Machine Translation with `parfda`, Moses, `kenlm`, `nplm`, and PRO

Ergun Biçici
ergun.bicici@boun.edu.tr
Electrical and Electronics Engineering Department, Boğaziçi University
orcid.org/0000-0002-2293-2031

Abstract

We build `parfda` Moses statistical machine translation (SMT) models for most language pairs in the news translation task. We experiment with a hybrid approach using neural language models integrated into Moses. We obtain the constrained data statistics on the machine translation task, the coverage of the test sets, and the upper bounds on the translation results. We also contribute a new testsuite for the German-English language pair and a new automated key phrase extraction technique for the evaluation of the testsuite translations.

1 Introduction

Parallel feature weight decay algorithms (`parfda`) (Biçici, 2018) is an instance selection tool we use to select training and language model instances to build Moses (Koehn et al., 2007) phrase-based machine translation (MT) systems to translate the test sets in the news translation task at WMT19 (Bojar et al., 2019). The importance of `parfda` increase with the increasing size of the parallel and monolingual data available for building SMT systems. In the light of last year's evidence that shows that `parfda` phrase-based SMT can obtain the $2nd$ best results on a testsuite in the English-Turkish language pair (Biçici, 2018) when generating the translations of key phrases that are important for conveying the meaning, we obtain phrase-based Moses results and its extension with a neural LM in addition to the n-gram based LM that we use. We experiment with neural probabilistic LM (NPLM) (Vaswani et al., 2013). We record the statistics of the data and the resources used.

Our contributions are:

- a test suite for machine translation that is out of the domain of news task to take the chance of taking a closer look at the current status of SMT technology used by the task participants when translating 38 sentences about international relations concerning cultural artifacts,
- `parfda` Moses phrase-based MT results and data statistics for the following translation directions:
  - English-Czech (en-cs)
  - English-Finnish (en-fi), Finnish-English (fi-en),
  - English-German (en-de), German-English (de-en),
  - English-Kazakh (en-kk), Kazakh-English (kk-en),
  - English-Lithuanian (en-lt), Lithuanian-English (lt-en),
  - English-Russian (en-ru), Russian-English (ru-en),
- upperbounds on the translation performance using lowercased coverage to identify which models used data in addition to the parallel corpus.

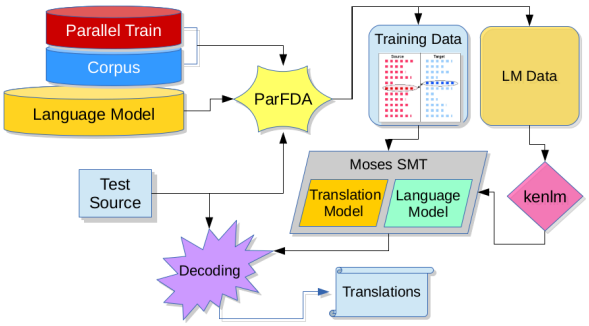

Figure 1: `parfda` Moses SMT workflow.

The sections that follow discuss the instance selection model (Section 2), the machine translation model (Section 3), the testsuite used for evaluating MT in en-de and de-en, and the results.

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 122–128
Florence, Italy, August 1-2, 2019.

$S \rightarrow T$	Data	Training Data #word S (M)	#word T (M)	#sent (K)	tcov	LM Data #word (M)	tcov
en-cs	C	587.2	659.8	44436	0.758	1439.6	0.835
en-cs	parfda	111.4	98.4	2474	0.693	371.3	0.779
en-de	C	832.6	879.0	39959	0.792	4252.0	0.864
en-de	parfda	139.0	130.7	2467	0.736	450.8	0.795
de-en	C	879.0	832.6	39959	0.865	12382.8	0.92
de-en	parfda	132.6	141.3	2441	0.827	487.8	0.871
en-fi	C	96.2	125.3	5657	0.528	1598.9	0.746
en-fi	parfda	73.9	56.1	2168	0.512	419.1	0.676
fi-en	C	130.1	100.4	6254	0.783	12382.8	0.926
fi-en	parfda	51.1	66.4	2021	0.771	416.8	0.869
en-kk	C	1.6	1.9	204	0.262	173.5	0.576
en-kk	parfda	1.9	1.5	202	0.242	175.0	0.576
kk-en	C	1.9	1.6	204	0.591	12382.8	0.907
kk-en	parfda	1.5	1.9	202	0.584	337.7	0.835
en-lt	C	38.2	45.0	2191	0.532	1523.4	0.539
en-lt	parfda	45.0	38.2	2191	0.532	310.7	0.539
lt-en	C	45.0	38.2	2191	0.794	12382.8	0.933
lt-en	parfda	34.1	40.5	1877	0.754	383.5	0.89
en-ru	C	212.0	181.9	9296	0.738	11459.4	0.888
en-ru	parfda	92.3	80.0	2260	0.713	469.0	0.803
ru-en	C	181.7	211.8	9287	0.857	12382.8	0.937
ru-en	parfda	78.2	90.5	2212	0.839	437.0	0.894

Table 1: Statistics for the training and LM corpora in the constrained (C) setting compared with the `parfda` selected data. #words is in millions (M) and #sents in thousands (K). tcov is target 2-gram coverage.

	scov					tcov				
	1	2	3	4	5	1	2	3	4	5
en-cs	0.9762	0.8399	0.5686	0.2809	0.1085	0.9792	0.7557	0.3985	0.1646	0.0618
en-de	0.9673	0.8683	0.6288	0.3301	0.1296	0.96	0.7916	0.5102	0.2438	0.0898
en-fi	0.9535	0.779	0.4829	0.2122	0.0745	0.9009	0.5283	0.2337	0.0849	0.0229
en-kk	0.8399	0.4643	0.1623	0.0363	0.0075	0.7404	0.262	0.0648	0.0104	0.0017
en-lt	0.9519	0.7214	0.3896	0.1374	0.0355	0.909	0.5324	0.2125	0.0663	0.0156
en-ru	0.9743	0.8251	0.5362	0.2434	0.0813	0.9606	0.7384	0.4102	0.1794	0.0673

Table 2: Constrained training data lowercased source feature coverage (scov) and target feature coverage (tcov) of the test set for n-grams.

2 Instance Selection with `parfda`

`parfda` parallelize feature decay algorithms (FDA) (Biçici and Yuret, 2015), a class of instance selection algorithms that decay feature weights, for fast deployment of accurate SMT systems. Figure 1 depicts `parfda` Moses SMT workflow.

We use the test set source sentences to select the training data and the target side of the selected training data to select the LM data. We decay the weights for both the source features of the test set and the target features that we already select to increase the diversity. We select about 2.2 million instances for training data and about 12 million sentences for each LM data not including the selected training set, which is added later. Table 1 shows size differences with the constrained dataset (C).[1] We use 3-grams to select training data and 2-grams for LM data and split the hyphenated words using the "`-a`" option of the tokenizer used in Moses (Sennrich et al., 2017). tcov lists the target coverage in terms of the 2-grams of the test set. The maximum sentence length is set to 126. Table 2 lists the lowercased coverage of the test set by the constrained training data of WMT19.

3 Machine Translation with Moses, `kenlm` and `nplm`, and PRO

We train 6-gram LM using `kenlm` (Heafield et al., 2013). For word alignment, we use mgiza (Gao and Vogel, 2008) where GIZA++ (Och and Ney, 2003) parameters set max-fertility to 10, the number of iterations to 7,5,5,5,7 for IBM models 1,2,3,4, and the HMM model, and learn 50 word classes in three iterations with the mkcls tool during training. We use "`-mbr`" option when decoding the test set.[3] The development set con-

[1] Available at `https://github.com/bicici/parfdaWMT2019`

[3] As practiced in the parallel corpus filtering task `http://www.statmt.org/wmt19/`

BLEU	de-en	fi-en	kk-en	lt-en	en-cs	en-de	en-fi	en-kk	en-lt
kenlm	0.309	0.202	0.105	0.225	0.152	0.235	0.127	0.029	
nplm	0.292	0.18		0.215	0.142		0.119	0.029	0.073
bilingual nplm			0.102					0.03	
kenlm + nplm	0.307			0.226	0.156	0.238		0.03	0.078
kenlm with hyphen splitting	0.3074	0.2024	0.0999	0.2245	0.1522	0.2395	0.1294	0.03	0.0828

Table 3: parfda BLEU cased results with different LM on text that is not hyphen splitted compared with after hyphen splitting.

BLEU	de-en	fi-en	kk-en	lt-en	ru-en	en-cs	en-de	en-fi	en-kk	en-lt	en-ru
parfda	0.3074	0.2024	0.0999	0.2245	0.3179	0.1522	0.2395	0.1294	0.03	0.0828	0.1846
topC	0.428	0.33	0.305	0.365	0.401	0.299	0.449	0.274	0.111	0.191	0.363
- parfda											
avg diff	0.1405										

Table 4: parfda results compared with the top results in WMT19 and their difference.[2]

tains up to 5000 sentences randomly sampled from previous years' development sets (2013-2018) and remaining come from the development set for WMT19. We obtain robust optimization results using monotonically increasing n-best list size in the beginning of tuning with pairwise ranking optimization (PRO) (Hopkins and May, 2011; Biçici, 2018). This allows us to find parameters whose tuning score reach 1% close to the best tuning parameter set score in only 4 iterations but we still run tuning for 21 iterations. Truecasing updates the casing of words according to the most common form. We truecase the text before building the SMT model as well as after decoding and then detruecase before preparing the translation, which provided better results than simply detruecasing after decoding (Biçici, 2018).

We trained nplm LM in 10 epochs. We also experimented with bilingual nplm, which uses nplm in a bilingual setting to use both the source and the target context and builds a LM on the training set (Devlin et al., 2014). Both nplm and bilingual nplm can be used with Moses as a feature within its configuration file.[4] On average, results in Table 3 shows that using only nplm decrease the scores and improvements are obtained when both nplm and kenlm are used. However, the gain from splitting hyphenated words is more and it is a less computationally demanding option. kenlm takes about 20 minutes whereas building a single nplm model took us 11.5 to 14.25 days or 1000 times longer and it takes about 56 GB space on the disk.

parfda results at WMT19 are in Table 4 using BLEU over tokenized text where we compare with the top constrained submissions (topC). All top models use NMT in 2019 and most use back-translations, which means that their tcov is upper bounded by LM tcov. topC is 14.05 BLEU points on average better than parfda in 2019 and the difference was 12.88 in 2018.

4 Translation Upper Bounds with tcov

We obtain upper bounds on the translation performance based on the target coverage (tcov) of n-grams of the test set found in the selected parfda training data using lowercased text. For a given sentence T', the number of OOV tokens are identified:

$$OOV_r = \texttt{round}((1 - \text{tcov}) * |T'|) \quad (1)$$

where $|T'|$ is the number of tokens in the sentence. We obtain each bound using 500 such instances and repeat for 10 times. tcov BLEU bound is optimistic since it does not consider reorderings in the translation or differences in sentence length. Each plot in Figure 2 locates tcov BLEU bound obtained from each n-gram and from n-gram tcovs combined up to and including n and ■ locates the parfda result and ★ locates the top constrained result. Based on the distance between the top BLEU result and the bound, we can obtain a sorting of the difficulty of the translation directions in Table 5.

5 German-English Testsuite

We prepared a MT test suite that is out of the domain of news translation task to take a closer

parallel-corpus-filtering.html

[4] http://www.statmt.org/moses/?n=FactoredTraining.BuildingLanguageModel#ntoc32

[4] We use the results from matrix.statmt.org.

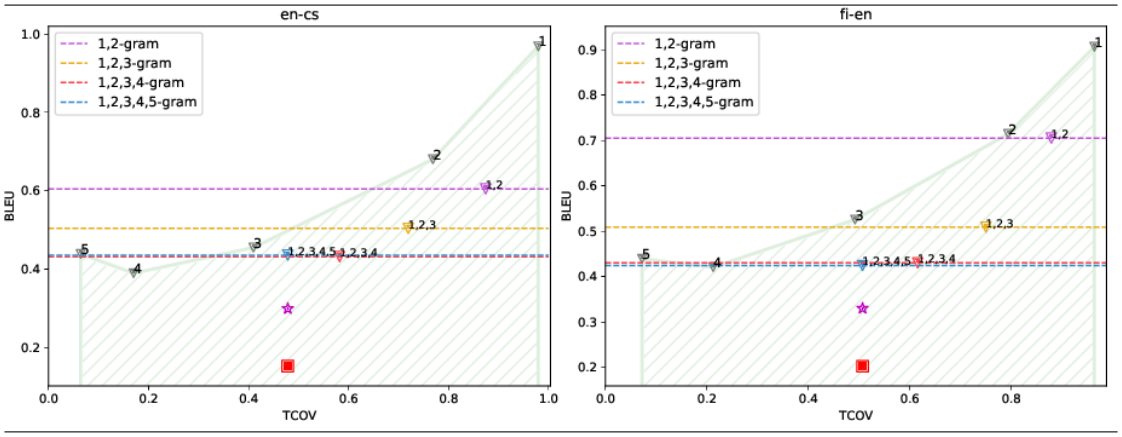

Figure 2: `parfda` results (■) and OOV_r tcov BLEU upper bounds for kk and lt.

BLEU distance	translation direction
0.0041	en-de
0.0092	en-kk
0.0277	en-ru
0.0296	en-fi
0.0372	de-en
0.0407	kk-en
0.0594	lt-en
0.0722	en-lt
0.0849	ru-en
0.0943	fi-en
0.1365	en-cs

Table 5: Difficulty of translation directions based on the distance of the top result to the upper bound.

			bits	% info.
word	order	$\log_2 25!$	83.7	16.8%
	choice	$\log_2 10^{125}$	415.2	83.2%
	total	$\log_2 25! \times 10^{125}$	498.9	100.0%
phrase	order	$\log_2 10!$	21.8	8.6%
	choice	$\log_2 10^{70}$	232.5	91.4%
	total	$\log_2 10! \times 10^{70}$	254.3	100.0%
BPE	order	$\log_2 50!$	214.2	24.4%
	choice	$\log_2 10^{200}$	664.4	75.6%
	total	$\log_2 50! \times 10^{200}$	878.6	100.0%
char	order	$\log_2 125!$	695.2	80.7%
	choice	$\log_2 10^{50}$	166.1	19.3%
	total	$\log_2 125! \times 10^{50}$	861.3	100.0%

Table 6: Information contribution from granular parts of a sentence.

look at the current status of SMT technology used by the task participants to translate 38 sentences about international relations concerning cultural artifacts in German and English. The sentences and their translations are available at `https://github.com/bicici/SMTData` sourced from the press releases of the Prussian Cultural Heritage Foundation.[5] The scores of participants are in Table 10 in terms of BLEU (Papineni et al., 2002) and F_1 (Biçici, 2011) scores. However, such automatic evaluation metrics treat the features or n-grams equivalently or group them based on their length, without knowledge about their frequency in use or significance in conveying the meaning.

Word order in a sentence does not contain the majority of information (Landauer, 2002) for vocabulary size $|V| \geq n$ where n is the average sentence length. For $n = 25$ words with $|V| = 10^5$ with equivalent representation using $n = 10$ phrases with $|V| = 10^7$ or using $n = 50$ BPE tokens with $|V| = 10^4$ or using $n = 125$ chars with $|V| = 10^2$ have differring contribution to the information of the sentence in bits from token order or choice (Table 6). If we use keyword subsequences for F_1 based evaluation, we would cover about 91% of the information in a sentence whereas if we include punctuation characters, they will contribute at most 19.3%.

Key phrase identification is important since when scores are averaged, important phrases that are missing only decrease the score by $\frac{1}{|p|N_{|p|}}$ for BLEU calculation for a phrase of length $|p|$ over $N_{|p|}$ phrases with length $|p|$. We extend our evaluation of the testsuite translations using keywords (Biçici, 2018).

We automate key phrase identification within a reference set of N sentences by selecting among N_X candidate n-grams that:

- are representative and few

[5] `http://www.preussischer-kulturbesitz.de`

$$\min \ \mathbf{X}^T(\alpha \mathbf{X}_p \cdot \mathbf{X}_l \cdot \frac{1}{-\beta \mathbf{X}_c} + \mathbf{1}_{N_X})$$

$$\text{s.t.} \quad \mathbf{X}_d(\mathbf{X} \cdot \mathbf{L}) \geq 0.5\, \mathbf{L}_N \qquad \text{min. coverage}$$

$$0 \leq \mathbf{X} \leq 1$$

$$\alpha = 1, \beta = 2$$

Variables:

$\mathbf{X} \in \mathrm{R}^{N_X}$	phrase selection vector
$\mathbf{X}_p \in \mathrm{R}^{N_X}$	phrase probability vector
$\mathbf{X}_c \in \mathrm{R}^{N_X}$	phrase count vector
$\mathbf{L} \in \mathrm{R}^{N_X}$	phrase length vector
$\mathbf{L}_N \in \mathrm{R}^{N}$	sentence length vector
$\mathbf{X}_d \in \mathrm{R}^{N \times N_X}$	phrase distribution matrix

Table 7: Optimization constraints.

system	F_1	# match	# in reference
online-B	0.869	63	82
Facebook_FAIR	0.8531	61	82
NEU	0.8286	58	82
MLLP-UPV	0.8286	58	82
online-Y	0.8286	58	82
MSRA	0.8201	57	82
RWTH_Aachen	0.8201	57	82
UCAM	0.8201	57	82
online-A	0.8029	55	82
online-G	0.7941	54	82
parfda	0.7761	52	82
PROMT_NMT	0.7761	52	82
TartuNLP-c	0.7761	52	82
uedin	0.7761	52	82
dfki-nmt	0.7481	49	82
JHU	0.6557	40	82
online-X	0.4381	23	82

Table 8: de-en testsuite F_1 scores with key phrases.

- cover significant portion of the text
- are frequent ($\mathbf{X}_c$ for counts of phrases)
- are less likely to be found ($\mathbf{X}_p$ for the probability of phrases)

and formulate the task as a linear program in Table 7. We use up to 6-grams and set minimum coverage of each sentence to 0.5. We removed some stop words from the phrases: 'of', 'the', 'and', 'of the', 'a', 'an' and replaced those parts with '.*?' and obtained regular expressions. The key phrases we obtain are listed in Table 9. The key phrases are used to evaluate using the F_1 score (Table 10). We plan to extend this work towards more objective key phrase evaluation methods.

6 Conclusion

We use `parfda` for building task specific MT systems that use less computation overall and release our engineered data for training MT systems. We also contribute a new testsuite for the German-English language pair and a new automated key phrase extraction technique for evaluation.

Acknowledgments

The research reported here received financial support from the Scientific and Technological Research Council of Turkey (TÜBİTAK).

References

Ergun Biçici. 2011. *The Regression Model of Machine Translation*. Ph.D. thesis, Koç University. Supervisor: Deniz Yuret.

Ergun Biçici. 2018. Robust parfda statistical machine translation results. In *Third Conf. on Statistical Machine Translation (WMT18)*, Brussels, Belgium.

Ergun Biçici and Deniz Yuret. 2015. Optimizing instance selection for statistical machine translation with feature decay algorithms. *IEEE/ACM Transactions On Audio, Speech, and Language Processing (TASLP)*, 23:339–350.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Christof Monz, Mathias Müller, and Matt Post. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proc. of the Fourth Conf. on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2008. *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, chapter Parallel Implementations of Word Alignment Tool.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *51st Annual Meeting of the Assoc. for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Conf. on Empirical Methods in Natural Language Processing*, pages 1352–1362.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open

source toolkit for statistical machine translation. In *45th Annual Meeting of the Assoc. for Computational Linguistics Companion Volume Demo and Poster Sessions*, pages 177–180.

Thomas K. Landauer. 2002. On the computational basis of learning and cognition: Arguments from LSA. 41:43–84.

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 *40th Annual Meeting of the Assoc. for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, Washington, USA. Association for Computational Linguistics.

A de-en Testsuite Sentences

They live in seven communities
been granted by .*? community
Southwestern Alaska has been inhabited
Hermann Parzinger
speaking groups .*? Indians immigrated
Ethnological Museum
aim .*? building up
Chugach Alaska Corporation
objects
Chugach
exhibition module in
northwest coast
ethnographic observations than by tales
goods from Chenega Island
to protect people from danger
were therefore removed unlawfully from
indications are that
graves were opened solely for
Ethnological
are two broken masks
cultural heritage
Indians immigrated
items concerned are grave goods
origin .*? history
contacts with Europe existed since
Prince William Sound
grave goods identified in
color on these ones indicates
live in seven communities
Chugach people exist today
journey is .*? impressive
consent had been granted by
virtual presentation .*? all
proposal to this effect from
President
museum at
nineteenth century for
diplomatic note in support
it was decided to return
Corporation asked .*? Ethnological Museum
indigenous peoples
Memorandum .*? Understanding with
has been inhabited for thousands
American northwest coast
now be returning them to

Table 9: Key phrases for the de-en testsuite.

de-en

model	BLEU lc				BLEU				F_1 lc				F_1			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
online-B	0.6976	0.6828	0.6042	0.5889	0.54	0.525	0.4892	0.4745	0.6419	0.6279	0.5756	0.5625	0.5264	0.514	0.4865	0.4746
Facebook_FAIR	0.7169	0.7073	0.6143	0.6037	0.5391	0.5279	0.4764	0.464	0.6311	0.6228	0.5591	0.5505	0.5039	0.4948	0.4596	0.4504
RWTH_Aachen	0.7005	0.6814	0.5889	0.573	0.5132	0.4994	0.4517	0.4385	0.6135	0.5996	0.5416	0.5295	0.4876	0.4762	0.4458	0.4349
NEU	0.7072	0.6913	0.5971	0.5805	0.5195	0.5032	0.4563	0.4396	0.6129	0.5992	0.5401	0.5274	0.4852	0.4727	0.4416	0.4292
UCAM	0.6975	0.6795	0.5943	0.5748	0.5168	0.4958	0.4513	0.4283	0.6127	0.5969	0.5389	0.5219	0.4806	0.4623	0.4346	0.4159
MSRA	0.6894	0.6746	0.5769	0.564	0.4954	0.4844	0.4296	0.4196	0.6034	0.5942	0.5278	0.5196	0.4708	0.4632	0.4263	0.4193
online-A	0.6884	0.6651	0.5822	0.5559	0.5011	0.4728	0.4338	0.4036	0.6133	0.5879	0.5348	0.5092	0.4738	0.4481	0.4265	0.4007
JHU	0.7067	0.6705	0.5923	0.5539	0.5084	0.47	0.4411	0.4025	0.6027	0.5628	0.5231	0.4848	0.4634	0.4261	0.4173	0.3816
online-Y	0.6583	0.6414	0.5413	0.525	0.4597	0.444	0.3952	0.3797	0.5838	0.5682	0.5053	0.4911	0.4469	0.4333	0.4017	0.3884
MLLP-UPV	0.6872	0.6671	0.5666	0.5428	0.4794	0.4562	0.4106	0.3884	0.5888	0.567	0.5067	0.4861	0.447	0.4275	0.4012	0.3829
dfki-nmt	0.6864	0.6503	0.5723	0.5312	0.4902	0.4442	0.4233	0.3737	0.5915	0.5463	0.5133	0.4675	0.4545	0.4082	0.4085	0.362
uedin	0.6493	0.6304	0.5309	0.5116	0.4509	0.4303	0.3862	0.3646	0.5751	0.5585	0.4945	0.4775	0.4344	0.4173	0.3888	0.3718
online-G	0.6536	0.6281	0.5269	0.5008	0.4429	0.4161	0.3809	0.3545	0.5642	0.535	0.4846	0.456	0.4284	0.4005	0.3849	0.3571
PROMT_NMT	0.6565	0.6374	0.5289	0.5074	0.4374	0.4153	0.3642	0.343	0.5529	0.5329	0.4704	0.4512	0.4094	0.3918	0.3617	0.3455
TartuNLP-c	0.6295	0.6137	0.5064	0.4911	0.4186	0.4039	0.3479	0.3339	0.5371	0.5228	0.455	0.442	0.3941	0.382	0.3472	0.3364
parfda	0.6096	0.5969	0.4642	0.4521	0.3686	0.3591	0.2994	0.2931	0.515	0.5021	0.4264	0.4159	0.3658	0.3579	0.3218	0.3153
online-X	0.6316	0.567	0.4837	0.4052	0.3828	0.2997	0.3031	0.2222	0.5149	0.4362	0.4253	0.3464	0.3601	0.2849	0.3113	0.2413

Table 10: Testsuite BLEU and F_1 results.