Exploring Model Consensus to Generate Translation Paraphrases

Zhenhao Li[1], **Marina Fomicheva**[2], **Lucia Specia**[12]
[1] Department of Computing, Imperial College London
[2] Department of Computer Science, University of Sheffield
{zhenhao.li18, l.specia}@imperial.ac.uk
{m.fomicheva}@sheffield.ac.uk

Abstract

This paper describes our submission to the 2020 Duolingo Shared Task on Simultaneous Translation And Paraphrase for Language Education (STAPLE). This task focuses on improving the ability of neural MT systems to generate diverse translations. Our submission explores various methods, including N-best translation, Monte Carlo dropout, Diverse Beam Search, Mixture of Experts, Ensembling, and Lexical Substitution. Our main submission is based on the integration of multiple translations from multiple methods using Consensus Voting. Experiments show that the proposed approach achieves a considerable degree of diversity without introducing noisy translations. Our final submission[1] achieves 0.5510 weighted F1 score on the blind test set for the English-Portuguese track.

1 Introduction

Machine Translation (MT) systems are typically used to produce a single output for a given source sentence, whereas in human translation the same source sentence can often be translated in various different ways while still preserving its meaning.

In the 2020 Duolingo Shared Task on Simultaneous Translation And Paraphrase for Language Education (STAPLE) (Mayhew et al., 2020), participating MT systems are evaluated using multiple reference translations to measure their ability to generate diverse, yet high quality translations. For that, a new dataset with multiple human translations for each source sentence is provided. These human translations were produced by language learners as part of a translation exercise on the Duolingo platform[2] where they were asked to translate sentences from the language they were learning (e.g. English) to their native language. Each translation in the dataset is assigned a weight based on the learner response frequency. Table 1 gives an example of the weighted translations in the dataset for English-Portuguese. The STAPLE dataset includes five language pairs: English to Portuguese, Hungarian, Japanese, Korean, and Vietnamese. In the shared task, we only participated in English-Portuguese (En-Pt) track.

Original	is my explanation clear?
Translation	minha explicação está clara? | 0.2673 minha explicação é clara? | 0.1616 a minha explicação está clara? | 0.1111 a minha explicação é clara? | 0.0878 minha explanação está clara? | 0.0572 está clara minha explicação? | 0.0443 minha explanação é clara? | 0.0392 ...

Table 1: An example of weighted translations in the STAPLE dataset for English-Portuguese.

In this paper, we experiment with various methods to improve the diversity of translations, while preserving their quality. We show that simply by generating N-best translations with larger beam size, we can already achieve a considerable degree of diversity. Our final submission is based on the integration of multiple translations from various methods, namely N-best translation, Monte Carlo dropout, Mixture of Experts, Ensembling, and Lexical Substitution, through a consensus voting mechanism. It achieves 0.5510 weighted F1 score on the official blind test set.

This paper is structured as follows: Section 2 describes the methods we used in our experiments. Section 3 introduces the experimental settings, including data preparation, model hyperparameters, and the evaluation procedure. Section 4 describes the results and analysis. Section 5 presents our three official submissions to STAPLE blind test set. Finally, Section 6 summarises our submission to the shared task and our contributions.

[1] https://github.com/Nickeilf/STAPLE20
[2] https://www.duolingo.com

Proceedings of the 4th Workshop on Neural Generation and Translation (WNGT 2020), pages 161–168
Online, July 10, 2020.
www.aclweb.org/anthology/D19-56%2d

2 Methods

In what follows we describe the methods used in our experiments, including N-best translation, Monte Carlo dropout, Diverse Beam Search, Mixture of Experts, Ensembling and Lexical Substitution. We combine all of these methods except the Diverse Beam Search in our official submissions through a consensus voting mechanism. Details about the submissions can be found in Section 5.

2.1 N-best

The simplest method to generate multiple translations for a given sentence is to use N-best translations with a large beam size during decoding. Larger beam size might lead to more translation options with similar meanings. We experimented with multiple sizes for N, and used the same value for N-best and beam size.

2.2 MC Dropout

Gal and Ghahramani (2016) proposed the Monte Carlo (MC) dropout method to estimate predictive NMT model uncertainty. The method consists in running several forward passes through the model (i.e., at inference time), each applying dropout before every weight layer and collecting posterior probabilities generated by the model with parameters perturbed by dropout. The mean and variance of the resulting distribution can then be used to represent model uncertainty. Instead of using this method for scoring translations, we use it as a way to generate alternative MT hypotheses for a given source sentence. Specifically, we run inference with dropout M times and collect the resulting translations. In our experiments, the dropout rate is set to 0.1 and $M = 10$.

2.3 Diverse Beam Search

Vijayakumar et al. (2016) proposed the Diverse Beam Search algorithm to improve the diversity of beam hypotheses. The algorithm proceeds by dividing the beam budget into groups and enforcing diversity between groups of beams. In our experiments we use the implementation of this algorithm in `fairseq` (Ott et al., 2019) with default parameters.

2.4 Mixture of Experts

Shen et al. (2019) introduced the Mixture of Experts (MoE) framework to capture the inherent uncertainty of the MT task where the same input sentence can have multiple correct translations. A mixture model introduces a multinomial latent variable to control generation and produce a diverse set of MT hypotheses. In our experiment we use hard mixture model with uniform prior and 5 mixture components.

2.5 Ensembling

Training an ensemble of various MT models initialized with different random seeds is a common strategy used to boost the output quality (Garmash and Monz, 2016). Unlike the typical ensembling method that combines prediction distributions from different models by averaging, we use each system in the ensemble to generate a separate set of translation hypotheses, and take the set of dictinct translations as the final output.

2.6 Lexical substitution

In the STAPLE dataset, we observed that many of the paraphrases in translations are simple variants with word substitutions in the target language. Therefore, we built a dictionary containing all lexical substitutions from the STAPLE training data. The substitutions are sorted according to two criteria: 1) number of occurrences 2) substitution probability. The substitution probability is calculated as follows:

$$P(sub) = \frac{\text{Count}(sub(w1, w2))}{\text{Count}(w1)} \quad (1)$$

The top-5 lexical substitutions from frequency-sorted and probability-sorted dictionaries are listed in Table 2. We filtered the substitution dictionary with a stopword list[3] and a threshsold (which can be either frequency count or substitution probability), to avoid generating ungrammatical translations.

Frequency		**Probability**	
substitution	count	substitution	prob
neste-nesse	5091	baixar->descarregar	1.0
irá-vai	4920	descarregar->baixar	1.0
vou-irei	4645	situa-se->fica	1.0
local-lugar	2989	achasse->encontrasse	1.0
bem-bastante	2694	localizasse->achasse	1.0

Table 2: Top-5 lexical substitutions in frequency-sorted and probability-sorted dictionaries.

[3] http://snowball.tartarus.org/algorithms/portuguese/stop.txt

2.7 Consensus voting

To integrate translations from different models, we employed a consensus voting mechanism by counting the number of systems that predicted each translation. A threshold T_{con} is set, meaning that a translation must be predicted by at least $T_{con} + 1$ systems, otherwise it is removed. Considering the lexical translation might generate rare but correct translation, we assign the lexical-substituted translations a weight W_{sub} so that they can be seen as generated by W_{sub} systems. The consensus method guarantees a high precision by removing translations that are likely to be incorrect.

3 Experiments

3.1 Data

To build the NMT model, we used parallel corpora for En-Pt from OPUS (Tiedemann, 2012) as out-of-domain data, including ParaCrawl[4], EUbookshop[5], Europarl[6], Wikipedia[7], QED[8], and Tatoeba[9]. The combination of these corpora contains 22.42 million parallel sentence pairs. The STAPLE dataset, which contains 4000 source sentences with 526,466 translations, is used as in-domain data for fine-tuning.

Since in the STAPLE dataset a source sentence have an average number of 131 reference translations, we constructed parallel data by duplicating the source sentence to match the number of translations, as shown in Figure 1.

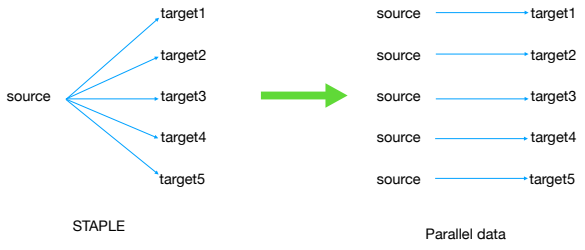

Figure 1: Constructing parallel fine-tuning data from the STAPLE dataset.

[4] http://opus.nlpl.eu/ParaCrawl-v5.php
[5] http://opus.nlpl.eu/EUbookshop-v2.php
[6] http://opus.nlpl.eu/Europarl-v8.php
[7] http://opus.nlpl.eu/Wikipedia-v1.0.php
[8] http://opus.nlpl.eu/QED-v2.0a.php
[9] http://opus.nlpl.eu/Tatoeba-v20190709.php

We also experimented with different data filtering strategies on the STAPLE dataset by only keeping the top-K translations with the highest weights (we refer to this as tune-K). Statistics regarding the corpus size after filtering are shown in Table 3.

Filtering	Source	Translations
tune-5	20,000	5.00
tune-10	40,000	10.00
tune-20	78,439	19.61
tune-all	526,466	131.62

Table 3: Size of parallel fine-tuning data after filtering the STAPLE dataset. **Source** indicates the number of source sentences and **Translations** indicates the average number of translations per source sentence

All sentences are tokenized with Moses (Koehn et al., 2007), and then processed via Byte-Pair-Encoding (BPE) (Sennrich et al., 2016). A shared vocabulary of 40,000 subwords is constructed for both English and Portuguese. The training data was then cleaned by removing sentence pairs with more than 250 subwords or with length ratio over 1.5, using the `clean-corpus-n.perl`[10] script in Moses.

3.2 Model and hyperparameters

We used the Transformer model (Vaswani et al., 2017) as our baseline model. The model is trained using `fairseq` toolkit (Ott et al., 2019) with the default hyperparameter settings using `transformer_wmt_en_de` architecture. The model was trained on 8 GPUs with a batch size of 4096 tokens on each GPU. We used mixed-precision training to accelerate the training. The model was pre-trained on OPUS data for 30 epochs and then fine-tuned on STAPLE data. We set 5 as the number of experts for training the MoE system. For ensembling, we pretrained with 3 random seeds and fine-tuned with 4 random seeds, resulting in 12 different MT systems.

3.3 Generation of Translations

When generating an integration of translations from multiple systems, we follows the procedure as described below:

1. Generate translations from N systems, resulting in N translation sets $s_1, s_2, s_3, ..., s_N$

[10] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/training/clean-corpus-n.perl

2. Apply consensus voting to the N system translations with threshold T_{con}, resulting in one translation set $s_{consensus}$

3. Apply lexical substitution to $s_{consensus}$, resulting in a separate translation set $s_{lexical}$

4. Apply consensus voting to the N system translations and the lexical substitution translation $s_1, s_2, s_3, ..., s_N, s_{lexical}$ with threshold T_{con} and weight W_{sub}, resulting in the final translation set $s_{lexical\&consensus}$.

3.4 Evaluation

The shared task provides a blind dev set (blind-dev) and a blind test set (blind-test) for evaluation. Since the number of submissions is limited, we also take a small random split from the STAPLE training set for dev (heldout-dev) and test (heldout-test) sets with 500 source sentences.

The translations are evaluated at sentence-level as a classification problem where true positives (TP) occur when the system produces one of the translations in the given set of references, false positives (FP) when a translation out of this set is produced, and false negatives (FN) when translations in this set are missed by the system. The official evaluation metric is a weighted macro F1-score averaging over all source sentences. The weighted F1 score is calculated with weighted recall and unweighted precision:

$$recall = \sum_{t \in TP} \text{weight}(t)$$

$$precision = \frac{TP}{TP + FP}$$

$$weighted\, F_1 = \frac{2 * precision * recall}{precision + recall}$$

$$weighted\, macro\, F_1 = \sum_{s \in S} \frac{weighted\, F_1(s)}{|S|}$$

4 Results

N-best We present the F1 score with respect to n-best size (from 1 to 20) in Figure 2. The models fine-tuned with different filtered data are evaluated on our heldout test set. As shown in Figure 2, the pre-trained model (tune-0) shows a poorer performance than the other fine-tuned models. The tune-1 model shows a good performance when the N-best size is small, but experiences a degradation when N-best increases. Models fine-tuned with 5, 10, and 20 reference translations show similar performances with F1 score around 0.49. However, the optimal n-best size is closely related to the number of translations used for fine-tuning, with N-best=3,10,12,18 for model tuned with 1, 5, 10 and 20 references respectively. The models fine-tuned with all translations in the STAPLE dataset show a growing trend in F1 score as n-best size increases, but the overall F1 score is still much lower than for the three fine-tuned models. We found that the upper bound for tune-all model is around 0.415 F1 score.

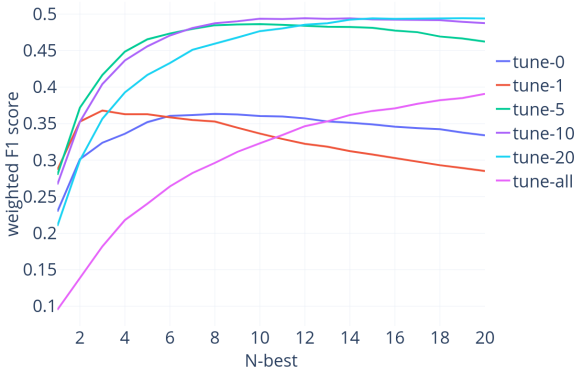

Figure 2: F1 score w.r.t N-best size for models fine-tuned with different number of reference translations.

MC dropout Table 4 shows a comparison on the heldout-test set between the N-best and N-best with MC dropout. It can be seen that the N-best12 achieves a higher recall than the N-best5, which leads to an increase of 0.038 in F1 score. When decoding with dropout, the N-best5 could match the performance of N-best12. Although noticing that MC Dropout could improve the performance for small N-best size, we found that when the N-best size gets larger the weighted F1 score does not improve further.

	Precision	**Recall**	**F1**
N-best12	0.717	0.452	0.494
N-best5	0.839	0.360	0.456
+dropout(H=10)	0.725	0.441	0.497

Table 4: A comparison between N-best and N-best with MC Dropout.

Diverse beam search When evaluating diverse beam search on the heldout-test set, we found that the model performance lags behind the N-best

baseline to a large extent, with F1 score of only 0.292. We looked into some translation examples and noticed that although diverse beam search can lead to more diversity in translations, it sometimes adds an extra full stop at the end of translations. Considering that the evaluation is conducted at sentence-level, such a minor modification can lead to a large false positive number. In the final submission, we left this method out.

Mixture of experts Regarding the MoE method, we found that different experts show inconsistent performance. As shown in Table 5, with the same N-best size, experts 2, 3, and 5 show a good performance, achieving an F1 score over 0.4. However, the other two experts, especially expert 4, exhibit poorer performance. This might be caused by insufficient training for the experts that perform poorly. In the final submission, we removed translations from experts 1 and 4 to avoid incorrect predictions.

Expert	Precision	Recall	F1
1	0.425	0.320	0.312
2	0.708	0.426	0.475
3	0.647	0.374	0.415
4	0.276	0.217	0.193
5	0.640	0.404	0.437

Table 5: An illustration of the inconsistent performance from different experts in MoE (with N-best=12).

Ensemble & Consensus In Table 6, we present our ensembling submission and consensus submission (with threshold T_{con} set to 1) on the blind-dev set. Both ensembling and consensus voting improve over the N-best by increasing the recall and reducing the precision. However, since consensus voting removed translations with fewer votes from other systems, the precision score is higher than that of ensembling while the recall is similar. This leads to a higher F1 score with the consensus submission.

	Precision	Recall	F1
N-best	0.714	0.483	0.521
+Ensemble	0.617	0.549	0.523
+Consensus($T_{con} = 1$)	0.652	0.534	0.530

Table 6: A comparison between ensembling and consensus voting.

Ensembling can be seen as a special case of consensus voting, with the threshold T_{con} being zero. Ensembling maximizes the recall by taking translations from all the systems but sacrifices the precision. Increasing the value of the threshold T_{con} would compensate for the precision loss while maintaining the gain in recall.

Lexical substitution Table 7 shows the submissions on the blind-dev set after applying lexical substitution to a consensus output combining ensembled N-best, MC dropout, and MoE systems.We first generated a set of translations with all lexical substitutions, using the translations from an N-best system. The translations with lexical substitution achieve an F1 score of 0.127, which shows potential benefits of this method. However, as shown in Table 7, simply adding the substituted translations will harm performance, and this will happen for both frequency-based sorting and probability-based sorting. This is due to the fact that the translations after substitution are likely to be ungrammatical since the substituted word does not fit in the context. To alleviate this, we added the substituted translations to the consensus pool for higher precision. This only improves over the consensus system without lexical substitution by +0.002 F1 score.

	F1
Lexical only	0.127
Consensus($T_{con} = 5$)	0.542
+lexical (freq > 1000)	0.512
+lexical (prob > 0.85)	0.532
+lexical (prob > 0.85, consensus)	0.544

Table 7: An illustraction of the benefit and harm from lexical substitution (evaluated on blind-dev set). The Consensus system combines the ensembled N-bset, MC-Dropout, and MoE systems.

In the experiment combining theses methods, we found that the N-best translations contributes the most score among all these methods. While an N-best system could achieve a weighted F1 score of nearly 0.5, other methods such as MC-Dropout, Ensembling and Consensus would only result in an extra improvement of less than 0.05 weighted F1 score. In our experiments, Diverse Beam Search and Mixture of Experts systems didn't contribute much.

5 Official submissions

Our official submissions combine translations from 12 tune-10 N-best systems (12 random seeds, finetuned with top-10 references, $N = 12$), 12 tune-20 N-best systems (12 random seeds, finetuned with top-20 references, $N = 20$), 2 MC Dropout systems ($n = 3$, $M = 50$; $n = 5$, $M = 10$), 3 experts from the MoE system, and lexical substitution (with a probability threshold of 0.7). The consensus voting threshold T_{con} is set to be 10, and the weight W_{sub} for lexical substitution is 9. Results for our three official submissions to the blind test set are shown in Table 8.

	Precision	**Recall**	**F1**
Consensus(T_{con}=10)+lexical	0.741	0.516	**0.551**
Consensus(T_{con}=10)	**0.757**	0.501	0.545
Consensus(T_{con}=1)+lexical	0.579	**0.580**	0.521

Table 8: Our three official submissions to STAPLE blind-test set.

The best submission, which achieves the best F1 score of 0.5510, applies both consensus voting and lexical substitution. As shown in the second submission, removing lexical substitution would reduce the F1 score by 0.006, although the precision is improved marginally. In the third submission, we set the consensus voting threshold T to be 1 to see the upper bound for recall. The recall increases from 0.516 to 0.580 while the precision drops significantly from 0.741 to 0.579.

Our best submission achieves the second position in the English-Portuguese track, with only 0.0006 weighted F1 score behind the winning submission. The official result on STAPLE test set is shown in Table 9.

Participant	**Weighted F1**
jbrem	0.5516
Ours	0.5510
rakchada	0.5440
aws_baseline	0.2130
fairseq_baseline	0.1357

Table 9: Official results on STAPLE test set in English-Portuguese translation (top-3 submissions and baselines).

6 Conclusions

This paper describes our submissions to the STAPLE shared task for English-Portuguese translation. We showed that simply generating N-best translations already achieves a considerable degree of diversity and quality. We experimented with various methods to improve the diversity in the MT output, including N-best translation, MC Dropout, Diverse Beam Search, Mixture of Experts, Ensembling, Consensus Voting, and Lexical Substitution. We showed the benefits and drawbacks of these methods in generating diverse, high quality translations. Our systems combining these methods further improve over the N-best translation and achieve 0.5510 weighted F1 score on STAPLE blind test set, which is only 0.0006 behind the winning submission.

References

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *international conference on machine learning*, pages 1050–1059.

Ekaterina Garmash and Christof Monz. 2016. Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418.

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, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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. Association for Computational Linguistics.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. *arXiv preprint arXiv:1902.07816*.

Jrg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

A Appendices

A.1 Checkpoiting vs tune-K

Table 10 presents the best finetuning checkpoint for models finetuned with different number of references. Models trained with more references might converge faster, and when the tuning number is larger than 40, only 1 epoch is used for finetuning.

Finetuning	Best checkpoint
tune-1	10
tune-5	10
tune-10	6
tune-20	4
tune-40	1
tune-all	1

Table 10: The best finetuning checkpoint vs the number of finetuning reference translations

A.2 Submission on blind-dev set

To provide a comprehensive understanding of the different methods, we selectively list our submissions to the blind-dev set in Table 11.

ID	System	Precision	Recall	F1	Hyperparameters
1	nbest	0.714	0.484	0.521	N=12, tune-10
2	ensemble	0.617	0.549	0.523	N=12, tune-10, 3 pretrain * 4 finetune seeds
3	nbest	0.645	0.522	0.518	N=18, tune-10
4	nbest	0.635	0.522	0.511	N=20, tune-20
5	MoE	0.368	0.527	0.385	N=12, tune-10, experts=5
6	MC Dropout	0.660	0.496	0.514	N=3, tune-10, M=50
7	MC Dropout	0.672	0.485	0.511	N=5, tune-10, M=10
8	Consensus	0.653	0.534	0.530	12*nbest systems(tune-10), $T_{con} = 1$
9	Consensus	0.641	0.541	0.529	12*nbest systems(tune-10), 2 MC Dropout systems(row 6 and 7), 5 experts, $T_{con} = 2$
10	Consensus	0.677	0.527	0.536	same as Row 9, $T_{con} = 3$
11	Lexical	0.443	0.538	0.428	same as Row 10, add lexical substitutions (frequency $>$ 4000)
12	Lexical	0.612	0.534	0.509	same as Row 11, frequency $>$ 5000
13	Consensus	0.633	0.565	0.538	24*nbest systems(tune-10, tune-20), 2 MC Dropout systems, 5 experts, $T_{con} = 4$
14	Consensus	0.652	0.558	0.542	same as Row 13, $T_{con} = 5$
15	Consensus	0.655	0.557	0.543	24*nbest systems(tune-10, tune-20), 2 MC Dropout systems, 3 experts, $T_{con} = 5$
16	Lexical	0.607	0.578	0.533	same as Row 15, add lexical substitution(probability $>$ 0.85)
17	Lexical+Consensus	0.651	0.561	0.544	same as Row 15, add lexical substitution to consensus voting ($W_{sub} = 3$)
18	Lexical+Consensus	0.667	0.553	0.546	same as Row 17, $T_{con} = 6$
19	Lexical+Consensus	0.682	0.546	0.548	same as Row 17, $T_{con} = 7$
20	Lexical+Consensus	0.697	0.540	0.550	same as Row 17, $T_{con} = 8$
21	Consensus	0.710	0.530	0.550	same as Row 15, $T_{con} = 9$
22	Consensus	**0.721**	0.526	0.551	same as Row 15, $T_{con} = 10$
23	Lexical only	0.243	0.114	0.127	
24	Lexical+Consensus	0.720	0.526	0.550	same as Row 22, add lexical substitutions to consensus voting (frequency $>$ 1000, $W_{sub} = 3$)
25	Consensus	0.564	**0.582**	0.506	36*nbest systems(tune-10, tune-20, tune-40), 2 MC Dropout systems, 3 experts, $T_{con} = 10$
26	Consensus	0.618	0.565	0.526	same as Row 25, $T_{con} = 11$
27	Lexical+Consensus	0.722	0.527	0.552	same as Row 22, add lexical substitutions to consensus voting (probability $>$ 0.99, $W_{sub} = 3$)
28	Lexical+Consensus	0.720	0.529	0.552	same as Row 27, $W_{sub} = 7$
29	Lexical+Consensus	0.718	0.531	0.553	same as Row 27, $W_{sub} = 9$
30	Lexical+Consensus	0.715	0.535	0.554	same as Row 27, (probability $>$ 0.90, $W_{sub} = 9$)
31	Lexical+Consensus	0.710	0.539	**0.555**	same as Row 27, (probability $>$ 0.70, $W_{sub} = 9$)

Table 11: Submissions on the blind-dev set.