Automatic Translation for Multiple NLP tasks: a Multi-task Approach to Machine-oriented NMT Adaptation

Amirhossein Tebbifakhr
FBK, Trento, Italy
University of Trento, Italy
atebbifakhr@fbk.eu

Matteo Negri
FBK, Trento, Italy
negri@fbk.eu

Marco Turchi
FBK, Trento, Italy
turchi@fbk.eu

Abstract

Although machine translation (MT) traditionally pursues "human-oriented" objectives, humans are not the only possible consumers of MT output. For instance, when automatic translations are used to feed downstream Natural Language Processing (NLP) components in cross-lingual settings, the translated texts should ideally pursue "machine-oriented" objectives that maximize the performance of these components. Tebbifakhr et al. (2019) recently proposed a reinforcement learning approach to adapt a generic neural MT (NMT) system by exploiting the reward from a downstream sentiment classifier. But what if the downstream NLP tasks to serve are more than one? How to avoid the costs of adapting and maintaining one dedicated NMT system for each task? We address this problem by proposing a multi-task approach to machine-oriented NMT adaptation, which is capable to serve multiple downstream tasks with a single system. Through experiments with Spanish and Italian data covering three different tasks, we show that our approach can outperform a generic NMT system, and compete with single-task models in most of the settings.

1 Introduction

Neural Machine Translation (NMT) systems are typically developed considering humans as the end-users, and are hence optimized pursuing human-oriented requirements about the output quality. To meet these requirements, supervised NMT models are trained to maximize the probability of the given parallel corpora (Bahdanau et al., 2015; Sutskever et al., 2014), which embed the adequacy and fluency criteria essential for the human comprehension of a translated sentence. In another line of research, these objectives are directly addressed in Reinforcement Learning (Ranzato et al., 2016; Shen et al., 2016) and Bandit Learning (Kreutzer et al., 2017; Nguyen et al., 2017), where model optimization is driven by the human feedback obtained for each translation hypothesis.

However, humans are not the only possible consumers of MT output. In a variety of application scenarios, MT can in fact act as a pre-processor to perform other natural language processing (NLP) tasks. For instance, this is the case of text classification tasks for which, in low-resource conditions, the paucity of training data provides a strong motivation for exploiting translation-based solutions. In tasks like sentiment classification, hate speech detection or document classification (the three application scenarios addressed in this paper) a translation-based approach would allow: *i)* translating the input text data from an under-resourced language into a resource-rich target language for which high-performance NLP components are available, *ii)* run a classifier on the translated text and, finally, *iii)* project the results back to the original language.

This approach represents a straightforward solution in low/medium-resource[1] language settings

[1] Jain et al. (2019) consider as "medium-resource" languages those for which, although annotated training corpora do not exist, off-the-shelf (MT) systems like Google Translate are available.

where reliable NLP components for specific tasks are not available, and represents a strong baseline in a variety of multilingual and cross-lingual NLP tasks (Conneau et al., 2018). However, the NMT systems normally used are still optimized by pursuing human-oriented adequacy and fluency objectives, which are not necessarily the optimal ones for this pipelined solution. These models can indeed produce translations in which some properties of the input text are altered or even lost. For instance, as shown in (Mohammad et al., 2016), this happens in sentiment classification, where automatic translations can fail to properly project core traits of the input text into the target language. When this happens, the downstream linguistic processor will likely produce results of lower quality.

In light of these considerations, Tebbifakhr et al. (2019) argued that when the role of NMT is to feed a downstream NLP component instead of a human, translating into fluent and adequate sentences is not necessarily the main priority. Rather, if the goal is producing translations that are "easy to process" by the downstream component, other optimization strategies might be more effective, even if they result in low-quality output from the point of view of human comprehension. Back to the sentiment classification example: before meaning and style, a "machine-oriented" translation should prioritize the optimal projection of the sentiment traits of the input text, which are the key clues from the automatic sentiment classification standpoint.

To pursue machine-oriented translation objectives, Tebbifakhr et al. (2019) proposed Machine-Oriented Reinforce (MO-Reinforce), a method based on Reinforce (Williams, 1992; Ranzato et al., 2016). While in Reinforce the objective is to maximize the reward given by humans to NMT systems' output, in MO-Reinforce the human feedback is replaced by the reward coming from a downstream NLP system. Focusing on sentiment classification, where the classifier's output is a probability distribution over the classes for each input text, they define the reward as the probability of predicting the correct class. Evaluation results computed on Twitter data show that a downstream English sentiment classifier performs significantly better when it is fed with machine-oriented translations rather than the human-oriented ones produced by a general-purpose NMT system.

Despite its potential usefulness, MO-Reinforce has a limitation that might reduce its general applicability: it requires one NMT model for each downstream task. This represents a possible bottleneck in real industry scenarios, where training and maintaining multiple task-oriented NMT systems (one for each possible downstream task) would be costly and time-consuming, if not unfeasible. To overcome this limitation, in this paper we explore the possibility to simultaneously address multiple downstream tasks with a single NMT system. In this direction, we propose a multi-task learning approach that has two main potential strengths. One is the higher flexibility for industrial deployment due to its architectural simplicity. The other is the possibility to exploit knowledge transfer across similar tasks (Zhang and Yang, 2017), eventually improving the results achieved by the single-task MO-Reinforce approach.

We test the viability of our multi-task approach on two source languages (Spanish and Italian[2]) for which data covering different tasks (sentiment classification, hate speech detection and document classification) have to be translated into English and then processed by dedicated NLP components. Our results show that translating with the proposed multi-task extension yields significant gains in classification performance with respect to both *i)* a generic NMT system and *ii)* the original single-task MO-Reinforce by Tebbifakhr et al. (2019).

Besides exploring for the first time a multi-task approach to "machine-oriented" NMT, this paper provides two technical contributions that explain the reported performance gains, namely: *i)* a reward normalization strategy to weigh the importance of each sample in the course of training, and *ii)* the application of dropout while sampling the translation candidates, which makes the model more reactive and avoids local optima. On the experimental side, another contribution of this work is the first evaluation on multi-class classification data (i.e., those used for the document classification task), a more challenging scenario compared to the binary task considered by Tebbifakhr et al. (2019).

[2]Although one of the motivations for machine-oriented translation is to support NLP in under-resourced settings, the chosen source languages do not fall in this category. The choice is motivated by the fact that they provide us with all the necessary infrastructure (e.g. test data) to perform a sound comparative evaluation. Here, indeed, we focus on testing the general applicability of our approach, while its evaluation in real under-resourced settings (conditioned to the availability of benchmarks for multiple tasks) is left for future work.

2 Background

2.1 Human-oriented NMT

Formally, in MT, the probability of generating the translation $\mathbf{y}$ with length of N given a source sentence $\mathbf{x}$ is computed as follows:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{N} p_\theta(\mathbf{y}_i|\mathbf{y}_{<i}, \mathbf{x}) \quad (1)$$

where p_θ is a conditional probability defined by sequence-to-sequence NMT models (Bahdanau et al., 2015; Sutskever et al., 2014; Vaswani et al., 2017). In these models, an encoder first encodes the source sentence and then, at each time step, a decoder outputs the probability distribution over the vocabulary conditioned on the encoded source sentence and the translation prefix $\mathbf{y}_{<i}$. In supervised NMT, the parameters of the model θ are trained by maximizing the log-likelihood of the given parallel corpus $\{\mathbf{x}^s, \mathbf{y}^s\}_{s=1}^{S}$:

$$\begin{aligned}\mathcal{L} &= \sum_{s=1}^{S} \log P(\mathbf{y}^s|\mathbf{x}^s) \\ &= \sum_{s=1}^{S}\sum_{i=1}^{N^s} \log p_\theta(\mathbf{y}_i^s|\mathbf{y}_{<i}^s, \mathbf{x})\end{aligned} \quad (2)$$

By maximizing this objective, the model indirectly pursues the human-oriented objectives of adequacy and fluency embedded in the training parallel corpora.

In addition to normal NMT training, these objectives can be directly addressed using reinforcement learning methods such as Reinforce (Ranzato et al., 2016). This method maximizes the expected reward from the end-user:

$$\begin{aligned}\mathcal{L} &= \sum_{s=1}^{S} E_{\hat{\mathbf{y}}\sim P(.|\mathbf{x}^s)}\Delta(\hat{\mathbf{y}}) \\ &= \sum_{s=1}^{S}\sum_{\hat{\mathbf{y}}\in\mathbf{Y}} P(\hat{\mathbf{y}}|\mathbf{x}^s)\Delta(\hat{\mathbf{y}})\end{aligned} \quad (3)$$

where $\Delta(\hat{\mathbf{y}})$ is the reward of the sampled translation candidate $\hat{\mathbf{y}}$, and $\mathbf{Y}$ is the set of all the possible translation candidates. Since the size of this set $\mathbf{Y}$ is exponentially large, Equation 3 is estimated by sampling one translation candidate out of this set using multinomial sampling or beam search:

$$\hat{\mathcal{L}} = \sum_{s=1}^{S} P(\hat{\mathbf{y}}|\mathbf{x}^s)\Delta(\hat{\mathbf{y}}), \hat{\mathbf{y}} \sim P(.|\mathbf{x}^s) \quad (4)$$

Since collecting human rewards is costly, the process can be simulated by comparing the sampled translation candidates with the corresponding reference translations using automatic evaluation metrics like BLUE (Papineni et al., 2002).

The two learning strategies (supervised and reinforcement) have two main commonalities: *i)* the learning objectives are human-oriented, and *ii)* they both need parallel data, respectively for maximizing the probability of the translation pair in supervised learning and for simulating the human reward in reinforcement learning.

2.2 Machine-oriented NMT

To pursue machine-oriented objectives and to bypass the need for parallel corpora, in the MO-Reinforce algorithm proposed by (Tebbifakhr et al., 2019), the human reward is replaced by the reward from a downstream classifier (in that case, a polarity detector predicting the positive/negative sentiment of a translated sentence). This reward is defined as the probability of labeling the translated text with the correct class and it can be easily computed since the output of the downstream classifier is a probability distribution over the possible classes. Therefore, given a small amount of labeled data in the source language[3] $\{\mathbf{x}^s, \mathbf{l}^s\}_{s=1}^{S}$, in which $\mathbf{l}$ is the label of the corresponding source text $\mathbf{x}$, Equation 4 can be redefined as follows:

$$\hat{\mathcal{L}} = \sum_{s=1}^{S} P(\hat{\mathbf{y}}|\mathbf{x}^s)\Delta(\hat{\mathbf{y}}, \mathbf{l}^s), \hat{\mathbf{y}} \sim P(.|\mathbf{x}^s) \quad (5)$$

where $\Delta(\hat{\mathbf{y}}, \mathbf{l}^s)$ is the probability that the downstream classifier assigns $\mathbf{l}^s$ to a sampled candidate.

In order to increase the contribution of the reward and to sample "useful" translation candidates, the proposed sampling strategy randomly extracts K candidates and eventually chooses the one with the highest reward to update the model. This strategy results in the selection of candidates that influence the initial model towards translations that maximize the performance of the downstream processor. For instance, in the sentiment classification scenario, these are NMT outputs that preserve, or even emphasize, relevant aspects like the proper handling of sentiment-bearing terms. Although they are poor in terms of the human-oriented notion of quality (as shown by BLEU scores close

[3] In (Tebbifakhr et al., 2019), MO-Reinforce is shown to result in better classification performance than the original Reinforce (Ranzato et al., 2016) with few hundred labeled instances (~ 500).

to zero when compared against human references), their high sentiment polarization considerably simplifies the polarity labelling task.

Despite the significant gains compared to the classification performance achieved by translating with a generic NMT system, a limitation of MO-Reinforce lies in its applicability to one task at a time. Serving multiple tasks would only be possible by training multiple NMT models (one for each possible downstream classifier), which is a sub-optimal solution for the actual deployment of the approach in real industrial settings. To overcome this issue, in the next section we propose an extension aimed at simultaneously serving multiple classifiers with a single NMT system. Later, in the experimental part of the paper (sections 4 and 5), we will evaluate it in a multi-task scenario involving both binary and multi-class tasks.

3 Multi-task Machine-oriented NMT

Our multi-task extensions of MO-Reinforce include: *i)* prepending task-specific tokens to the input for managing multiple domains and computing normalized rewards to avoid under/over-fitting (Section 3.1), and *ii)* adding randomness to the sampling process to push for higher exploration of the probability space (Section 3.2).

3.1 Normalized Reward

To serve multiple downstream classifiers with a single NMT system, the model has to be trained on a mixture of the labeled datasets available for the different tasks. To define the target task, we prepend a task-specific token to each input sample within the corresponding dataset. In this way, the NMT model is informed about the target downstream application for which the input text has to be translated. This idea is drawn from multilingual NMT, in which an effective solution is to prepend to the input sentences a token defining the desired target language (Johnson et al., 2017).

To avoid under/over-fitting when training the NMT model on mixed datasets that can have different sizes, we need to schedule the sampling from these datasets. In multilingual NMT, two fixed sampling schedules have been proposed, namely: *i)* proportionally with respect to the dataset size (Luong et al., 2015), or *ii)* uniformly from each dataset (Dong et al., 2015). However, these fixed scheduling approaches are not optimal solutions. The first one gives higher importance to tasks with larger datasets, so that those with less training material might remain under-fitted. The second one gives equal importance to all the tasks, which implies that larger datasets for some tasks will not be fully exploited, reducing systems' performance on those tasks.

To overcome these limitations, adaptive scheduling strategies can be adopted to update the importance of each task in the course of training. The idea is that, when the performance of the model is low on one task, higher importance is given to that task. This can be done by keeping the schedule fixed and scaling the gradients (Chen et al., 2017), or directly by changing the sampling weights (Jean et al., 2019). In the first approach by Chen et al. (2017), the adaptation is done based on the magnitude of the gradients. However, the computed gradients loosely correlate with the performance of the model and do not directly measure model's performance for the corresponding task. The second one (Jean et al., 2019), requires knowing the performance of the single-task models for each task on the development set before starting the training. Then, after each epoch, the results of the multi-task model on the same development set are compared with those achieved by the single-task models, and the weights get updated accordingly. As a direct indicator, models' performance on the development set represents a more reliable alternative compared to exploiting the indirect information provided by gradients' magnitude. However, it is more computationally intensive and it assumes knowing in advance the performance of the single-task models, which is not always available.

We hence opt for the idea of scaling the gradients while keeping the schedule fixed and uniform across tasks. We make the adaptation based on the reward from the downstream task, which reflects the performance of the model for the corresponding input sample. Equation 6 shows the stochastic gradient of the MO-Reinforce objective function.

$$\nabla \hat{\mathcal{L}} = \sum_{s=1}^{S} \Delta(\hat{\mathbf{y}}, \mathbf{l}^s) \nabla \log P(\hat{\mathbf{y}}|\mathbf{x}^s) \qquad (6)$$

In this formulation, since the magnitude of the reward scales the computed gradient for each sample, those samples with higher rewards will also have higher influence on the model adaptation process. This can have a negative impact when the samples come from challenging tasks or even from

challenging classes within a specific task. These samples, in fact, will likely get lower reward leaving the corresponding tasks/classes under-fitted.

To avoid this problem and to boost performance when dealing with challenging samples, we propose a reward normalization step, which extends MO-Reinforce with the possibility to weight the importance of each sample during training. The idea is that the average reward for the K translation candidates sampled by MO-Reinforce in order to chose the most useful one (see Section 2.2) can be considered as an indicator of the level of difficulty of each task. Therefore, to normalize the reward, this average value can be subtracted from the original reward as follows:

$$\hat{\Delta}(\hat{\mathbf{y}}, \mathbf{l}) = \Delta(\hat{\mathbf{y}}, \mathbf{l}) - \frac{\sum_{k=1}^{K} \Delta(\hat{\mathbf{y}}_k, \mathbf{l})}{K} + \alpha \quad (7)$$

where K is the number of sampled translation candidates. We add a constant value α to prevent zero reward for the cases in which all the rewards have the same value. This normalization reduces more the reward of easy samples, whose average is high, and subsequently results in giving more importance to challenging samples with low reward.

3.2 Noisy Sampling

Two sampling strategies are used for sampling the translation candidates in reinforcement learning. The first one is *beam search* (Sutskever et al., 2014). It is a heuristic search, which maintains a pool of highest probability translation prefixes with size B. At each step, the prefixes in the pool are expanded by B highest probability words from the model's distribution output. Then, the resulting $B \times B$ hypotheses are pruned by keeping B-highest probability prefixes. The search continues until all the prefixes in the pool reach the *EOS* token. The second one is *multinomial sampling* (Ranzato et al., 2016) where, at each time step, a word is generated by sampling from the model's distribution output. The generation is terminated when the *EOS* token is generated.

For a given application, the choice between the two sampling strategies depends on the known trade-off between exploration and exploitation in reinforcement learning. Indeed, while beam search exploits more the model's knowledge, multinomial sampling is more oriented to exploring the probability search space. In light of this difference, in MO-Reinforce the sampling is done using multinomial sampling, which achieves better results in NMT (Wu et al., 2018). This is needed, since the parameters of the model are initialized by a generic NMT system, which is trained on parallel data pursuing human-oriented objectives. Pushing for the exploration of the probability space instead of exploiting the original model's knowledge will promote the generation of more diverse candidates and eventually increase the chance to influence system's behaviour towards our machine-oriented objectives.

Although for these reasons multinomial sampling represents a better choice compared to beam search, in MO-Reinforce the exploration of the probability space does not always result in a boost of candidates' diversity. For instance, the higher randomness in generating the translation candidates might not suffice when the model's probability distribution is very peaked (i.e. when, at a given time step, the number of plausible options for the next word is very small). In this case, multinomial sampling will likely generate the same candidate at different iterations on the data. If its reward is the highest one among the K samples, this candidate will be chosen and the model will be updated to increase the candidate's probability. The result will be an even more peaked distribution that, in turn, will increase the chance of making the model stuck in a local optimum by repeatedly generating the same candidate.

To avoid these local optima and make MO-Reinforce more reactive to handle multi-task data, our last extension aims to perturb the model's probability distribution. We do this by enabling *dropout* (Srivastava et al., 2014) while generating the candidates, which is usually disabled while generating the translation outputs. Dropout adds permutation in sampling, which helps the model to generate different translation candidates at different passes over the data even in the case of highly peaked probability distributions.

4 Experiments

Our multi-task extension of MO-Reinforce is evaluated on two source languages: Spanish and Italian. For Spanish, we consider the downstream tasks of document classification and hate speech detection. For Italian, we select document classification and sentiment analysis. The evaluation is done by feeding dedicated English classifiers (one for each downstream task) with translations produced by different NMT models, namely: *i)*

Spanish Tasks						
	MLDoc				Hate Speech	
	CCAT	ECAT	GCAT	MCAT	Non-Hateful	Hateful
Train	100	100	100	100	400	400
Developement	314	201	208	277	500	500
Test	1246	731	794	1229	278	222
Italian Tasks						
	MLDoc				Sentiment	
	CCAT	ECAT	GCAT	MCAT	Negative	Positive
Train	100	100	100	100	2289	1450
Developement	239	248	238	275	254	161
Test	963	1066	976	995	733	316

Table 1: Statistics of datasets used for the Spanish and Italian tasks.

	Europarl	JRC	Wikipedia	ECB	TED	KDE	News11	News	Total
Es-En	2M	0.8M	1.8M	0.1M	0.2M	0.2M	0.3M	0.2M	5.6M
It-En	2M	0.8M	1M	0.2M	0.2M	0.3M	0.04M	0.02M	4.56M

Table 2: Statistics of the parallel corpora used for training the generic NMT systems

a general-purpose NMT system, *ii)* the original single-task MO-Reinforce, and *iii)* different variants of our multi-task extension. The goal is to maximize the classification performance on each downstream task. As another term of comparison for the three translation-based solutions, we consider the results obtained by directly processing the input sentences with task-specific Spanish and Italian classifiers trained on the same small datasets used to adapt the general-purpose NMT system.

In line with (Tebbifakhr et al., 2019), the multi-task approach is expected to outperform the generic (human-oriented) NMT system, as well as the task/language-specific classifiers trained on few data points. Ideally, thanks to the solutions proposed in Section 3, it should also compete with the single-task (machine-oriented) models. This would indicate the viability of a single-model approach to simultaneously address multiple tasks.

In the following, we describe the task-specific data used for model adaptation and evaluation, as well as the parallel corpora used for training the generic NMT system. Their statistics are respectively reported in Tables 1 and 2.

Document Classification. For this multi-class labelling task, we use the MLDoc corpora (Schwenk and Li, 2018), which cover 8 languages, including English, Spanish and Italian. They comprise news stories labeled with 4 different categories: CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Government/Social), and MCAT (Markets). For each language, the training, development and test sets respectively contain 10K, 1K, and 4K documents uniformly distributed into the 4 classes. Following (Bell, 1991), for training and evaluation we only consider the first sentence of each document, which usually provides enough information about the general content of the document. We use the whole English training set to build our downstream classifiers. To simulate an under-resourced setting, we randomly sample 100 documents for each class from the Spanish and Italian training sets. We use these samples to adapt the generic NMT system for the downstream task, while for development and test we use the whole sets.

Hate Speech Detection. For this binary task, we use the English and Spanish datasets published for the multilingual hate speech detection shared task at SemEval 2019 (Basile et al., 2019). We train the downstream classifier on the whole English training set, including 3,783 hateful and 5,217 non-hateful Twitter messages. We randomly sample 400 tweets for each class from the Spanish training set in order to simulate the under-resourced setting. Since the test set is not publicly available, we use the development set as final evaluation benchmark, and we sample 500 tweets for each class from the rest of the training set as the development set.

Sentiment Classification. For this binary task, we use a collection of annotated tweets released for the Italian sentiment analysis task at Evalita 2016 (Barbieri et al., 2016). After filtering out the subjective tweets and the ones with mixed polarity,

Models	Spanish-English		Italian-English	
	MLDoc	Hate Speech	MLDoc	Sentiment
Generic	82.58	54.49	75.43	51.89
Source	84.86	75.29	73.24	64.06
Single-task MO-Reinforce	88.36	64.24	76.86	**70.27**
Multi-task MO-Reinforce (proportional sampling)	86.18	62.93	10.83	70.11
Multi-task MO-Reinforce (uniform sampling)	86.45	55.07	68.26	68.01
Multi-task MO-Reinforce (normalization)	86.98	66.52	75.11	66.70
Multi-task MO-Reinforce (dropout)	87.73	**77.56**	80.31	68.98
Multi-task MO-Reinforce (dropout & normalization)	**90.13**	77.08	**80.90**	66.73

Table 3: Classification results (F1) obtained by: *i)* translating with the *Generic* NMT system, *ii)* directly processing the untranslated data (*Source*), *iii)* translating with separate *Single-task MO-Rinforce* models, *iv)* one *Multi-task MO-Reinforce* model with different sampling strategies, *v)* one *Multi-task MO-Reinforce* model with reward normalization and noisy sampling.

we train the downstream system using a balanced set of 1.6M negative and positive tweets (Go et al., 2009).

Generic NMT systems We train the generic NMT system using the parallel corpora reported in Table 2. After filtering out long and imbalanced pairs, we encode the corpora using 32K byte-pair codes (Sennrich et al., 2016). Our NMT model uses Transformer with parameters set as in the original paper (Vaswani et al., 2017). In all the settings, we start the training by initializing the NMT model with the trained generic NMT systems. Then, we continue the training for 50 epochs and choose the best performing checkpoint based on the average F1 score measured on the development set of each task. We set K (i.e. the number of sampled translation candidates at each time step) to 5, and used the development set to evaluate different values of α (i.e. the constant value added to prevent zero rewards – see Section 3.1). The best-performing value of 0.1 was then used in all the experiments. For developing the classifiers (both the downstream English ones and the language-specific ones used as baseline), we fine-tune the multilingual BERT (Devlin et al., 2019).

5 Results and Discussion

Our experimental results are shown in Table 3, which reports the classification performance (F1) obtained on each downstream task by:

- Feeding the English classifiers with translations from different NMT models (i.e. *Generic*, *Single-task MO-Reinforce* and different variants of *Multi-task MO-reinforce*);
- Running language-specific classifiers on the original untranslated data (*Source*).

The F1 scores obtained by the *Generic* NMT systems in document classification (MLDoc) show that the simplest translation-based approach produces competitive results compared to those achieved by language-specific classifiers trained on small in-domain data. The situation is different for tasks whose data differ significantly from those used to train the general-purpose system. On the user-generated content used for hate speech detection and sentiment classification (i.e. Twitter data), the *Generic* results are indeed poor. This shows that NMT models trained by only pursuing human-oriented criteria might not fit to target downstream tasks, for which machine-oriented adaptation becomes necessary.

Machine-oriented adaptation with single-task *MO-Reinforce* yields the expected benefits, with improvements (+3.25 F1 points for document classification, +18.38 for sentiment classification in Italian) that allow to outperform the language-specific (*Source*) classifiers in three tasks out of four. These gains confirm and validate on multiple tasks (including multi-class classification) the findings of Tebbifakhr et al. (2019), showing that *MO-Reinforce* can leverage the feedback from external linguistic processors to adapt the NMT model towards translations that maximize the performance in downstream applications.

The middle part of Table 3 shows the first results obtained by our multi-task adaptation of *MO-Reinforce*. This is done by prepending the task-specific tokens and comparing the two fixed sampling schedules (proportional to datasets' size and uniform). As expected (see Section 3.1), when sampling proportionally, the task with less training data (MLDoc) starves in training and remains under-fitted. This is particularly evident for Italian, where the document classification dataset is ten

Models	Spanish-English		Italian-English	
	MLDoc	Hate Speech	MLDoc	Sentiment
Single-Task MO-Reinforce	88.36	64.24	76.86	70.27
Single-Task MO-Reinforce (dropout)	**89.91**	35.73	**81.87**	65.67
Single-Task MO-Reinforce (dropout & normalization)	88.55	**78.33**	81.22	**70.97**

Table 4: Classification results (F1) obtained by translating with the original single-task MO-Reinforce and two variants of multi-task MO-Reinforce (with noisy sampling – dropout – alone and combined with reward normalization).

times smaller than the sentiment analysis one, and performance is particularly low (10.83). On Spanish, where the hate-speech dataset is only twice as big as the document classification one, the problem exists but it is less evident. Although uniform sampling helps the task with less training data (MLDoc) to achieve better performance, it harms those with more data, which remain under-fitted (lower performance than proportional sampling). Analysing the performance of the multitask and single task variants of *MO-Reinforce*, we notice that, although the former still outperforms the *Generic* NMT system in three tasks out of 4, its results are worse compared to the single-task *MO-Reinforce*. For the task with the most unbalanced data (MLDoc Italian), uniform sampling helps to increase the performance, but it is not sufficient to reach the scores achieved by *Generic* NMT. On hate speech data, the results of the language-specific classifiers (*Source*) are still the highest ones. The results reported so far would not allow a user to replace the single task systems with the multitask one.

The bottom part of Table 3 reports the classification results obtained by *MO-Reinforce* with reward normalization and noisy sampling (both separately and together). As it can be seen, reward normalization is beneficial for both the Spanish tasks, with a larger performance gain on hate speech with respect to both the sampling strategies (+3.59 and +11.45 F1 points). For Italian, reward normalization helps in the MLDoc task (+6.85 over the best sampling strategy), but it results in a performance drop in sentiment classification (-1.31). In general, reward normalization shows to be useful for tasks that tend to remain under-fitted with proportional or uniform sampling. Concerning the sentiment analysis task, our intuition is that, in presence of a large quantity of task-specific data in the target language, both the English classifier and the computed rewards are reliable enough. Scaling the rewards with their average value (see Eq. 7) reduces the learning capability of the NMT system, resulting in an under-fitted model. Although adding reward normalization reduces the gap in performance with respect to the single-task *MO-Reinforce* and the Source classifiers, it is not yet sufficient to replace them.

The results are significantly better with the noisy sampling approach discussed in Section 3.2. In both the languages and in all the tasks, the reported F1 scores approach those obtained by the single-task variant of *MO-Reinforce* (which in two cases is even outperformed) and always improve over the language-specific Source classifiers. This confirms that enabling dropout while generating the translation candidates avoids the model to get stuck in local optima, and promotes diversity in producing candidates that eventually receive higher rewards.

Combined, the two contributions of this paper (reward normalization and noisy sampling) yield mixed outcomes. For Spanish, we observe a further improvement compared to noisy sampling in document classification (+2.40), which comes at the cost of a small drop in hate speech detection (-0.48). Also for Italian there is an improvement over noisy sampling alone in document classification (+0.59), but a larger drop in sentiment classification performance (-2.25). However, it's worth remarking that: *i)* the size of the Italian sentiment analysis dataset is almost 10 times larger than the size of the document classification dataset, and *ii)* the data used to train the English classifiers are even more unbalanced. Being able to harmonize the results of the two task hence becomes quite difficult. Nevertheless, combining reward normalization and noisy sampling has a general positive effect, which allows the multi-task *MO-Reinforce* system to approach and, in some tasks, even to outperform the single task models.

In our final analysis, we investigate the effect of introducing dropout and reward normalization when *MO-Reinforce* is used in the single-task scenario. As shown in Table 4, enabling dropout improves the document classification results in both the languages. The reported scores show that the

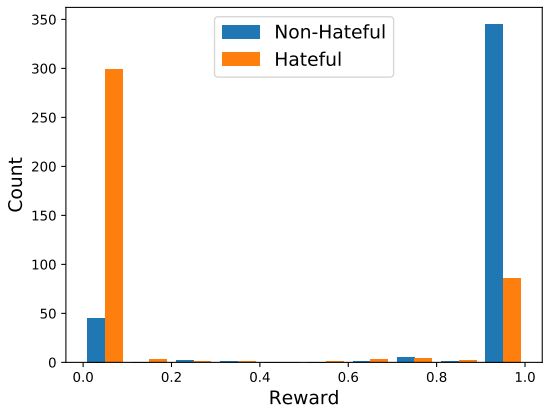

Figure 1: Rewards distribution for the hate speech detection training set translated with the *Generic* NMT system.

added noise introduced by dropout helps the model to explore more the probability space and avoid local optima, even when dealing with a single task. However, for hate speech detection in Spanish and sentiment analysis in Italian, this exploration of the probability space results in lower performance compared to the original *MO-Reinforce*. To understand the reasons of this drop, Figure 1 shows the distribution of the rewards obtained in hate speech detection when translating the training set with the generic NMT system. This distribution shows that the downstream classifier is very biased toward the non-hateful class (right side of Figure 1), with most of the hateful samples obtaining zero reward (left side). While the model is exploring the probability space, this extreme imbalance in the rewards does not allow the hateful samples to get a non-zero reward, and this drastically scales down their gradients preventing the NMT system to actually learn from these samples. Eventually, this results in a "catastrophic forgetting", where the NMT system learns only from one class and totally forgets the other. Whatever it will receive in input, this system will generate a translation with no hate nuances, which will be classified as non-hateful by the downstream classifier. The very low F1 (35.73) is the result of this process.

Adding reward normalization minimizes the "catastrophic forgetting" effect by keeping the magnitude of the rewards balanced across the classes. In terms of performance, hate speech detection and sentiment analysis benefit of it by achieving higher results compared to the original *MO-Reinforce* (respectively, +14.09 and +0.77). On both the languages, the document classification results slightly drop compared with *MO-Reinforce* with dropout, but they still outperform those achieved by translating with the original approach by (Tebbifakhr et al., 2019).

Looking at the output of the system, we noticed that the translations are shorter and are not adequate compared to the output of the *Generic* system. For instance, in document classification, the samples belonging to the Corporate class are usually translated to "*The company.*", or the positive samples in sentiment analysis are translated to "*I'm very happy.*", which are easier to be classified by the downstream classifiers.

6 Conclusion

We proposed an extension of the MO-Reinforce algorithm, targeting "machine-oriented" NMT adaptation in a multi-task scenario. In this scenario, different NLP components are fed with translations produced by a single NMT system, which is adapted to generate output that is "easy to process" by the downstream processing tools. To close the performance gap between the single and the multi-task variants of MO-Reinforce, we enhanced the latter with reward normalization and noisy sampling strategies. Our experiments show that, with these two features, the multi-task MO-Reinforce approach achieves significant gains in performance that make it competitive with the single-task solution (though, having one single model to build and maintain, at considerably lower deployment costs). Furthermore, we show that reward normalization and noisy sampling can also help in the single-task setting, where our approach outperforms the original MO-Reinforce in four tasks.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proc. of ICLR 2015, San Diego, CA, USA, May.

Barbieri, F, V Basile, D Croce, M Nissim, N Novielli, and V Patti. 2016. Overview of the evalita 2016 sentiment polarity classification task. In Proc. of EVALITA 2016, Naples, Italy, December.

Basile, Valerio, Cristina Bosco, Elisabetta Fersini, Debora Nozza, et al. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In Proc. of SEMEVAL 2019, pages 54–63, Minneapolis, Minnesota, USA, June.

Bell, A. 1991. The Language of News Media. Language in society. Blackwell.

Chen, Zhao, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2017. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. arXiv preprint arXiv:1711.02257.

Conneau, Alexis, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, et al. 2018. XNLI: Evaluating cross-lingual sentence representations. In Proc. of EMNLP 2018, pages 2475–2485, Brussels, Belgium, November.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proc. of NAACL-HLT 2019, pages 4171–4186, Minneapolis, Minnesota, June.

Dong, Daxiang, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In Proc. of ACL 2015, pages 1723–1732, Beijing, China, July.

Go, Alec, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(12):2009.

Jain, Alankar, Bhargavi Paranjape, and Zachary C. Lipton. 2019. Entity projection via machine translation for cross-lingual NER. In Proc. of EMNLP 2019, pages 1083–1092, Hong Kong, China, November.

Jean, Sébastien, Orhan Firat, and Melvin Johnson. 2019. Adaptive scheduling for multi-task learning. arXiv preprint arXiv:1909.06434.

Johnson, Melvin, Mike Schuster, Quoc V. Le, Maxim Krikun, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. Transactions of the Association for Computational Linguistics, 5:339–351.

Kreutzer, Julia, Artem Sokolov, and Stefan Riezler. 2017. Bandit structured prediction for neural sequence-to-sequence learning. In Proc. of ACL 2017, pages 1503–1513, Vancouver, Canada, August.

Luong, Minh-Thang, Quoc V Le, Ilya Sutskever, et al. 2015. Multi-task sequence to sequence learning. arXiv preprint arXiv:1511.06114.

Mohammad, Saif M., Mohammad Salameh, and Svetlana Kiritchenko. 2016. How translation alters sentiment. Journal of Artificial Intelligence Research, 55(1):95–130, January.

Nguyen, Khanh, Hal Daumé III, and Jordan Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. In Proc. of EMNLP 2017, pages 1464–1474, Copenhagen, Denmark, September.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In Proc. of ACL 2002, pages 311–318, Philadelphia, PA, USA, July.

Ranzato, Marc'Aurelio, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In Proc. of ICLR 2016, San Juan, Puerto Rico, May.

Schwenk, Holger and Xian Li. 2018. A Corpus for Multilingual Document Classification in Eight Languages. In Proc. of LREC 2018, Miyazaki, Japan, May.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proc. of ACL 2016, pages 1715–1725, Berlin, Germany, August.

Shen, Shiqi, Yong Cheng, Zhongjun He, Wei He, et al. 2016. Minimum risk training for neural machine translation. In Proc. of ACL 2016, pages 1683–1692, Berlin, Germany, August.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(56):1929–1958.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems 27, pages 3104–3112. Curran Associates, Inc.

Tebbifakhr, Amirhossein, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Machine translation for machines: the sentiment classification use case. In Proc. of EMNLP 2019, pages 1368–1374, Hong Kong, China, November.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, et al. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc.

Williams, Ronald J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256.

Wu, Lijun, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In Proc. of EMNLP 2018, pages 3612–3621, Brussels, Belgium, November.

Zhang, Yu and Qiang Yang. 2017. A survey on multi-task learning. arXiv preprint arXiv:1707.08114.