Log-Linear Reformulation of the Noisy Channel Model for Document-Level Neural Machine Translation

Sébastien Jean
New York University
sj2233@nyu.edu

Kyunghyun Cho
New York University
kyunghyun.cho@nyu.edu

Abstract

We seek to maximally use various data sources, such as parallel and monolingual data, to build an effective and efficient document-level translation system. In particular, we start by considering a noisy channel approach (Yu et al., 2020) that combines a target-to-source translation model and a language model. By applying Bayes' rule strategically, we reformulate this approach as a log-linear combination of translation, sentence-level and document-level language model probabilities. In addition to using static coefficients for each term, this formulation alternatively allows for the learning of dynamic per-token weights to more finely control the impact of the language models. Using both static or dynamic coefficients leads to improvements over a context-agnostic baseline and a context-aware concatenation model.

1 Introduction

Neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015) has been reported to reach near human-level performance on sentence-by-sentence translation (Läubli et al., 2018). Going beyond sentence-level, document-level NMT aims to translate sentences by taking into account neighboring source or target sentences in order to produce a more cohesive output (Jean et al., 2017; Wang et al., 2017; Maruf et al., 2019). These approaches often train new models from scratch using parallel data.

In this paper, in a similar spirit to Voita et al. (2019a); Yu et al. (2020), we seek a document-level approach that maximally uses various available corpora, such as parallel and monolingual data, leveraging models trained at the sentence and document levels, while also striving for computational efficiency. We start from the noisy channel model (Yu et al., 2020) which combines a target-to-source translation model and a document-level language model. By applying Bayes' rule, we reformulate this approach into a log-linear model. It consists of a translation model, as well as sentence and document-level language models. This reformulation admits an auto-regressive expression of token-by-token target document probabilities, facilitating the use of existing inference algorithms such as beam search. In this log-linear model, there are coefficients modulating the impact of the language models. We first consider static coefficients and, for more fine-grained control, we train a *merging module* that dynamically adjusts the LM weights.

With either static or dynamic coefficients, we observe improvements over a context-agnostic baseline, as well as a context-aware concatenation model (Tiedemann and Scherrer, 2017). Similarly to the noisy channel model, our approach reuses off-the-shelf models and benefits from future translation or language modelling improvements.

2 Log-linear reformulation of the noisy channel model

Given the availability of various heterogeneous data sources that could be used for document-level translation, we seek a strategy to maximally use them. These sources include parallel data, at either the sentence or document level, as well as more broadly available monolingual data.

As the starting point, we consider the noisy channel approach proposed by Yu et al. (2020). Given a source document $(X^{(1)}, \dots, X^{(N)})$ and its translation $(Y^{(1)}, \dots, Y^{(N)})$, they assume a generation process where target sentences are produced from left to right, and where each source sentence is translated only from the corresponding target sentence. Under these assumptions, the probability of a source-target document pair is given by

Proceedings of 4th Workshop on Structured Prediction for NLP, pages 95–101
November 20, 2020.

$$P(X^{(1)},\ldots,X^{(N)},Y^{(1)},\ldots,Y^{(N)}) = \prod_{n=1}^{N} P(X^{(n)}|Y^{(n)})P(Y^{(n)}|Y^{(<n)})$$

As such, the conditional probability of the target document given the source is expressed by

$$\begin{aligned} P(Y^{(1)},\ldots,Y^{(N)}|X^{(1)},\ldots,X^{(N)}) &\propto \prod_{n=1}^{N} P(X^{(n)}|Y^{(n)})P(Y^{(n)}|Y^{(<n)}) \\ &= \prod_{n=1}^{N} \underbrace{P(Y^{(n)}|X^{(n)})\frac{P(Y^{(n)}|Y^{(<n)})}{P(Y^{(n)})}}_{\propto P(Y^{(n)}|X^{(n)},Y^{(<n)})}. \end{aligned}$$

We therefore generate context-aware translations by combining a translation model (TM) $P(Y^{(n)}|X^{(n)})$ with both sentence-level $P(Y^{(n)})$ and document-level $P(Y^{(n)}|Y^{(<n)})$ language models (LM). To calibrate the generation process, we introduce coefficients $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ to control the contribution of each language model, which are tuned on a validation set:

$$\begin{aligned} &\log P(Y^{(n)}|X^{(n)},Y^{(<n)}) \\ &= \sum_{i=1}^{L_n} \Big[\log P(y_i^{(n)}|y_{<i}^{(n)},X^{(n)}) \\ &\quad + \alpha \log P(y_i^{(n)}|y_{<i}^{(n)},Y^{(<n)}) \\ &\quad - \beta \log P(y_i^{(n)}|y_{<i}^{(n)}) + C_i^{(n)} \Big], \end{aligned} \tag{1}$$

where $C_i^{(n)}$ is a normalization constant and L_n is the target sentence length.

Similarly to the noisy channel approach (Yu et al., 2020), we use off-the-shelf translation and language models. As such, future improvements to either translation or language modelling can easily be leveraged. Our reformulation however admits a more efficient search procedure, unlike that by Yu et al. (2020).

2.1 Model parameterization

The translation model is implemented as any auto-regressive neural translation model. We use the Transformer encoder-decoder architecture (Vaswani et al., 2017). Given a source sentence $x_1,\ldots,x_L$, each token and its position are projected into a continuous embedding $s_{0,1},\ldots,s_{0,L}$. These representations are passed through a sequence of M encoder layers that each comprise self-attention and feed-forward modules, resulting in the final representations $s_{M,1},\ldots,s_{M,L}$. The decoder updates target embeddings through similar layers, which additionally attend to the encoder output, to obtain final hidden states $t_{M,1},\ldots,t_{M,L}$. Token probabilities may be obtained by projecting these representations and applying softmax normalization.

Language models are implemented as Transformer decoders without cross-attention. We use a single language model trained on sequences of consecutive sentences to obtain both sentence-level and document-level probabilities.

3 Dynamic merging

As extra-sentential information is not uniformly useful for translation, we propose dynamic coefficients for the different models by generalizing Eq. 1:

$$\begin{aligned} \mathcal{L} = -\sum_{n=1}^{N}\sum_{i=1}^{L_n} \Big[&\log P(y_i^{(n)}|y_{<i}^{(n)},X^{(n)}) \\ &+ \alpha_i^{(n)} \log P(y_i^{(n)}|y_{<i}^{(n)},Y^{(<n)}) \\ &- \beta_i^{(n)} \log P(y_i^{(n)}|y_{<i}^{(n)}) + C_i^{(n)} \Big]. \end{aligned} \tag{2}$$

With the translation and language models kept fixed, the coefficients $\alpha_i^{(n)}$ and $\beta_i^{(n)}$ are computed by an auxiliary neural network which uses $Y^{(<n)}$, $Y^{(n)}$ and $X^{(n)}$. We call this network a *merging module* and implement it as a feed-forward network on top of the translation and language models.

3.1 Dynamic coefficient computation

For every token, the corresponding last hidden states of the translation model, sentence-level LM and document-level LM are concatenated. Each non-final layer ($k = 1,\ldots,K-1$) is a feed-forward block

$$h_k = \text{LN}(h_{k-1} + \text{drop}(W_{k,2}(\text{ReLU}(W_{k,1}h_{k-1}))),$$

where LN and drop respectively denote layer normalization and dropout (Ba et al., 2016; Srivastava et al., 2014). The final layer is similar, but there is no residual connection (and no

dropout) as the final linear transformation projects the result to 2 dimensions, so that $(\alpha, \beta) = W_{K,2}(\text{ReLU}(W_{K,1}h_{K-1}))$.

4 Experiments

4.1 Settings

Data We run experiments on English-Russian data from OpenSubtitles (Lison et al., 2018), which was used in many recent studies on document-level translation (Voita et al., 2019b,a; Mansimov et al., 2020; Jean et al., 2019). Language models are trained on approximately 30M sequences of 4 consecutive sentences (Voita et al., 2019a).The parallel data was originally preprocessed by Voita et al. (2019b), yielding 6M examples. For 1.5M of these data points, the 3 preceding source and target sentences are provided. We use this subset to train the *merging module* that predicts the per-token coefficients for each model. We uniformly set the number of contextual sentences between 1 and 3 to match the test condition.

We apply byte-pair encoding (BPE) (Sennrich et al., 2016), with a total of 32k merge operations, separately on each language pair, as Russian and English use different sets of alphabets.

Models Translation models are standard Transformers in their base configuration (Vaswani et al., 2017). The language model is implemented as a Transformer decoder of the same size, except for a smaller feed-forward dimension $d_{ff} = 1024$. The *merging module* has 2 layers, with $d_{ff} = 1536$.

Learning The translation and language models, as well as the *merging module*, are trained with label smoothing set to 10%. The TM is trained with 20% dropout, while it is set to 10% for the LMs and *merging module*.

Evaluation Translation quality is evaluated with tokenized BLEU on lowercased data, using beam search with its width set to 5. We average 5 checkpoints for the translation models. Sentences are generated from left to right, and the beam is reset for every sentence.

4.2 Results

With our approach, using static coefficients, we reach a BLEU score of 34.31, which is a modest gain of 0.21 BLEU over the baseline and 0.8 over a model trained on concatenated sentences (Table 1). By optimizing dynamic coefficients, we reach a similar score of 34.22.

	BLEU
Baseline	34.10
Concat	33.51
Static coeffs.	34.31
Dynamic coeffs.	34.22
CADec	33.86
DocRepair	**34.60**

Table 1: Test set BLEU scores (beam width 5, all 4 sentences concatenated). CADec and DocRepair results from (Voita et al., 2019a).

β \ α	0	0.2	0.4	0.6
0	31.5	31.0	29.3	26.9
0.2	30.7	**31.7**	31.2	29.5
0.4	23.3	30.1	31.6	31.1
0.6	14.3	21.9	26.9	30.8

Table 2: Greedy validation BLEU (last sentence only) for different static values of α and β. Both LMs are critical to the approach.

DocRepair (Voita et al., 2019a), a two-pass method that post-edits the output of a baseline system, obtains a slightly higher BLEU score of 34.60. Both approaches could be combined by instead post-editing the output of our models, which we leave for future investigation.

BLEU-NLL correlation We observe limited correlation between BLEU and reference NLL (Och, 2003; Lee et al., 2020). On the validation set, the per-token baseline loss (with label smoothing) is 13.09. Using static coefficients, it actually increases to 13.23, while it decreases to 12.86 with dynamic coefficients.

Contribution of each language model (static) Table 2 presents the BLEU scores on the validation set using greedy validation for different static values of α and β. Only using the document-level LM ($\alpha > 0, \beta = 0$) leads to worse performance than the baseline. It is critical to counter-balance the document-level LM with the sentence-level LM.

Dynamic coefficients The dynamic coefficients α and β predicted by the *merging module* are highly correlated (Figure 1 (left)). As a conjecture, this high correlation may be explained by the use of the same language model to obtain both sentence and document-level scores.

Figure 1 (right) shows the average value of the

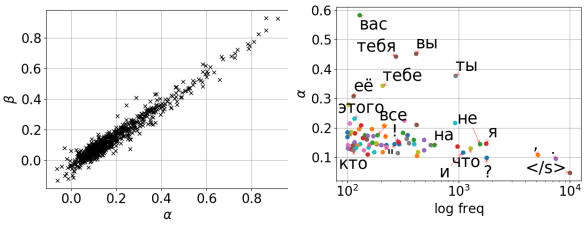

Figure 1: Scatter plot of α and β for tokens appearing at least 100 times over the validation set (left). Average dynamic coefficient α for frequent words over the validation set (right).

	D	LC	I	VP
LM difference	95.5	91.7	71.8	85.6
Baseline	50.0	45.9	53.4	26.6
Concat	84.9	47.7	84.2	78.6
Static	66.6	65.5	56.6	40.2
Dynamic	74.2	51.1	57.8	56.8
CADec	81.6	58.1	72.2	**80.0**
DocRepair	**91.8**	**80.6**	**86.4**	75.2

Table 3: Deixis (D), lexical cohesion (LC), inflection ellipsis (I) and VP ellipsis (VP) accuracy (%). Best scores from translation models only are highlighted.

dynamic coefficient α for frequent words within the validation reference set. In particular, Ты and Вы, which are translations of *you* that depend on plurality and formality, are assigned high weights.

Challenge sets While static and dynamic coefficients lead to similar BLEU, using dynamic coefficients often results in better performance on multiple-choice scoring-based challenge sets targeting specific translation phenomena (Table 3) (Voita et al., 2019b).[1] We conjecture this likely happens because dynamic coefficients can more narrowly focus on particular subsets of target sentences that benefit from document-level context.

5 Related work

Document-level NMT Neural machine translation may be extended to include extra-sentential information in many ways, as surveyed by Maruf et al. (2019). The model architecture may be modified, for example by encoding previous source sentences or generated translations and attending to them (Jean et al., 2017; Wang et al., 2017; Voita et al., 2018; Zhang et al., 2018; Miculicich et al., 2018; Maruf and Haffari, 2018; Tu et al., 2018). Otherwise, by simply concatenating multiple sentences together as input, existing model architectures may be used without additional changes (Tiedemann and Scherrer, 2017; Junczys-Dowmunt, 2019).

Voita et al. (2019b) and Voita et al. (2019a) propose refining the output of a context-agnostic baseline, using a new model trained from either document-level parallel data or from round-trip translated monolingual data. The noisy channel approach similarly uses large-scale monolingual data (Yu et al., 2020) to refine translations, while using arbitrary, and potentially pre-trained, translation or language models, as discussed in Sec. 2.

Our approach shares many similarities with the above, but admits a more straightforward generation process. If desired, we could still rerank the beam search output with a channel model, which might improve general translation quality for reasons not necessarily related to context.

Language modelling Language model probabilities have been used to rerank NMT hypotheses (see, e.g., Stahlberg et al., 2019). Additionally, direct integration of a language model into a translation model, using various fusion techniques, improves generation quality and admits the use of single-pass search algorithms (Gulcehre et al., 2015). To promote diversity in dialogue systems, model scores may be adjusted by negatively weighing a language model (Li et al., 2015).

6 Conclusion

In this paper, we set to use heterogeneous data sources in an effective and efficient manner for document-level NMT. We reformulated the noisy channel approach (Yu et al., 2020) and end up with a left-to-right log-linear model combining a baseline machine translation model with sentence-level and document-level language models.

To modulate the impact of the language models, we dynamically adapt their coefficients at each time step with a *merging module* taking into account the translation and language models. We observe improvements over a context-agnostic baseline and using dynamic coefficients helps capture document-level linguistic phenomena better.

Future directions include combining our approach with MT models trained on back-translated documents, exploring its applicability to other modalities such as vision and speech, and considering deeper fusion of the models.

[1] Using the difference of language models scores gives higher accuracy, but they cannot be used in isolation to generate relevant translations.

Acknowledgements

This work was supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: from pattern recognition to AI), Samsung Research (Improving Deep Learning using Latent Structure) and NSF Award 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Sébastien Jean, Ankur Bapna, and Orhan Firat. 2019. Fill in the blanks: Imputing missing sentences for larger-context neural machine translation. *arXiv preprint arXiv:1910.14075*.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796.

Jason Lee, Dustin Tran, Orhan Firat, and Kyunghyun Cho. 2020. On the discrepancy between density estimation and sequence generation. *arXiv preprint arXiv:2002.07233*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Elman Mansimov, Gábor Melis, and Lei Yu. 2020. Capturing document context inside sentence-level neural machine translation models with self-training. *arXiv preprint arXiv:2003.05259*.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1275–1284.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2019. A survey on document-level machine translation: Methods and evaluation. *arXiv preprint arXiv:1912.08494*.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Felix Stahlberg, Danielle Saunders, Adrià de Gispert, and Bill Byrne. 2019. Cued@ wmt19: Ewc&lms. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 364–373.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *NIPS*.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association of Computational Linguistics*, 6:407–420.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1264–1274.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831.

Lei Yu, Laurent Sartran, Wojciech Stokowiec, Wang Ling, Lingpeng Kong, Phil Blunsom, and Chris Dyer. 2020. Putting machine translation in context with the noisy channel model. *TACL*.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542.

A Expanded derivation

The conditional probability of the target document given the source is expressed by

$$P(Y^{(1)}, ...Y^{(N)}|X^{(1)}, ..., X^{(N)}) =$$
$$\frac{\prod_{n=1}^{N} P(X^{(n)}|Y^{(n)})P(Y^{(n)}|Y^{(<n)})}{P(X^{(1)}, ..., X^{(N)})} =$$
$$\frac{\prod_{n=1}^{N} \frac{P(Y^{(n)}|X^{(n)})P(X^{(n)})}{P(Y^{(n)})} P(Y^{(n)}|Y^{(<n)})}{P(X^{(1)}, ..., X^{(N)})} =$$
$$C(X) \prod_{n=1}^{N} P(Y^{(n)}|X^{(n)}) \frac{P(Y^{(n)}|Y^{(<n)})}{P(Y^{(n)})},$$

where $C(X) = \frac{\prod_{n=1}^{N} P(X^{(n)})}{P(X^{(1)}, ..., X^{(N)})}$ does not affect the optimal target sentences given a source document.

B Hyper-parameters

Translation model We validate models with greedy search. We use the base transformer configuration (Vaswani et al., 2017). We use effective batches of approximately 31500 source tokens and optimize models with Adam (Kingma and Ba, 2014). We follow a learning rate schedule similar to Vaswani et al. (2017), with 16,000 warmup steps and scaled by 4. We experimented with 10% and 20% dropout, obtaining higher validation BLEU with the latter. We use pre-LN transformer layers (Xiong et al., 2020).

Language model We use a similar configuration to the translation model, except with 64,000 warmup steps and post-LN transformer layers (Xiong et al., 2020).

Static coefficients We evaluate greedy validation BLEU with a grid search over $(\alpha, \beta) \in \{0, 0.1, \ldots, 1\} \times \{0, 0.1, \ldots, 1\}$.

Dynamic coefficients We varied the number of layers between 1 and 3. We also considered adding cross-attention within the *merging module*, but we did not observe improvements in preliminary experiments.

C Label smoothing

If we train the *merging module* without label smoothing (instead of 10%), greedy validation BLEU drops by approximately 1 BLEU point. We also observe much higher variability in the coefficients, which may be caused by the unbounded optimal value of α when a target token is the most likely according to the document-level LM.

D Challenge set validation scores

	D	LC
LM difference	95.4	92.6
Baseline	50.0	46.2
Concat	86.6	47.8
Static	65.6	67.8
Dynamic	74.6	50.4

Table 1: Deixis (D) and lexical cohesion (LC) validation accuracy (%).

E Number of parameters

TM: 77,633,536 LM: 29,399,040, *Merging module*: 7,088,642

F Computing infrastructure

We train models with PyTorch 1.2.0 (Paszke et al., 2019). We use a single NVIDIA 1080 Ti or 2080 Ti, running CUDA 10.2 on CentOS Linux 7 (Core).

G Links

Data:
https://box.com/shared/static/qmad0j3e6qknas9nwznyw1w0l5vgpdf4.zip
multi_bleu.perl:
https://raw.githubusercontent.com/moses-smt/mosesdecoder/master/scripts/generic/multi-bleu.perl