The AFRL WMT17 Neural Machine Translation Training Task Submission

Grant Erdmann, Katherine Young, Jeremy Gwinnup
Air Force Research Laboratory
grant.erdmann@us.af.mil, katherinemccreightyoung@gmail.com, jeremy.gwinnup.1@us.af.mil

Abstract

The WMT17 Neural Machine Translation Training Task aims to test various methods of training neural machine translation systems. We describe the AFRL submission, including preprocessing and its knowledge distillation framework. Teacher systems are given factors for domain, case, and subword location. Student systems are given multiple teachers' output and a subselected set of the training data designed to match the target domain. Numerical results indicate that the student systems surpass the teachers in translation quality and that this benefit comes directly from the inclusion of the teachers' output.

1 Introduction

This paper describes our development of systems for the WMT17 Neural Machine Translation (NMT) Training Task (WMT, 2017). This task tests methods of adjusting the NMT training process, with a fixed size and format for the final English-to-Czech system. A large (approx. 50 million line) general-domain (mostly subtitles) bilingual corpus is provided as a training set. A domain is provided for each line of this corpus. News text, the application domain, composes about 0.5% of the corpus (see Table 1, column "Given"). A subword expansion to be used is explicitly provided as well. We preprocess the training data to standardize some punctuation and character encoding differences. We filter the data to remove some lines of foreign languages and little information, approximately 5% of the training data.

We follow a teacher-student (aka knowledge distillation) paradigm for this task (Ba and Caruana, 2014). We train ten replicate systems larger than the final system, based on all the training data available. These systems are aware of different factors (domain, case, subword location) for each subword, allowing them to use this information to learn finer details of translation. They also produce different outputs, based on randomness in training. We translate the entire news-domain training corpus with all replicate systems. These outputs are added to the most applicable training data as another set of references, and the final NMT systems are trained from this decimated and augmented training set.

We choose to resist making many changes to the given systems, in order to provide useful *a posteriori* comparisons. To this end, we use:

- only neuralmonkey, or branches thereof, for NMT
- the given data only
- alterations to given 4GB and 8GB configurations only.

for all intermediate systems.

2 Preprocessing of Training Data

2.1 Normalization

We use several simple steps of text normalization to produce a more standardized training set. Some lines had been doubly-tokenized, and we correct these (e.g., "`& quot ;`" becomes "`&quot;`"). Punctuation and spacing indicators are made uniform (e.g., "`Tha \x09D s`" becomes "`That ' s`"). Several characters were denoted by non-standard Unicode codepoints, and these are normalized. For instance, both of the codepoint sequences `\x03BF` and `\x043E` look similar to and were used for "o" in the English text.

2.2 Factor definition

We add several factors to the training set for the teacher systems.

We add a sentence-level factor of domain, with the following given categories: fiction, subtitles, paraweb, medical, and news.

We add a word-level factor for case (e.g., for the line "`Why did Dr . Henry Philip McCoy use BASIC ?`") with the values:

1. no case ("`.`", "`?`")
2. lowercase ("`did`", "`use`")
3. all uppercase, more than one character ("`BASIC`")
4. mixed-case: any uppercase noninitial letter ("`McCoy`")
5. capitalized, at the beginning of a line or after punctuation ("`Why`", "`Henry`")
6. capitalized abbreviation: preceding period and not last word ("`Dr`")
7. other ("`Philip`")

A word's case factor comes from the first matching condition in the above list.

Lowercasing is performed after this step, on both the source and target sides, for the teachers' training data. Information as to how the source word is mixed-case is lost for the teacher systems (e.g., "`mPa`" and "`MPa`" are equivalent).

Byte-pair-encoded (Gage, 1994) source text is given a subword-level factor for position in subword (non-subword, subword start, subword interior, subword end).

These factors are embedded into spaces with dimension equal to that of the square of the number of factors (e.g., 25 dimensions for the five factors in domain). While theoretically unimportant at convergence, this increase in dimension might encourage the training optimization to spend more effort in understanding factors.

2.3 Cleanup

Byte-pair-encoded source text is filtered for use as training data, based on two conditions. An English line must be at least 75% alphanumeric or spaces. An English line must be at most 25% "`@`" (two "`@`" being the subword continuation marker, so this is a measure of rare subwords). The filtering is based on the lowercased parallel corpus used by the teacher, but the filtered lines are also excluded from the students' cased training data.

This rough filter removes many of the non-English lines of the source text. The English lines that are filtered out appear to have little usable content. Table 1 shows how severely the different domains were filtered and their relative representation in the cleaned-up data used for training the teacher systems. The effect of filtering can be seen by comparing the "Given" and "Teacher" columns.

Approximately 5% of the initial data is filtered out by this process. Both the normalization and cleanup processes have little quantitative effect in final system quality, as seen by comparing "Given" to "Teacher" in §5. However, the processes require few resources, and we expect they have minor time and quality benefits.

3 Factored teacher systems

The teacher systems are based on the given 8 GB model configuration and trained using neuralmonkey's bandit-neuralmonkey branch[1], which seems to have good support for factors. The teacher systems are provided the lowercased general-domain training dataset, along with its domain, case, and subword location factors. Vectors for the factor embeddings are merely concatenated to the subword vectors. Convergence is declared when none of the ten teacher systems improve its validation set score for two days of training. This occurred after approximately seven passes through the training dataset. At that time, each teacher's model with the best validation score was used to translate the news-domain from the training data. The performance of the teacher systems on the validation set newstest2016 is provided in Table 2.

4 Student systems

Training data for the student systems consist of three parts. First, we include the news-domain data from the "Teacher" set. Second, we add the output of all ten teacher systems from translating this news-domain training data.

The third component is the bilingual training data from other domains, selected to be most suitable for training a news-domain system. To make this corpus, we first limit the data to lines where both languages (after the BPE process) are less than 50 words, which is a limit in the model specification. We next remove duplicate lines in the data, since some long lines similar to news data are repeated many times, and we do not want them

[1] github.com/juliakreutzer/bandit-neuralmonkey

Table 1: Breakdown of training corpora by domain, with numbers of lines in millions. "Given" is all data provided for the task. "Teacher" is cleaned data used to train teachers. "Subselector" is length-filtered and deduplicated data from which we subselect, along with news-domain data from "Teacher". "Selected" is output of subselector, along with news-domain data. "Student" is subselected data, along with news-domain data, both from "Teacher" and as translated by the ten teacher systems.

	Given		Teacher		Subselector		Selected		Student	
Domain	Lines	%	Lines	%	Lines	%	Lines	%	Lines	%
fiction	5.9	12.2	5.8	12.5	5.4	16.2	1.6	22.7	1.6	16.7
subtitles	38.6	79.5	37.1	80.1	26.6	80.1	4.6	67.3	4.6	49.5
paraweb	2.3	4.7	1.8	3.9	0.5	1.6	0.3	3.9	0.3	2.9
medical	1.5	3.1	1.4	3.0	0.4	1.3	0.2	2.4	0.2	1.8
news	0.2	0.5	0.2	0.5	0.2	0.7	0.2	3.6	0.2	2.7
teacher news	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.5	26.5
Total	48.6		46.4		33.3		6.8		9.3	

represented disproportionately. The distribution of the remaining lines is given by the "Subselector" columns (non-news rows) of Table 1.

Next, we break the corpus (approximately 33 million lines) into 44 parts (approximately 750,000 lines each) and apply our subselection algorithm (Gwinnup et al., 2016). We use 4-grams and below in the subselection coverage metric, and monolingual coverage scores are summed to get the bilingual coverage score that determines whether to include a line. We find a total of 6.6 million news-like lines from the non-news domains to use in training the student system, distributed as given by the "Selected" column of Table 1. It is noteworthy that the "fiction" domain was the most useful non-news domain, with its percentage of the training data increasing dramatically from "Given" to "Selected".

The final training data distribution for the student system is given in Table 1, in the "Student" columns. Our submitted student models had both the 4 GB and 8 GB configurations provided by the task organizers. Our 4 GB model made 8 passes through the student training set, and our 8 GB model made 4 passes. The performance on the validation set newstest2016 is provided in Table 2. Two replicates of each configuration were trained, and the systems with the highest validation set scores were submitted.

Table 2: Validation set BLEU scores of intermediate, factored 8GB teacher systems and final student systems. Scores are computed internally by neuralmonkey. Starred systems are submission systems.

System	Replicate	Score
Teacher	0	17.19
Teacher	1	16.98
Teacher	2	16.96
Teacher	3	17.07
Teacher	4	17.10
Teacher	5	17.01
Teacher	6	17.09
Teacher	7	16.82
Teacher	8	16.80
Teacher	9	17.14
Student 4GB	0	17.47
∗Student 4GB	1	17.58
∗Student 8GB	0	18.15
Student 8GB	1	18.05

5 Analysis

From Table 2 we see that the student systems perform even better on the validation set than the teacher systems. The 4 GB systems perform about half a BLEU point better, and the 8 GB systems

perform about one BLEU point better. To determine which stage in processing yielded the most benefit, 4 GB systems matching the submission criteria are built using all five of the training sets seen in Table 1. Graphical training histories are shown in Figure 1, summarized in Table 3. We see clearly that using the "Student" training data set both trains the fastest and leads to the highest-scoring systems, beating others by about a BLEU point on the validation set. The systems trained on other datasets lead to scores within about half a BLEU point of each other, with the smallest dataset (i.e., "Selected") training fastest and the largest datasets (i.e., "Given" and "Teacher") training the most slowly.

We believe that the success of the "Student" training data is caused by using training data with reachable and realistically conflicting translations. The conflicting translations provide "translator noise" and might prevent a system from overtraining or finding a strictly local optimum.

To test this theory, we build systems with exactly the same size training set as "Student", but with different composition. For these, the teacher output is replaced with:

- DupNews: ten identical copies of the news data from the given bilingual corpus (for a total of eleven copies).
- DupTeach: ten identical copies of the output from the best teacher system (i.e., from replicate Teacher-0).

As shown in Table 3, both of these adjustments begin training somewhat faster than "Selected" but are negligibly different after one week of training, at which point training is halted. This behavior supports our hypothesis that realistically conflicting translations improve the final system.

6 Discussion

We have given our method for creating the systems we submitted to the WMT17 Neural Machine Translation Training Task. After cleaning the data, we used factors to teach larger, teacher NMT systems. We trained our student submission systems using in-domain output from the teacher systems, rounded out with the most in-domain data from the general training data. The output from multiple teacher systems was used to encourage the student systems to include language ambiguity in their training. Numerical results show that we

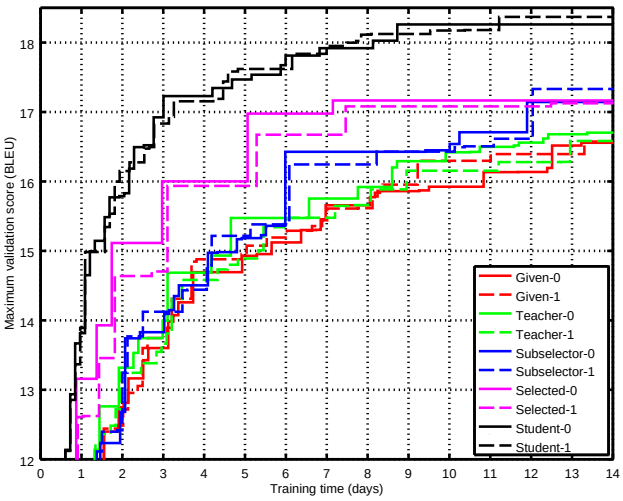

Figure 1: Scores of 4GB systems on validation set throughout training, with differing training data. Two replicates were trained per dataset. Scores are computed internally by neuralmonkey.

Table 3: Validation set BLEU scores of 4GB systems trained using different data. Scores are computed internally by neuralmonkey. "DupNews" and "DupTeach" training was halted after one week, since negligible improvement over "Selected" was found.

System-Replicate	4-day	7-day	14-day
Given-0	14.69	15.66	16.56
Given-1	14.88	15.44	16.57
Teacher-0	14.69	15.75	16.70
Teacher-1	14.58	15.48	16.58
Subselector-0	14.50	16.43	17.14
Subselector-1	14.44	16.25	17.33
Selected-0	16.00	16.98	17.17
Selected-1	15.94	16.67	17.13
Student-0	17.23	17.92	18.26
Student-1	17.15	17.95	18.37
DupNews-0	16.83	16.83	
DupNews-1	16.68	16.68	
DupTeach-0	16.66	16.99	
DupTeach-1	16.66	16.94	

can distill knowledge of multiple well-informed teacher systems into smaller student systems.

References

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? *Advances in Neural Information Processing Systems* pages 2654–2662.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal* 12:23–38.

Jeremy Gwinnup, Tim Anderson, Grant Erdmann, Katherine Young, Michaeel Kazi, Elizabeth Salesky, and Brian Thompson. 2016. The AFRL-MITLL WMT16 news-translation task systems. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 296–302. http://www.aclweb.org/anthology/W16-2313.

WMT. 2017. Findings of the 2017 Conference on Statistical Machine Translation. In *Proc. of the Second Conference on Statistical Machine Translation (WMT '17)*. Copenhagen, Denmark.

Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government. Cleared for public release on 22 May 2017. Originator reference number RH-17-117141. Case number 88ABW-2017-2505.