LIMSI Submission for WMT'17 Shared Task on Bandit Learning

Guillaume Wisniewski
LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, 91 405 Orsay, France
guillaume.wisniewski@limsi.fr

Abstract

This paper describes LIMSI participation to the WMT'17 shared task on Bandit Learning. The method we propose to adapt a seed system trained on out-domain data to a new, unknown domain relies on two components. First, we use a linear regression model to exploit the weak and partial feedback the system receives by learning to predict the reward a translation hypothesis will get. This model can then be used to score hypotheses in the search space and translate source sentences while taking into account the specificities of the in-domain data. Second, we use the UCB1 algorithm to choose which of the 'adapted' or 'seed' system must be used to translate a given source sentence in order to maximize the cumulative reward.

Results on the development and train sets show that the proposed method does not succeed in improving the seed system. We explore several hypotheses to explain this negative result.

1 Introduction

The first Bandit Learning for Machine Translation shared task (Sokolov et al., 2017) aims at adapting a 'seed' MT system trained on out-domain corpora to a new domain considering only a 'weak' signal, namely a translation quality judgment rather than a reference translation or a post-edition. Such a situation arises when the user is not a skilled translator but can nevertheless decide whether a translation is useful or not. The signal is qualified as 'weak' as only the score of the translation produced by a system can be known, the same sentence can not be translated twice and no reference is ever revealed.

Adapting a MT system from a weak signal raises three main challenges. First, the parameters of the MT system must be estimated without knowing the reference translation which rules out most of the usual optimization methods for MT such as MERT, MIRA or the computation of likelihood at the heart of NMT systems (Neubig and Watanabe, 2016). Second, the system must be trained in a 'one-shot' way as each source sentence can only be translated once and will result in a single reward. Third, no information about the target domain is available and its specificities must be discovered 'on-the-fly'.

To address these challenges, we propose an adaptation method that relies on two components. First, we use a linear regression model to exploit the weak and partial feedback the system receives by learning to predict the reward a translation hypothesis will get. This model can then be used to score hypotheses of the search space and translate source sentences while taking into account the specificities of the in-domain data. Second, we use the UCB1 algorithm to choose which of the 'adapted' or 'seed' system must be used to translate a given source sentence in order to maximize the cumulative reward.

The rest of this article is organized as follows: we will first describe the shared task and the different challenges it raises (§2). Then we will describe the proposed method (§3–4) and discuss their results in §5.

2 Task Description

Bandit learning for MT follows an online learning protocol: at the i-th iteration, a new source sentence x_i is received; the learner translates it and gets a reward $r_i \in [0, 1]$ (a smoothed sentence-level BLEU score in this shared task). The higher the reward, the better the translation but no infor-

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 674–679
Copenhagen, Denmark, September 711, 2017.

mation about the actual reference is available. The goal of the task is to maximize the cumulative reward over T rounds: $\sum_{i=1}^{T} r_i$.

Maximizing the cumulative reward faces an exploration/exploitation dilemma: if all the sentences are translated using the seed system (i.e. a system trained on out-domain data), the specificities of the domain will never be taken into account and only 'average' translations will be predicted (assuming the seed system is 'good enough'). However, training a new MT system from scratch is also not a good strategy as, at the beginning the system will predict many bad translations which i) will have a negative impact on the cumulative reward ii) might hinder training as the system will only see bad hypotheses (i.e. only a small part of the search space of a MT system will be explored). Moreover, as no information about the target domain is available, the seed system may be, in fact, very good for some input sentences and the best strategy will simply be not to do any adaptation.

3 System Overview

We will now describes the two components of our system: the first one (§3.1) will allow us to exploit the weak and partial feedback we receive and the second one (§3.2) will allow to discover the MT system that translates in-domain data the best.

3.1 Optimizing a MT System from Weak Feedback

Estimating the parameters of a MT system from the rewards can not be done with the usual MT optimization methods: as the reference is not known, it is impossible to score a n-best list as required by methods optimizing a classification criterion such as MERT or MIRA (Neubig and Watanabe, 2016). Moreover, as only one translation hypothesis is scored, methods optimizing a ranking criterion, such as PRO, can also not be used.

Instead we propose to simply learn a linear regression to predict the reward a translation hypothesis will get based on a joint feature representation $\phi(h_i, x_i)$ of the hypothesis and the source sentence. Using a linear model allows us to easily integrate it into the decoder to score translation hypotheses: given a weight vector w, translating a source sentence x consists in looking, in the search space, for the hypothesis that maximizes the predicted reward, which amounts to finding the longest path in a weighted directed acyclic graph (Wisniewski et al., 2010; Wisniewski and Yvon, 2013).

More precisely the weights of the MT system are chosen by optimizing the regularized mean squared error (MSE):

$$\min_{w} \sum_{i} (r_i - w \cdot \phi(h_i, x_i))^2 + \lambda_2 \cdot ||w||_2^2 + \lambda_1 \cdot ||w||_1 \quad (1)$$

where λ_1 and λ_2 are hyper-parameters controlling the strength of the regularization. Solving Equation (1) with a stochastic gradient descent allows us to update the weight vector each time a new reward is received and to integrate learning in the bandit protocol. Features and optimization methods are detailed in Section 4.2.

It is important to note that in the context of Bandit MT, examples are not independently distributed: the score of the i-th observation depends on the current value of the weight vector that, in turn, depends on all the examples that have been previously observed. This is a second aspect of the exploration/exploitation dilemma described in Section 2 as we have to trade off exploration of the search space (to ensure that we correctly predict the reward of any 'kind' of hypotheses and eventually discover better translations) while focusing on the part of the search space that contains, according to our current knowledge (i.e. value of the weight vector), the best hypotheses.

In the following, we will denote ADAPTED the MT system that uses the predicted reward to translate a source sentence.

3.2 Trading off Exploration and Exploitation

Our system relies on the observation that each new source sentence can be translated by different systems: either the SEED system, the parameters of which have been estimated on an out-domain data set or the ADAPTED system the parameters of which are continuously updated from the rewards. The bandit learning task aims at deciding, for a given input sentence, which system must be used to translate it in order to maximizes the cumulative reward.

The quality of a translation predicted by a given system i can be modeled by a $[0, 1]$-valued random variable X_i distributed with an unknown distribution and possessing an unknown expected value μ_i. Would μ_i be known, the best strategy would be to always translate sentences with the system that has the highest μ_i. The challenge here is that

μ_i is unknown and can change over time.

This framework corresponds to the multi-armed bandit scenario (Bubeck and Cesa-Bianchi, 2012). Many algorithms have been proposed to find the best *policy*.[1] In this shared task, we considered the UCB1 algorithm (Auer et al., 2002), that consists in choosing the system that maximizes $\bar{x}_j + \sqrt{2 \log \frac{t}{n_j}}$, where n_j represents the number of times system j was chosen so far, t the number of rounds and $\bar{x}_j$ the empirical mean reward of the j-th system. After each decision, a reward is observed and used to i) update the estimated empirical mean reward of the system that has just been chosen and ii) update the weight vector of the ADAPTED system by doing one SGD step. Intuitively, this strategy selects a decision that has either a 'good' expected reward or has not been played for long. Importantly it never permanently rules out a system no matter how poorly it performs.

It can be proven (Auer et al., 2002) that the UCB1 expected cumulative regret after T rounds is at most $\mathcal{O}\left(\sqrt{K \cdot T \cdot \log T}\right)$ where K is the number of decisions that can be made. This means that the difference between the cumulative reward achieved by the UCB1 strategy and the cumulative reward that would have been achieved by always making the best decision is upper-bounded, i.e. the UCB1 will allow us to discover which was the best decision to make without making too many bad decisions.

In the following, we denote UCB1 the strategy that consists in using the UCB1 algorithm to choose between the SEED and ADAPTED translation systems.

3.3 Variants

After analyzing our results on the development set (see §5), we decide to consider two more strategies:

- UCB1-SELECT that considers the same systems as the UCB1 strategy but only the translation hypothesis associated to a reward r in $[0.1, 1[$ are considered to estimate the weights of the ADAPTED system (other observations are discarded);

- UCB1-SAMPLING in which two more MT systems are considered (in addition to the ADAPTED and SEED systems): the first one, SAMPLE-SEED samples translations from the search space according to their score predicted by the SEED system (the higher its predicted score, the higher the probability to select this hypothesis); the other one, SAMPLE-UNIFORM samples translation hypotheses uniformly from the search space.

The latter strategy allows us to increase the diversity of translation hypotheses seen when estimating the weights of the ADAPTED system. The former is motivated by our observations that many good translation hypotheses have very low rewards because the references used to compute them are not a direct translation that can be produced by the MT system (i.e. the references are unreachable) or that many source sentence do not actually need to be translated (i.e. the source and the reference are the same). Table 1 shows such examples. We assume that these observations hinder the estimation of the model used to predict the rewards has its gold value is completely unrelated to the features describing the hypothesis.

source	einfach genial und absolut cool !
hyp.	simply brilliant and totally cool !
score	0.008633400213704501
source	schwarz gr.xxl / xxxl
hyp.	black gr.xxl / xxxl
score	0.0360645288
source	00603.117 , bt .
hyp.	00603.117 , bt .
score	1.0

Table 1: Example of a 'good' translation with very bad rewards and of a perfect translation.

4 Experimental Details

4.1 The SEED System

We consider as our SEED system a phrase-based system trained using the standard Moses pipeline (Koehn et al., 2007): all corpora are cleaned[2] and tokenized; compounds are split on the German side using our re-implementation of (Koehn and Knight, 2003). Parallel data are

[1] A policy is a randomized algorithm which makes a decision in each round based on the history of decisions and observed rewards so far

[2] Moses scripts are applied in the following order to clean corpora: removing non-printing characters, replacing and normalizing Unicode punctuation, lowercasing, pretokenizing.

aligned using FASTALIGN (Dyer et al., 2013) and 5-gram language models is estimated using KENLM (Heafield et al., 2013).

The language model is estimated on the monolingual corpus resulting from the concatenation of the EUROPARL (v7), NEWSCOMMENTARY (v12) and NEWSDISCUSS (2015–2016) corpora. At the end, our monolingual corpus contain 193,292,548 sentences. The translation model is estimated from the `CommonCrawl`, `NewsCo`, `Europarl` and `Rapid` corpora, resulting in a parallel corpus made of 5,919,142 sentences.

Weights of the MT systems are estimated with MERT on `newstest-2016`.

4.2 Training the Regression Model

We use Wowpal Wabbit (Agarwal et al., 2014) to efficiently train a regressor to predict the rewards by optimizing the Mean Squared Error with a stochastic gradient descent. We consider 32 features: the 15 features of a baseline Moses system[3] as well the score of the SEED system. We also consider the logarithm of these features.

To account for the different feature ranges and the mix of continuous and discrete features, we enhance the standard SGD by adding the following three additional factors affecting the weight updates when optimizing the MSE objective function:

- normalized updates to adjust for the scale of each feature (Ross et al., 2013);
- adaptive, individual learning rate for each feature (Duchi et al., 2011);
- importance aware update (Karampatziakis and Langford, 2011).

The value of the hyper-parameters λ_1 and λ_2 are chosen by maximizing prediction performance on the 5,000 first examples on the development set.

5 Results

Performance of the proposed methods have been evaluated on the two corpora provided by the shared task organizers: a development set containing about 40,000 sentences and an official training set containing 1,300,000 sentences, which will be use to rank the participants. Unfortunately, given its size, we were not able to translate all the training set.

[3] 1 language model score, 4 translation model scores, 6 scores describing lexical reordering, one distortion score, as well as word, phrase and unknown word penalty

Strategy	Cumulative BLEU
SEED	6970.21399
UCB1	6533.67157
UCB1-SAMPLING	6059.92188
UCB1-SELECT	6596.03351

Table 2: Results on the Development data set

The quality of the systems is evaluated both by the cumulative reward (see §2) and by computing the BLEU score on a specific corpus at different 'checkpoints'.

Table 2 shows the cumulative reward achieved by our systems on the development set. It appears that all the methods we proposed are outperformed by the seed system. Looking at the number of times each system was used by the different strategies (Table 3), shows that, most of the time, the seed system is selected, which confirms that it achieves the best translation performance. Results of the off-line evaluation, reported in Figure 1 and on the training set confirm these observations.

Several hypotheses can be formulated to explain these negative results:

- trying to adapt an MT system by changing only the scores of a few models and without additional resources or knowledge of the target domain may not offer enough flexibility;
- the estimation error of regressor may be too large to discriminate the best translation hypothesis of the search space. In practice the mean squared error on the training data is around 0.06.
- Our exploration strategy is not efficient enough, and the learners never learns to score 'good' hypotheses. Indeed, as shown in Figure 2, most of the hypotheses seen during training are of very low quality or correspond to very short sentences that can be translated trivially. In both cases, extracting useful information is difficult.

Analyzing these hypotheses in more depth is difficult without access to the references and results on the training set.

Strategy	Out-Domain	In-Domain	Sample Moses	Sample Uniform
SEED	100%	—	—	—
UCB1	90.77%	9.23%	—	—
UCB1-SAMPLING	78.04%	7.67%	7.36%	6.94%
UCB1-SELECT	90.15%	9.85%	—	—

Table 3: Number of times each translation system is chosen by the UCB1 strategy on the development set. 'Out-Domain' refers to the seed system, In-Domain to the system trained on the rewards and the last two systems to systems sampling randomly hypotheses from the search space.

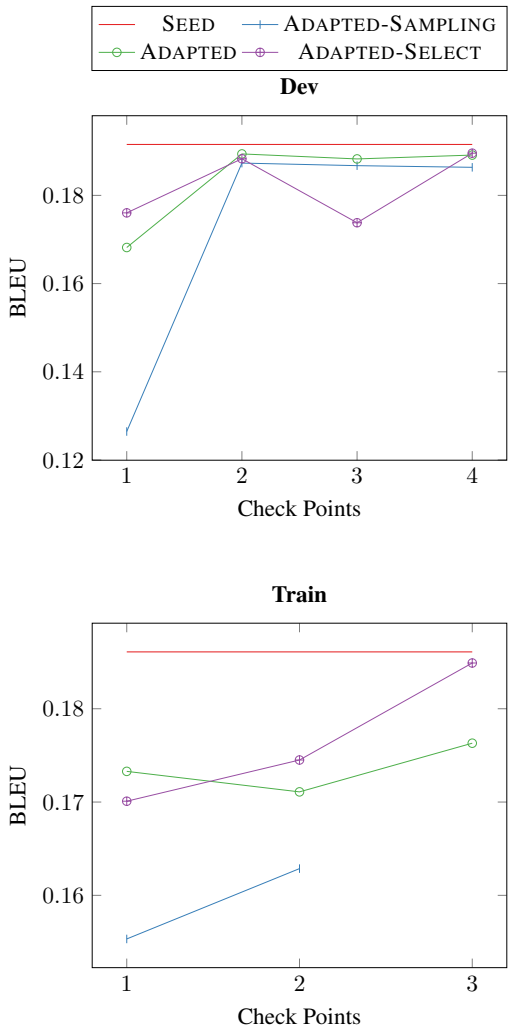

Figure 1: Evolution of the BLEU score at the different 'check-points' of the development and training datasets.

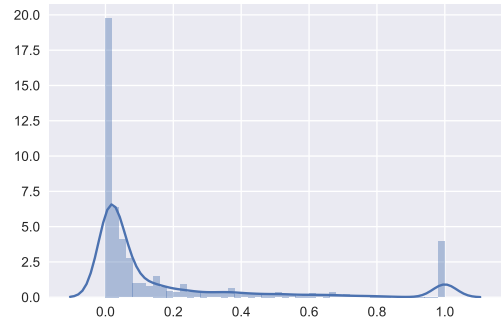

Figure 2: Distributions of the rewards the SEED system got on the development dataset.

Acknowledgments

This work has been partially funded by the *Agence Nationale de la Recherche* (ParSiTi project, ANR-16-CE33-0021). Warm thanks to François Yvon for his feedback on this work.

References

Alekh Agarwal, Olivier Chapelle, Miroslav Dudík, and John Langford. 2014. A reliable effective terascale linear learning system. *J. Mach. Learn. Res.*, 15(1):1111–1133.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256.

Sébastien Bubeck and Nicolò Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

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

Nikos Karampatziakis and John Langford. 2011. Online importance weight aware updates. In *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*, pages 392–399. AUAI Press.

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 *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.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 187–193, Stroudsburg, PA, USA. Association for Computational Linguistics.

Graham Neubig and Taro Watanabe. 2016. Optimization for statistical machine translation: A survey. *Computational Linguistics*, 42(1):1–54.

Stéphane Ross, Paul Mineiro, and John Langford. 2013. Normalized online learning. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*. AUAI Press.

Artem Sokolov, Julia Kreutzer, Kellen Sunderland, Pavel Danchenko, Witold Szymaniak, Hagen Fürstenau, and Stefan Riezler. 2017. A shared task on bandit learning for machine translation. In *Proceedings of the Second Conference on Machine Translation (WMT)*.

Guillaume Wisniewski, Alexandre Allauzen, and François Yvon. 2010. Assessing phrase-based translation models with oracle decoding. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 933–943, Cambridge, MA. Association for Computational Linguistics.

Guillaume Wisniewski and Franois Yvon. 2013. Fast large-margin learning for statistical machine translation. In *International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2013)*, page 12p.