data, meaning that the extraction performance, which is bound to the extraction decisions of the representations, stays below its potential. ii) The lack of monolingual data to train high-quality gu embeddings as well as iii) the rareness of homographs in this rather distant language pair makes the initialization difficult. Extraction in the first epochs is usually dependent on such homographs and a lack thereof reduces the number of identifiable pairs in the initialization phase of the model.

4 Conclusions

We presented two approaches for the WMT 2019 news translation shared task. We participated in the en2de task with a data-based coreference-aware NMT system. The corpus is enriched with this document-level information at sentence level so that the standard training procedure can be used. However, the amount of data we can use is smaller than in the standard pipeline and therefore the global quality can be damaged. We expect the manual evaluation to show improvements on the tackled phenomena such as gender translation.

For the en–gu task, we used a NMT architecture that can be trained on comparable corpora. In this case we downloaded news web pages as well as linked Wikipedia articles in Gujarati and English to extract and train on. Our experiments show that very few sentences could be used from this corpus and our results are close to the baseline one can get with the available parallel resources. Given the final amount of data, our state-of-the-art SMT system performed clearly better than our NMT one.

Acknowledgments

The project on which this paper is based was partially funded by the German Federal Ministry of Education and Research under the funding code 01IW17001 (Deeplee) and by the Leibniz Gemeinschaft via the SAW-2016-ZPID-2 project (CLuBS). Responsibility for the content of this publication is with the authors.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost)


