# CUNI systems for WMT21: Terminology translation Shared Task

**Josef Jon** and **Michal Novák** and **João Paulo Aires** and **Dušan Variš** and **Ondřej Bojar**
Charles University
{jon,aires,varis,bojar}@ufal.mff.cuni.cz

## Abstract

This paper describes Charles University submission for Terminology translation Shared Task at WMT21. The objective of this task is to design a system which translates certain terms based on a provided terminology database, while preserving high overall translation quality. We competed in English-French language pair. Our approach is based on providing the desired translations alongside the input sentence and training the model to use these provided terms. We lemmatize the terms both during the training and inference, to allow the model to learn how to produce correct surface forms of the words, when they differ from the forms provided in the terminology database. Our submission ranked second in Exact Match metric which evaluates the ability of the model to produce desired terms in the translation.

## 1 Introduction

Terminology integration, or, more generally, constrained translation in NMT was extensively studied in recent years. Lexically constrained translation means that aside from the source sentence, we have available some additional knowledge of what tokens or expressions should appear in the translation and we want to force the system to include them in the generated output. Three main ways of enforcing these constraints have been studied.

First, replacing the source part of the constraint that is found in the source sentence with a placeholder which is then copied by the model into the output. There it gets replaced by the target part of the constraint (Luong et al. (2015); Crego et al. (2016)).

Second way is to modify the decoding search algorithm in a way that only allows hypotheses containing the constraints to be marked as finished (Anderson et al. (2017); Hasler et al. (2018); Chatterjee et al. (2017); Hokamp and Liu (2017); Post and Vilar (2018); Hu et al. (2019))

Finally, some works focus on providing the constraints directly to the model as part of the input sequence. The model is trained to incorporate these constraints into the output, for example Dinu et al. (2019); Chen et al. (2020); Song et al. (2019) or Bergmanis and Pinnis (2021).

As apparent from previous paragraphs, the problem of integrating lexical constraints into NMT is well studied, but one issue was largely ignored. In inflected languages, the surface form of the constraint in the output cannot be known beforehand, as there are usually many possible ways to translate a sentence and many of them need different surface forms of the constraint to be fluent and grammatically correct. For example, let's say we have a terminology database containing term pair *influenza -> grippe* and this source sentence:

During the 2018-2019 **influenza** season.

Possible correct translation is:

Pendant la saison **grippale** 2018-2019.

Where the term base noun form *grippe* is inflected into adjective *grippale*. Traditional constraint integration methods will try to enforce the term DB form *grippe* instead.

We have studied this problem in our recent work (Jon et al., 2021) concurrently with Bergmanis and Pinnis (2021), who used a very similar approach. Both works use different languages and evaluation pipelines and both show that the proposed approach is feasible.

## 2 Method

NMT models are known to produce fluent, consistent and grammatically correct outputs (Popel et al., 2020). Thus, it makes sense to utilize this ability of the model to inflect the constraint into correct form, instead of trying to disambiguate the form externally. Our approach is based on annotating



*Proceedings of the Sixth Conference on Machine Translation (WMT)*, pages 828–834
November 10–11, 2021. 

source sentences with the desired target constraints and training the model to incorporate these constraints into the output. We publish our preprocessing scripts at https://github.com/ufal/bergamot/wmt21-terminology

### 2.1 Term annotation

There are multiple possibilities in how to exactly annotate the source sentence. For example, let's say the terminology database contains entries:

*runny nose -> nez qui coule*
*fever -> fièvre*

and we have a sentence:

*And are you having a **runny nose** or **fever**?*

One way is to replace the part of the source sentence containing the source constraint with the target part of the constraint:

*And are you having a **nez qui coule** or **fièvre**?*

Another option is to insert the translation tokens after the source part of the constraint and use factors to mark which tokens of a sentence belong to source constraint, which tokens are part of the target constraint and which are neither. For example, if factor with value 2 denotes that the token is part of the translation, value 1 means that the token is part of a source constraint and 0 means that it is just ordinary token, we get:

$And_0$ $are_0$ $you_0$ $having_0$ $a_0$ ***runny***$_1$ ***nose***$_1$ ***nez***$_2$ ***qui***$_2$ ***coule***$_2$ $or_0$ ***fever***$_1$ ***fièvre***$_2$ $?_0$

We use simpler method to integrate the constraints in our systems: we append them to the source sentence as a suffix, separated by a special token (<sep>) and in case of multiple constraints for a single sentence, we separate them by a different token (<c>):

*And are you having a **runny nose** or **fever**? <sep> **nez qui coule** <c> **fièvre***

For more details about the possible modifications of our method, comparisons with other approaches and detailed evaluation and analysis, we refer the reader to our previous work (Jon et al., 2021).

### 2.2 Training data generation

We prepare synthetic constraints for parallel training data by sampling random token subsequences from the target sentence. These subsequences are used as a suffix for the source sentence as described earlier. There is a number of parameters guiding this process. Every token in a sentence can become a start of a constraint with probability $s$. Unless stated otherwise, we set $s = 0.1$. Any subsequent token in an open constraint can end the constraint with probability $e = 0.75$. We permit multiple non-overlapping constraints for a sentence. We skip the sentence for constraint generation (i.e. leave it without any constraints) with probability $n = 0.1$. In pseudocode:

```
s=0.1
e=0.75
n=0.1
for sent in text:
  r=random()
  constraints=[]
  if r > n:
    open=False
    constraint=""
    for t in tokens(sent):
      r=random()
      if open:
        if r < e:
          constraints.append(constraint)
          open=False
        else:
          constraint+=t
      else:
        if r < s:
          constraint+=t
          open=True
  print(sent, constraints)
```

Since the task allows for multiple target variants for a single source term, we have to account for such possibility in our training data generation. We assume that each generated constraint can have a variant with probability $v = 0.1$. This variant is sampled randomly (with no relation to the source sentence) from n-grams extracted from the target training corpus (so it is not a part of a current target sentence, but it is still a plausible subsequence in the target language). The variant has the same number of tokens as the original constraint with probability $l = 0.9$, otherwise the length of the variant is taken from triangular distribution between 1 and 9 with mode 2. The variants of a single constraint are delimeted with another special token <v>. None of the probabilities were tuned for improving results, we chose them based on manual inspection of the generated data. We use values that produced similar counts and lengths of the constraints as in

the validation set.

### 2.3 Lemmatization

The training data generation method described above works, but suffers from the issues described in the introduction – the system learns to generate only the exact tokens supplied as constraints in the suffix, but doesn't account for different possible inflections of the constraints in different contexts. To overcome this issue, we lemmatize the constraints both during the training and during test time. This way, the model learns to not only generate the correct words in the output, but also to correctly inflect them.

### 2.4 Source-side terminology matching

To find term pairs from terminology database in the input text, we lemmatize both the database source side and input sentences and search for the terms that appear either on lemma or surface form level. Since our lemmatizer works with context, we lemmatize both the text and the database word by word to ensure consistent lemmas. For the models trained with lemmatized constraints, we lemmatize also the target side of the terminology database and anntote the source sentence with lemmas of the target terms, instead of the surface forms.

## 3 Experiments

### 3.1 Data

We used all English-French corpora allowed by the organizers, aside from Paracrawl (with the exception of one model, which is marked). Namely this means Europarl v10, Common Crawl, UN Parallel Corpus v1.0, News Commentary v16 and Gigaword. We used WMT15 news test set as our validation set. After deduplication and filtering, the resulting training set consists of 24.6M sentences without Paracrawl and 125.9M including Paracrawl.

### 3.2 Tools

We use MarianNMT (Junczys-Dowmunt et al., 2018) to train Transformer-big models with standard parameters (Vaswani et al., 2017). The corpora are filtered using Moses cleaning script[1] and `fasttext` langid (Joulin et al., 2016). We split the text into subwords using FactoredSegmenter[2] based on SentencePiece (Kudo and Richardson, 2018) and lemmatize using UDPipe (Straka and Straková, 2017). BLEU scores are computed using SacreBLEU (Post, 2018), other metric are obtained by an evaluation script provided by the organizers[3](ibn Alam et al., 2021).

### 3.3 Evaluation

The script provided by the task organizers computes multiple metrics: BLEU, (Lemmatized) Exact Match, Window overlap and 1-TERm.

Exact match is a fraction of constraints which were produced in the outputs (the output sentences are lemmatized and the search is performed on both lemma and surface form level). This metric can be cheated in two ways – first, the system can place the target constraint at arbitrary place in the output, e.g. we can just translate with a non-constrained MT model, append the constraints at the end and obtain a perfect score. Second way is related to lemmatization – the system can produce any valid surface form of the constraint and even though this form is not grammatically correct in context of the output sentence, it still gets counted as matching. On the other hand, without lemmatization, only the word forms listed in the terminology database would get accepted, which would not cover all the possible correct forms.

Window overlap aims to overcome the first shortcoming of EM by evaluating placement of the constraint in the output. For each constraint in the translation and in the reference, windows of $n$ tokens are extracted and compared with each other to see if the system places the constraint in similar context as in the reference. 2 and 3 token windows are used.

TERm metric is weighted TER which uses higher weights for tokens which are part of a term from terminology database to increase sensitivity to differences in the terminology. In the experiments, we observed that 1-TERm score is influenced mainly by the overall translation quality and less so by the term integration. We believe that this metric alone is also not sufficient for comparing ability to integrate constraints in different models, as the results seem to rely mainly on the "baseline" model performance, i.e. big general NMT model, trained on more data, which provides better overall translation quality, but does not explicitly

[1] https://github.com/marian-nmt/moses-scripts

[2] https://github.com/microsoft/factored-segmenter

[3] https://github.com/mahfuzibnalam/terminology_evaluation

| Constraints | Corpus | Variants | BLEU | EM | window 2 | window 3 | 1-TERm |
|---|---|---|---|---|---|---|---|
| None | Base | - | 43.976 | 0.862 | 0.289 | 0.283 | 0.584 |
| None | Base+paracrawl | - | 45.084 | 0.851 | 0.283 | 0.279 | 0.587 |
| None | Base+bt | - | 42.319 | 0.834 | 0.282 | 0.275 | 0.575 |
| SF | Base | no | 43.771 | 0.953 | 0.297 | 0.290 | 0.581 |
| SF | Base | yes | 41.656 | 0.982 | 0.253 | 0.255 | 0.555 |
| Lemm | Base | yes | 42.317 | 0.919 | 0.278 | 0.274 | 0.552 |
| Lemm | Base | no | 44.959 | 0.961 | 0.302 | 0.296 | 0.591 |
| Lemm* | Base | no | 44.623 | 0.909 | 0.292 | 0.288 | 0.588 |
| Final combined | - | - | 45.590 | 0.989 | 0.309 | 0.304 | 0.600 |

Table 1: Results of our models on official validation set. The first column specifies whether the constraint were lemmatized (*Lemm*) or not SF (*SF*), second one shows which part of copora we used. Base means all parallel data allowed by the organizers with exception of Paracrawl. Third column says whether we provided all possible variants of the target term from terminology database to the model, on we only the first one. Asterisk in *Constraints* column means that the model was trained with these form of constraints, but no constraints were provided during the test time.

integrate constraints, may obtain higher scores than a smaller constrained model with perfect constraint integration ability.

### 3.4 Results

We trained our models by techniques described earlier and we present metrics computed by the official evaluation script in Table 1. Due to time and computing constraints, most of the models were trained without Paracrawl corpus and we only trained one baseline on dataset including Paracrawl for comparison. We compared integrating constraints in the surface form (so the model needs to produce exactly the same token as provided in the input) and constraints in lemmatized form (the model can produce different inflection of the provided constraint). We also compared providing all possible variants of the target constraint from terminology database (delimeted by <v>, as described earlier), or just the first possible translation.

We see that in most metrics, the model which is trained with lemmatized constraints and uses only one variant performs the best. Systems trained with multiple variants of the target term show large degradation in BLEU scores. We suppose one of the problems in our method is that during training, only the true constraint variant from the target is plausible translation of the source, others are n-grams sampled randomly from the whole corpus. Thus, the negative samples are very easy to distinguish during the training, but during the test time, the variants are provided by the term base and they are all plausible in the context. We will analyse these results further in the future.

Our final primary submission is a combination of all the models. They are ranked by their respective BLEU scores on validation set and we check if the produced translation contains the desired term either at lemma level. We use the best ranking systems' translation that does, or, in case none of the systems produced the term, we use the translation of baseline system.

The task organizers provide test set results.[4] Two metrics were considered for the ranking. First, COMET (Rei et al., 2020), which evaluates general translation quality without special regard for specific terminology. Secondly, exact match, which measures how many of the desired constraints were actually produced in the output, but suffers from the issues described earlier. Our primary submission was ranked on joint 6th-10th place out of 21 systems according to COMET and 1st-3rd according to exact match.

### 3.5 Error analysis

Our submitted system did not cover 10 out of 872 term occurrences in the validation set. We analyse these ten errors manually. Six of these errors are related to casing, notably by translating *SARS-CoV* as *Sars-CoV*, instead of keeping the original casing (five occurrences). This is caused by our lemmatization pipeline, which produces *Sars* as lemma of *SARS*. We confirmed that after manually fixing the input and restoring the original casing, the system produces correct output. Other five examples classified as errors are presented in Table 2.

[4] https://docs.google.com/spreadsheets/d/13-1kwDq9yerehSF4No6ZTLqPXjSaL7HOsksnZDjjO-Y/

| i | Source | Target terms | MT output |
|---|---|---|---|
| 1 | Many human **Coronavirus** have their origin in bats. | coronavirus | Beaucoup de **Coronavirus** humains ont leur origine dans les chauves-souris . |
| 2 | Data from these practices are reported online in a weekly return, which includes monitoring weekly rates of influenza-like illness (ILI) and other communicable and **respiratory diseases** in England. | maladies respiratoires / maladies communes des voies respiratoires / maladie respiratoire | Les données relatives à ces pratiques sont communiquées en ligne dans une déclaration hebdomadaire, qui comprend le suivi des taux hebdomadaires de maladies grippales(SG) et d'autres **maladies** transmissibles et **respiratoires** en Angleterre. |
| 3-4 | We will share the protocol with UK colleagues and the I-MOVE consortium who have recently obtained EU Horizon 2020 funding from the stream "Advancing knowledge for the clinical and public health response to the **novel coronavirus epidemic**" | coronavirus nouveau; epidémie / épidémies / épidémique | Nous partagerons le protocole avec nos collègues du Royaume-Uni et le consortium I-MOVE , qui ont récemment obtenu un financement de l'OMS horizon 2020 dans le cadre du projet «Advancing knowledge for the clinical and public health response to the **novel coronavirus epidemic**» |
| 5 | The statistical methodology is in support of a policy approach to widespread disease **outbreak**, where so-called nonpharmaceutical interventions (NPIs) are used to respond to an emerging pandemic to produce disease suppression. | épidémie / épidémies / épidémique | La méthodologie statistique est à l'appui d'une approche politique face à l'apparition de **maladies à grande échelle**, où les interventions dites non pharmaceutiques (ISP) sont utilisées pour répondre à une pandémie émergente afin d'éliminer les maladies. |

Table 2: Rest of the examples with uncovered terms. *Target terms* column shows possible translations of the source terms (bold) as provided in the terminology database.

Another casing error occurs in translation of the sentence (1) in the table. The model keeps the original source casing, but the evaluation script only checks for lower-case *coronavirus*. This sentence is also actually part of unsplit and wrongly tokenized source line *The large number of host bat and avian species, and their global range, has enabled extensive evolution and dissemination of coronaviruses.Many human coronavirus have their origin in bats.* This may be a source of further confusion for the model.

In example (2), the related terminology DB pair is *respiratory diseases -> maladies respiratoires*. In the model output, the adjective *transmissibles* is interjected between the terms, which is probably not an error from human point of view, but is hard to evaluate automatically.

In example (3-4), the model does not translate the name of the project in quotes, thus it does not produce the desired translations of both *epidemic -> épidemie* and *novel coronavirus -> coronavirus nouveau* .

Finally, (5) is a true failure of the model to use the provided term. The sentence produced by the model is a plausible and semantically correct translation, but it is not using the desired term. For further analysis, we manually replaced the produced translation of the term (*maladies à grande échelle*) with the term from the terminology database (*épidémie*). We computed cross-entropy scores for the modified sentence both with and without providing the constraint to the model. We saw that when provided with the constraint, the modified translation is more probable than without the constraint (but still slightly less probable than the translation that was actually produced.) This shows that the method still partially works in this case, but the bias towards producing the term in the output needs to be stronger – we plan to explore this further using contrastive learning.

## 4 Conclusion

We describe our submission to Terminology translation Shared Task at WMT21. We show our method can effectively incorporate the terminology without negative effects on overall translation quality. We analysed all ten examples in the validation set where our model did not cover the desired term constraint and we show that most of them can be explained by preprocessing issues.

## Acknowledgements

Our work is supported by the Bergamot project (European Union's Horizon 2020 research and innovation programme under grant agreement No 825303) aiming for fast and private user-side browser translation, GA ČR NEUREM3 grant (Neural Repre-

sentations in Multi-modal and Multi-lingual Modelling, 19-26934X (RIV: GX19-26934X)) and by SVV 260 575 grant.

The work described herein has also been using data provided by the LINDAT/CLARIAH-CZ Research Infrastructure, supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101).

## References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168, Copenhagen, Denmark. Association for Computational Linguistics.

Guanhua Chen, Yun Chen, Yong Wang, and Victor O.K. Li. 2020. Lexical-constraint-aware neural machine translation via data augmentation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3587–3593. International Joint Conferences on Artificial Intelligence Organization. Main track.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

Md Mahfuz ibn Alam, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, and Vassilina Nikoulina. 2021. On the evaluation of machine translation for terminology consistency.

Josef Jon, João Paulo Aires, Dusan Varis, and Ondřej Bojar. 2021. End-to-end lexically constrained machine translation for morphologically rich languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4019–4033, Online. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.