The ADAPT System Description for the STAPLE 2020 English-to-Portuguese Translation Task

Rejwanul Haque, Yasmin Moslem and Andy Way
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
firstname.lastname@adaptcentre.ie

Abstract

This paper describes the ADAPT Centre's submission to STAPLE (Simultaneous Translation and Paraphrase for Language Education) 2020, a shared task of the 4th Workshop on Neural Generation and Translation (WNGT), for the English-to-Portuguese translation task. In this shared task, the participants were asked to produce high-coverage sets of plausible translations given English prompts (input source sentences). We present our English-to-Portuguese machine translation (MT) models that were built applying various strategies, e.g. data and sentence selection, monolingual MT for generating alternative translations, and combining multiple *n*-best translations. Our experiments show that adding the aforementioned techniques to the baseline yields an excellent performance in the English-to-Portuguese translation task.

1 Introduction

The ADAPT Centre participated in STAPLE[1] (Mayhew et al., 2020), a shared task of the 4th WNGT which will be held at ACL 2020,[2] in the English-to-Portuguese language direction. The task focuses on a specific use case of MT, i.e. generating many possible translations for a given input text. Such situations are usually seen on language-learning platforms (e.g. Duolingo) where the learning process includes translation-based exercises, and evaluation is done by comparing learners' responses with a large set of human-curated acceptable translations. The shared task organisers (Duolingo) have released real language-learner data of Duolingo as training examples. We applied a number of strategies to our MT system building process, e.g. monolingual MT, extracting parallel sentences that are similar to the Duolingo's real language-learner data from the freely available external parallel corpora, assembling *n*-best translations from multiple translation systems, which essentially led us to generate high-coverage sets of possible translations of the English prompts (input source sentences).

The remainder of the paper is organised as follows: Section 2 explains our approaches; Section 3 details of the datasets used, explains the experimental setups and presents the results with some discussions; and Section 4 concludes our work with avenues for future work.

2 Methodology

2.1 Selecting External Datasets

Since the shared task organisers released training data with a limited number of prompts (only 4,000 English prompts for English-to-Portuguese translation) and allowed participants to use external data, we made use of parallel corpora from a variety of existing sources, e.g. OPUS[3] (Tiedemann, 2012). First, we found out which corpora are similar to Duolingo's training dataset. For this, we measured perplexity of the source and target texts of the external datasets on the in-domain language models (LMs) (i.e. LMs were built on the Duolingo's data). We selected those corpora whose sentences are found to be more similar to those of Duolingo's language learning data.

2.2 Selecting 'Pseudo In-domain' Parallel Sentences from External Data

The state-of-the-art sentence selection method of Axelrod et al. (2011) is used to extract 'pseudo in-domain' data from large corpora using bilingual cross-entropy difference. The extracted data is usually used to train domain-specific MT systems or to fine-tune generic MT systems. We con-

[1] https://sharedtask.duolingo.com/
[2] https://acl2020.org/
[3] http://opus.lingfil.uu.se/

sider Duolingo's language learning data released in this shared task as the real in-domain data. In this task, we took each of the external parallel corpora (usually large in size) chosen following the approach described in Section 2.1, and selected top *n* sentence-pairs as per low cross-entropy differences over each side of the corpus (source and target) following Axelrod et al. (2011). This provided us with a 'pseudo in-domain' corpus (i.e. the extracted top *n* sentence-pairs) whose sentences are similar to the sentences of the Duolingo's data in terms of domain and style. We appended the extracted 'pseudo in-domain' data to the STAPLE's (Duolingo's) training data for building different MT systems which are described latter in the paper (cf. Section 3.5).

2.3 Same-language MT

Same-language MT has been successfully used in many NLP applications, e.g. text-to-speech synthesis for creating alternative target sequences (Cahill et al., 2009), translation between varieties of the same language (Brazilian Portuguese to European Portuguese) (Fancellu et al., 2014), paraphrase generation (Plachouras et al., 2018), and producing many alternative sequences of a given input question in question answering (Bhattacharjee et al., 2020). In our case, we developed Portuguese-to-Portuguese MT systems that were able to generate *n*-best (same-language) alternative sentences of an input Portuguese sentence. Using this monolingual MT systems, we could obtain a set of alternative sequences of a given Portuguese translation.

As mentioned earlier, the Duolingo training data includes a high-coverage set of Portuguese translations of the English prompts. We generated a set of source–target pairs (Portuguese-to-Portuguese) from each of the high-coverage sets of alternative Portuguese translations. This served as our training data for Portuguese-to-Portuguese MT system building. Additionally, we used an existing paraphrasing resource (Ganitkevitch and Callison-Burch, 2014) for Portuguese and appended that to the training data.

2.4 Combining multiple *n*-best Translations

In this work, we built a number of MT systems using the state-of-the-art phrase-based statistical MT (PB-SMT) (Koehn et al., 2003) and neural MT (NMT) (Vaswani et al., 2017) approaches. The *n*-best translations produced by the different MT systems (i.e. a PB-SMT and several NMT systems) are combined adopting a variety of approaches (e.g. majority voting) to produce the final sets of translations of the English prompts.

3 Experiments and Results

3.1 The MT system setups

As pointed out earlier, we chose the classical PB-SMT and emerging NMT paradigms for building our MT systems. To build our PB-SMT systems, we used the Moses toolkit (Koehn et al., 2007). We used a 5-gram LM trained with modified Kneser-Ney smoothing (Kneser and Ney, 1995) using the KenLM toolkit (Heafield et al., 2013). Our PB-SMT log-linear features include: (a) 4 translational features (forward and backward phrase and lexical probabilities), (b) 8 lexicalised reordering probabilities (*wbe-mslr-bidirectional-fe-allff*), (c) 5-gram LM probabilities, (d) 5 OSM features (Durrani et al., 2011), and (e) word-count and distortion penalties. In our experiments, word alignment models are trained using the GIZA++ toolkit[4] (Och and Ney, 2003), phrases are extracted following the *grow-diag-final-and* algorithm of Koehn et al. (2003), Kneser-Ney smoothing is applied at phrase scoring, and a smoothing constant (0.8u) is used for training lexicalised reordering models. The weights of the parameters are optimised using the margin-infused relaxed algorithm (Cherry and Foster, 2012) on the development set. For decoding, the cube-pruning algorithm (Huang and Chiang, 2007) is applied, with a distortion limit of 12.

To build our NMT systems, we used the Marian-NMT (Junczys-Dowmunt et al., 2018) toolkit. The NMT systems are Transformer models (Vaswani et al., 2017). In our experiments, we followed the recommended best set-up from Vaswani et al. (2017). The tokens of the training, evaluation and validation sets are segmented into sub-word units using the Byte-Pair Encoding (BPE) technique (Sennrich et al., 2016). We performed 32,000 join operations. Our training set-up is as follows.

We consider the size of the encoder and decoder layers to be 6. As in Vaswani et al. (2017), we employ residual connection around layers (He et al., 2015), followed by layer normalisation (Ba et al., 2016). The weight matrix between the embedding layers is shared, similar to Press and Wolf (2016). Dropout (Gal and Ghahramani, 2016) between layers is set to 0.10. We use mini-batches of

[4] http://www.statmt.org/moses/giza/GIZA++.html

size 64 for updating. The models are trained with the Adam optimizer (Kingma and Ba, 2014), with the learning-rate set to 0.0003 and reshuffling the training corpora for each epoch. As in Vaswani et al. (2017), we also use the learning rate warm-up strategy for Adam. Validation on the development set is performed using three cost functions: cross-entropy, perplexity and BLEU. The early-stopping criteria is based on cross-entropy while the final NMT system is selected as per the highest BLEU score on the validation set. The beam size for search is set to 12.

3.2 The Shared Task Data

The data released by STAPLE is compiled from Duolingo's language learning courses. The training data for English-to-Portuguese translation contains 4,000 English prompts, with multiple Portuguese translations, from which we obtained a total of 526,467 source–target segment-pairs. We randomly sampled 2,000 sentence-pairs from the training set, and considered them as development set. As for the development set, we chose the highest scoring Portuguese translations of the English prompts. The development set (blind) released by STAPLE contains 500 English prompts. They also provided a high-quality automatic reference translations of the development set sentences by Amazon Translate.[5] We considered the development set (with translations by Amazon translate as references) as our test set in order to evaluate our MT systems. We removed those entries from the training set (526,467 source–target segment-pairs) that overlap with entries (source or target counterparts) of the development and test sets. The training set is left with 258,306 source–target segment-pairs after discarding the overlapping entries. The statistics of training, development and test set sentences are shown in Table 1.

	sentences	words (en)	words (pt)
train set	258,306	2,063,108	2,128,044
dev set	2,000	14,557	14,196
test set	500	3,551	3,322

Table 1: The shared task data statistics.

3.3 The baseline MT systems

We first built MT systems with only the data provided by the shared task (cf. Table 1). We computed the BLEU (Papineni et al., 2002) score to evaluate the MT systems on the test set, which are reported in Table 2. Note that we used translations by Amazon Translate provided by STAPLE as the reference translations, which are excellent in quality. Thus, the BLEU scores on the test set can provide indications how good or bad our MT systems are. Additionally, we have reported the MT systems' BLEU scores on the development set. As can be seen from Table 2, the BLEU scores of the MT systems are very low. These scores were expected given the (small) number of sentences used for training. Interestingly, PB-SMT outperforms NMT by a large margin in terms of BLEU, and this can happen in low-resource scenarios (Koehn and Knowles, 2017).

	BLEU	
	dev set	test set
PB-SMT	22.69	19.92
Transformer	9.57	9.23

Table 2: The BLEU scores of baseline MT systems.

3.4 The External Datasets Used

Since we (participants) are allowed to use external data, we decided to use freely available bilingual corpora whose sentences are similar to those of the Duolingo's English–Portuguese dataset. We took all bilingual corpora available in the OPUS repository, and measured perplexity of source and target texts on in-domain LMs (built on the Duolingo data only). We found that the most similar corpora to the English-side of the Duolingo's training corpus are OpenSubtitles[6] and Tatoeba[7] and to the Portuguese-side of that are Books[8] and Tatoeba. In addition to Tatoeba, Books and OpenSubtitles, we made use of ParaCrawl (parallel sentences crawled from Web)[9] and Wikipedia (parallel sentences extracted from Wikipedia)[10] which were found to be moderately similar to the task data according to the LM perplexity scores. Additionally, we also used several generic corpora for building an NMT

[5] https://aws.amazon.com/translate/

[6] http://opus.nlpl.eu/OpenSubtitles-v2018.php

[7] http://opus.nlpl.eu/Tatoeba-v20190709.php

[8] http://opus.nlpl.eu/Books-v1.php

[9] http://opus.nlpl.eu/ParaCrawl-v5.php

[10] http://opus.nlpl.eu/Wikipedia-v1.0.php

system, EUconst,[11] JRC-Acquis,[12] Europarl[13] and DGT.[14]

We manually looked at these datasets, and observed that the Tatoeba and Books corpora are good in quality. We directly used sentences of the Tatoeba and Books corpora for system building. In contrast, the OpenSubtitles corpus seems to have considerable noise intact in it. We also noticed that a corpus of one language (say, English) contains sentences of other languages, so we use a language identifier[15] in order to remove such noise. We applied a number of standard cleaning routines for removing noisy sentences, e.g. removing sentence-pairs that are too short, too long or which violate certain sentence-length ratios. The latter processing was applied to all corpora. In order to perform tokenisation for English and Portuguese, we used the standard tool[16] in the Moses toolkit.

As for the selection of "pseudo in-domain" sentence-pairs from the external bilingual corpora using bilingual cross-entropy difference measure (Axelrod et al., 2011) (cf. Section 2.2), we applied the strategy to every corpus except Tatoeba and Books. The so-called "pseudo in-domain" parallel sentences that were extracted from the out-of-domain data were appended to the in-domain (shared task) training data in order to build the MT systems.

3.5 The MT systems built using external datasets

3.5.1 The PB-SMT systems

This section presents the PB-SMT systems that were built on the training data augmented by appending external data (cf. Section 3.4) to the shared task data. As stated earlier, the performance of the PB-SMT systems on the development set and test sets in terms of BLEU are shown in Table 3. The second row of Table 3 represents the MT system built on the training data (Duo + Books + Tatoeba) that includes sentences of the Duolingo's training data and of two external corpora: Books and Tatoeba. We see that this MT system surpassed the baseline PB-SMT system (cf. Table 2) by a large margin in terms of BLEU.

		dev set	test set
(a)	Baseline	22.69	19.92
(b)	Duo + Books + Tatoeba	53.18	49.42
(c)	(b) + 3M ParaWiki	57.80	55.27

Table 3: The BLEU scores of the PB-SMT systems. Duo: Duolingo's training data.

We merged the ParaCrawl and Wikipedia corpora, and from now on, we call the combined corpus ParaWiki. We took low-scoring (bilingual cross-entropy difference) (cf. Section 2.2) sentence-pairs from ParaWiki, added them to the training data (i.e. Duo + Books + Tatoeba) in various proportions, and built PB-SMT systems on them. The BLEU scores of the best system out of the PB-SMT systems that were built on these sets of training data on the development and test sets are shown in the last row of Table 2. As illustrated by the table, sentences of ParaWiki have a positive impact on the system's performance since we get further gains in terms of BLEU. In short, our best-performing PB-SMT system is the one that is built on training data compiled by three million sentences from ParaWiki, all sentences of the Books and Tatoeba corpora and the Duolingo's training data. From now on, this PB-SMT system is referred as PB-BEST. In this context, we also carried out a number of experiments and built many PB-SMT systems by adding sentences from other data sets (e.g. OpenSubtitles) to the training data in various proportions as above. None of the setups results in any other MT system that could outperform PB-BEST.

		dev set	test set
(a)	Baseline	9.57	9.23
(b)	Duo + Books + Tatoeba	62.19	58.17
(c)	(b) + 3M ParaWiki	60.77	62.61
(d)	(b) + 6M ParaWiki	**65.69**	**67.97**
(e)	(b) + 3M Generic	46.15	43.38
(f)	(b) + 8M Sub	50.83	57.82
(g)	(d) + 8M Sub	55.73	61.07

Table 4: The BLEU scores of the NMT systems. Duo: Duolingo's training data.

3.5.2 The NMT systems

This section presents our NMT systems that were built on the training data prepared by appending

[11] http://opus.nlpl.eu/EUconst-v1.php
[12] http://opus.nlpl.eu/JRC-Acquis-v3.0.php
[13] http://opus.nlpl.eu/Europarl-v8.php
[14] http://opus.nlpl.eu/DGT-v2019.php
[15] cld2: https://github.com/CLD2Owners/cld2
[16] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

external data (cf. Section 3.4) to the shared task data. We evaluate the NMT systems on the test set, and report BLEU scores in Table 4. As above, we also show their performance (i.e. BLEU scores) on the development set. When we compare the scores of Table 3 with those of Table 4 (rows (b) and (c)), we see that NMT outperforms PB-SMT with large margins. This time, our best setup is the one when we use training data compiled by six million sentences from ParaWiki, all sentences of the Books and Tatoeba corpora and the shared task training data (cf. row (d) of Table 4). From now on, we call this MT system NEURAL-BEST.

We also built MT systems using sentences from other datasets. We merged sentences of the generic corpora (EUconst, JRC-Acquis, Europarl, DGT), selected low-scoring 3M sentence-pairs from the combined corpus (i.e. the most similar sentences to those of the Duolingo's data), and added them to the baseline training set. We can see from Table 4 (row (e)) that the the MT system built on this training data does not perform well. We call this MT system NEURAL-Generic.

As mentioned above (cf. Section 3.4), sentences of the OpenSubtitles corpus are quite similar to those of Duolingo's training corpus. To this end, we carried out a series of experiments by selecting different sizes of sentences from this dataset following the approach described in Section 2.2. We found that this dataset does have much impact on our system building (cf. row (f) of Table 4). In fact, we see from the last row of Table 4 that inclusion of a part of the OpenSubtitles corpus to the training set of the best setup deteriorates the system's performance. We call this MT system (row (g)) NEURAL-OpenSub.

3.6 The Portuguese-to-Portuguese MT Systems

In Section 2.3, we explained the purpose of creating monolingual MT systems. This section describes our Portuguese-to-Portuguese MT systems. We prepare training data for the Portuguese-to-Portuguese MT system from the high-coverage set of Portuguese translations for the English prompts. Thus, the training data contains source–target pairs whose source and target counterparts are two variations of Portuguese translation. In Table 5, we show the number of training examples used for building monolingual Portuguese MT systems. The table also shows the maximum number of variations for a Portuguese translation used for forming the training set. Additionally, we used Portuguese paraphrases from the PPDB database[17] (Ganitkevitch and Callison-Burch, 2014) as the part of the training data for this task. We also used the target-side (Portuguese) sentences of the shared task training set (cf. Table 1), and add them (i.e. identical copy) to the both sides of the training set. The first 1,000 sentences of the development set (cf. Table 1) serves as our development set and the remaining sentences of the development set serves as our test set.

	Sentences	Variations
PB-SMT	3,091,264	30 (max.)
NMT	12,523,886	75 (max.)
Paraphrases	16,915,010	

Table 5: Number of training examples used for the monolingual MT training.

	BLEU
PB-SMT	72.30
NMT	40.73

Table 6: The BLEU scores of the monolingual MT systems.

We obtain the BLEU scores to evaluate the monolingual MT systems on the test set, and report them in Table 6. The learning objective of the monolingual MT models is to generate alternative sequences of a language given the sentences of the same language. Therefore, these scores, to a certain extent, show how likely an MT system can produce their own versions of the Portuguese sentences. We see from Table 6 that PB-SMT outperforms NMT with a large margin in terms of BLEU.

3.7 Generating Translations of English Prompts

This section presents our translation framework that is expected to generate high-coverage sets of plausible translations given the English prompts. In the translation framework, we made use of our best system NEURAL-BEST, and two more Transformer models, NEURAL-OpenSub and NEURAL-Generic (cf. Section 3.5.2), and best SMT system PB-BEST (cf. Section 3.5.1). Note that we make each of our final NMT models with ensembles of 4 models that are sampled from the

[17] http://paraphrase.org/#/download

training run, and one of them is selected as per highest BLEU score on the validation set.

The STAPLE development and test sets are blind, and participants were asked to submit the system's output via CodaLab[18] which displays system's scores upon submission. The main scoring metric for evaluation is weighted macro F_1. The precision and recall are calculated in unweighted and weighted fashions, respectively. In short, the systems are scored based on how well they can return all human-curated acceptable translations, weighted by the likelihood that an English learner would respond with each translation.

	R	P	F_1
NEURAL-BEST	0.4325	0.6564	0.4601
NEURAL-OpenSub	0.3922	0.5542	0.3922
NEURAL-Generic	0.3224	0.5541	0.3451
PB-BEST	0.2044	0.5643	0.2512
SysCom	0.4566	0.6019	0.4620
SysCom + MMT-B1	0.4551	0.6204	0.4669
SysCom + MMT-B5	0.4656	0.6138	0.4710

Table 7: Performance of our MT systems on dev set (submitted).

In Table 7, we present the performance of our submitted MT systems on the development set. The first row of Table 7 represents our best NMT system NEURAL-BEST which is competitive and produces an F_1 score of 0.4601 on the development set. As for this system, we generate 12-best (n = 12) translations for each English prompt. We tried higher values for n and the systems with different values of n are more or less comparable to the one reported in the table. Therefore, we keep the value of n 12 for generating alternative translations for all our MT systems. The system that represents the first row of the table can be seen as our baseline.

The next three rows of Table 7 show the scores of other two neural models, NEURAL-OpenSub and NEURAL-Generic, and the best PB-SMT system PB-BEST. We see from Table 7 that these MT systems perform much worse than NEURAL-BEST as far as the F_1 scores on the development set are concerned. Interestingly, precision of PB-BEST is relatively better than the other two MT systems despite its low recall which in fact is responsible for its low final F_1. We found that many 12-best translations produced by PB-BEST are identical. This is the reason why its recall (and F_1) is too low.

The fifth row of Table 7 presents a system that combines n-best translations of different MT systems. We refer to this system as SysCom. The idea is to produce final sets of translations of the English prompts as exhaustive and precise as possible by combining 12-best translations produced by the different MT systems. We tried a number of ways for combining the translations produced by multiple systems, and the setup that worked best for us is described as follows. We took 12-best translations by NEURAL-BEST and 1-best translations by NEURAL-Generic, NEURAL-OpenSub and PB-BEST. Additionally, we also took translations from 12-best translations by NEURAL-Generic, NEURAL-OpenSub and PB-BEST with two out of three voting logic. The reason for using different MT methods and MT systems built on different data domains or styles in the system combination strategy is that such MT systems may produce some alternative translations that are to be different to each other. SysCom produces a 0.0019 F_1 point absolute (corresponding to 0.41% relative) gain over the baseline. Naturally, this approach increases the coverage of plausible translations of the English prompts, which causes a 0.0241 recall point absolute (corresponding to 5.57% relative) gain over the baseline however at the expense of precision.

The final two rows of Table 7 correspond to two system setups in which we used our monolingual Portuguese MT systems on top of the SysCom setup. In other words, we applied the monolingual Portuguese MT systems to the Portuguese translations of the English prompts in order to collect more viable alternative translations. We tried to avoid applying monolingual MT on the noisy translations generated in the previous stage (i.e. English-to-Portuguese MT). Accordingly, we adopted different setups in order to obtain translations that are to be as much error free as possible (i.e. a set of high precision n-best translations). The setup that worked best in our case is described as follows. We took 1-best (i.e. row 6 of Table 7; MMT-B1) or 5-best (i.e. row 7 of Table 7; MMT-B5) translations by both NEURAL-BEST and PB-BEST. We also took a set of translations which is an intersection of the sets of 12-best translations produced by NEURAL-BEST, NEURAL-Generic, NEURAL-OpenSub and PB-BEST. Let's call this set of translations *mono-set*.

[18] https://competitions.codalab.org/competitions/

We translated the sentences (i.e. Portuguese translations) of the mono-set using the monolingual Portuguese PB-SMT and NMT systems (cf. Section 3.6). For each sentence of mono-set we produce 12-best translations by the PB-SMT and NMT systems, take intersection of the two sets of 12-best translations, and consider the intersected set as the viable alternative translations. Nonetheless, this stage includes another screening that helps weed out as much noise as possible, which are described as follows. We used 12-best translations provided by the monolingual PB-SMT and NMT systems if and only if a certain portion of the translations by each system should appear in the output of SysCom. In other words, a ratio of the number of the overlapping translations by each of the monolingual MT systems with the translations provided by SysCom and the total number of the translations provided by SysCom should be greater than a threshold which we set to 0.2. The intersection operation on two sets of 12-best translations and the screening strategy, to a certain extent, ensures the translation variations generated by the monolingual MT systems of good quality. We see from the final two rows of Table 7 that the strategy of applying monolingual MT for generating alternative translations brings about moderate improvements in terms of F_1 over the baseline. The best setup (SysCom + MMT-B5) produces a 0.0109 F_1 point absolute (corresponding to 2.4% relative) gain over the baseline.

	R	P	F_1
SysCom + MMT-B1	0.4306	0.6379	0.4574
SysCom + MMT-B5	0.4377	0.6318	0.4597

Table 8: Performance of our MT systems on test set (submitted).

In Table 8, we show the performance of our submitted systems on the STAPLE 2020 test set. These systems in fact correspond to the two systems whose performance on the development set were presented in the last two rows of Table 7. The second row of Table 8 (SysCom + MMT-B5) represents the system whose submission earned us the fifth position in the competition.

4 Conclusion

This paper presents the ADAPT translation system for the STAPLE 2020 English-to-Portuguese Translation Task. We aimed to build a competitive translation system that can produce high-coverage sets of plausible translations given English prompts (input source sentences). For this, we applied various strategies, e.g. selecting data sources that are similar to STAPLE 2020 training data, selecting sentences from external corpora, applying monolingual MT for generating alternative translations, combining translations produced by multiple MT systems. We found that the systematic addition of these techniques to baseline yields moderate improvement over the baseline (0.0109 F_1 point absolute corresponding to 2.4% relative gain). The best experimental setup earned us the fifth position in the competition.

In the future, we aim to apply confusion network decoding in order to re-rank *n*-best translations generated by the multiple MT systems. We used monolingual MT for generating alternative sentences for the target (Portuguese) translations. We also aim to apply this strategy to the English prompts (i.e. source-side of the translation-pair) for generating alternative sequences of the input source sentences.

Acknowledgments

The ADAPT Centre for Digital Content Technology is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund. This project has partially received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 713567, and the publication has emanated from research supported in part by a research grant from SFI under Grant Number 13/RC/2077.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Santanu Bhattacharjee, Rejwanul Haque, Gideon Maillette de Buy Wenniger, and Andy Way. 2020. Investigating Query Expansion and Coreference Resolution in Question Answering on BERT. In *Pro-*

ceedings of Natural Language Processing and Information Systems - 25th International Conference on Applications of Natural Language to Information Systems, NLDB 2020, page (to appear), Saarbrücken, Germany.

Peter Cahill, Jinhua Du, Andy Way, and Julie Carson-Berndsen. 2009. Using same-language machine translation to create alternative target sequences for text-to-speech synthesis. In *Proceedings of Interspeech 2009, the 10th Annual Conference of the International Speech Communication Association*, pages 1307–1310, Brighton, UK.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1045–1054, Portland, Oregon, USA.

Federico Fancellu, Morgan O'Brien, and Andy Way. 2014. Standard language variety conversion using smt. In *Proceedings of the Seventeenth Annual Conference of the European Association for Machine Translation*, pages 143–149, Dubrovnik, Croatia.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. *CoRR*, abs/1512.05287.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The multilingual paraphrase database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4276–4283, Reykjavík, Iceland.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184 vol.1.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, Williams College, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the Interactive Poster and Demonstration Sessions*, pages 177–180, Prague, Czech Republic.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL 2003: conference combining Human Language Technology conference series and the North American Chapter of the Association for Computational Linguistics conference series*, pages 48–54, Edmonton, AB.

Stephen Mayhew, Klinton Bicknell, Chris Brust, Bill McDowell, Will Monroe, and Burr Settles. 2020. Simultaneous translation and paraphrase for language education. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Vassilis Plachouras, Fabio Petroni, Timothy Nugent, and Jochen L. Leidner. 2018. A comparison of two paraphrase models for taxonomy augmentation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 315–320, New Orleans, LA.

Ofir Press and Lior Wolf. 2016. Using the output embedding to improve language models. *CoRR*, abs/1608.05859.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, pages 2214–2218, Istanbul, Turkey.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.