La robustesse de la traduction neuronale : les système de traduction automatique neuronale à l'épreuve de la reproductibilité de l'expérience

Guillaume Wisniewski[†], Lichao Zhu[‡], Jean-Baptiste Yunès[♣], Nicolas Ballier[‡ †]

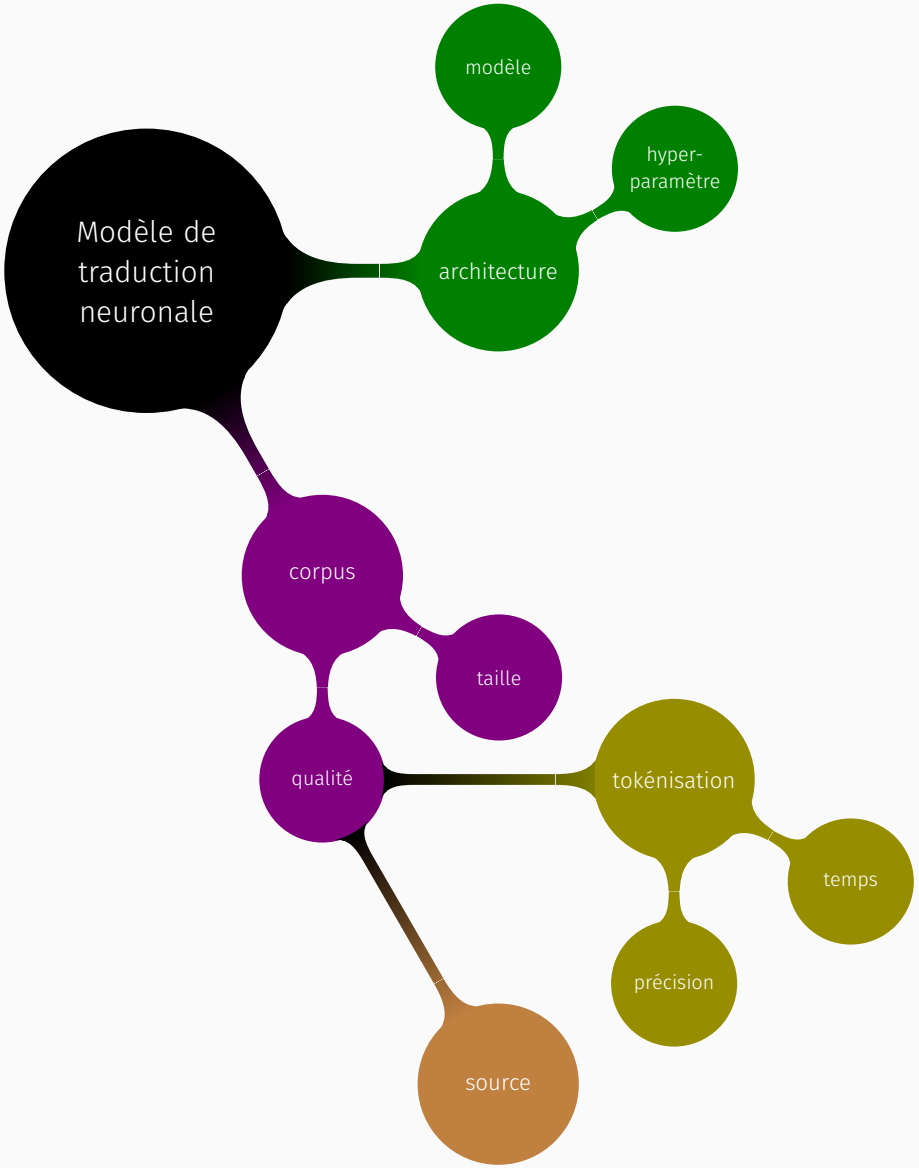
Robustesse des systèmes de TAL – 25 novembre 2022

[†] Université Paris Cité, CNRS, Laboratoire de linguistique formelle, F-75013 Paris, France
[‡] Université Paris Cité, CLILLAC-ARP, F-75013 Paris, France
[♣] Université Paris Cité, CNRS, IRIF, F-75013 Paris, France

Plan

Introduction

Conditionnements des implémentations de modèles de traduction neuronale



Corpus d’entraînement

Corpus

corpus aligné TED2020 (OPUS), en⇒fr

- $>$ 6 900 000 mots, 399 852 lignes
- distributions de corpus *train*, *dev*, *test*
  - *test* : 2 000 phrases
  - *dev* : 2 000 phrases
  - TRAIN : 395 852 phrases

Tokénisations et sous-tokénisations

Tokénisation

1. tokénisation avec SpaCy d'un grand corpus parallèle d'entraînement (en & fr) pour entraîner un model BPE SentencePiece Kudo and Richardson [2018] et SubWord-NMT Sennrich et al. [2016] (Common Crawl, News Commentary, Europarl v7 - WMT2015)
   - en : >116M de mots, >4M lignes
   - fr : >127M de mots, >4M lignes
2. tokénisation du corpus d'entraînement (TEDtalk 2020)

Tokénisation

Normalisations pour trois systèmes NMT

- données en minuscules pour JoeyNMT Kreutzer et al. [2019] et OpenNMT Klein et al. [2017]
- données normalisées et non normalisées pour Nematus Sennrich et al. [2017]

Sous-tokénisation (sentencepiece)

Paramètres de l'entraînement d'un modèle SENTENCEPIECE

- taille de vocabulaire : 32 000
- type de modèle : unigram
- couverture de caractères : 1

Sous-tokénisation (Subword-NMT)

Paramètres de l'entraînement d'un modèle Subword-NMT

- bpe operations : 40 000, bpe threshold : 50
- taille de vocabulaire : $>$34 000 vocabulaires monolingues
- construction d'un dictionnaire combinant la langue source et la cible : 51 868 vocabulaires

Sous-tokénisation (comparaisons)

SubWord-NMT sur un corpus non normalisé	(L@@ aughter) P@@ ut yourselves in my position .
sentencepiece sur un corpus normalisé	_(_laughter _) _put _yourselve s _in _my _position _.

Paramétrage de l’entraînement

Paramètres

- la même architecture TRANSFORMER
  - un décodeur et un encodeur composés chacun de 6 couches avec 8 têtes d'attention à chaque couche
  - une représentation des tokens sur 512 dimensions
  - une couche *feed-forward* de dimension 2 048.
- une quinzaine de paramètres correspondant à la définition des entrées/sorties
- une quinzaine de paramètres décrivant l'architecture TRANSFORMER à proprement parler et les autres correspondant aux paramètres de l'algorithme d'optimisation

Apprentissage des modèles

Nematus, lancées 1 et 2

RÉFÉRENCE	lancée 1	lancée 2
_cela _veut _dire _que _, _pour _chaque _perception _que _nous _posséd ons _, _il _faut _une _cor respondance _avec _ce _qui _l ' _a _précédé _, _sinon _la _continuité _est _perdue _, _et _nous _en _de venons _un _peu _dés orient és _.	_ce _que _cela _signifie _, _c ' _est _que _pour _chaque _perception _que _nous _avons _, _il _doit _mesurer _avec _celle _comme _avant _, _ou _nous _n ' _avons _pas _de _continuité _, _et _nous _de venons _un _peu _dés orient és _.	_cela _signifie _que _, _pour _chaque _perception _que _nous _avons _, _il _faut _calculer _avec _celle _comme _avant _, _ou _nous _n ' _avons _pas _de _continuité _, _et _nous _de venons _un _peu _dés orient és _.

Comparaison externe

	BLEU	CHRF2
JOEYNMT		
run 1	41,6	64,5
run 2	41,1	64,4
OPENNMT		
run 1	37,5	60,7
NEMATUS		
run 1	43,9	65,8
run 2	44,3	65,9

Table 1: Évaluation sur notre ensemble de test TEDTALK de la qualité des traductions obtenues par plusieurs implémentations d'un modèle TRANSFORMER

Trois niveaux d'évaluation de la reproductibilité

- l'exemple du farsi : problèmes de tokenisation pour le farsi
- l'exemple du chinois : trouver le bon tokeniseur pour le chinois avant de sous-tokeniser

Le cas du farsi

Pour le farsi, on peut consdidérer que le principal problème de la tokénisation est le ”ZERO WIDTH NON-JOINER” (U+200C en unicode), pseudo-espace qui devrait être remplacé par une espace inter-mots.

Le cas du chinois (tokénisation)

Évaluation des 6 tokéniseurs du chinois au moyen de distance d'édition à partir des annotations manuelles de 2 000 phrases (MultiParaCrawl) ⇒ Xinyi Zhong (M2 TAL, Inalco) :

hanlp	jieba	**pkuseg**	snownlp	stanza	thulac
1956,91	1781,28	**1968,44**	1916,47	1945,47	1868,83

Le cas du chinois (sous-tokénisation bpe)

hanlp

_德 累 斯 顿 _夜 生 活 _是 _丰 富 多 彩 _和 _激 动 _人 心 _的 _,
_所 以 _, _德 累 斯 顿 _被 _称 为 _是 _学 生 _俱 乐 部 _的 _首 府 _。

stanza

_德 累 斯 顿 _夜 生 活 _是 _丰 富 _多 彩 _和 _激 动 _人 心 _的 _,
_所 以 _, _德 累 斯 顿 _被 _称 为 _是 _学 生 _俱 乐 部 _的 _首 府 _。

jieba

_德 累 斯 顿 _夜 生 活 _是 _丰 富 多 彩 _和 _激 动 人 心 _的 _,
_所 以 _, _德 累 斯 顿 _被 _称 为 _是 _学 生 _俱 乐 部 _的 _首 府 _。

snownlp

_德 累 斯 顿 _夜 _生 活 _是 _丰 富 多 彩 _和 _激 动 人 心 _的 _,
_所 以 _, _德 累 斯 顿 _被 _称 为 _是 _学 生 _俱 乐 部 _的 _首 府 _。

pkuseg

_德 累 斯 顿 夜 _生 活 _是 _丰 富 多 彩 _和 _激 动 人 心 _的 _,
_所 以 _, _德 累 斯 顿 _被 _称 为 _是 _学 生 _俱 乐 部 _的 _首 府 _。

Les temps des tokénisations

	Time taken (pour 35,6 MB)
jieba	0:02:36.466316
pkuseg	0:13:48.343775
thulac	0:16:31.718059
snownlp	0:59:59.189956
stanford	0:41:54.229992
hanlp	1:47:26.123751

Un coût de reproductibilité assez élevé

- prise en main de plusieurs toolkits vs. interfaces en ligne
- une tâche pas si anodine : une personne par toolkit, un environnement CONDA spécifique pour gérer JOEYNMT et OPENNMT
- cohabitation de plusieurs toolkits : NEMATUS, OPENNMT, FAIRSEQ (conflit CUDA, pytorch)
- les tokenisations ne vont pas de soi
- les différences de paramétrage par défaut (seed non fixé dans openNMT)
- l'accessibilité de la documentation (notation plus subjective)
- la "facilité" de la prise en main variable
- le temps d'apprentissage : 22h35 avec JOEYNMT pour 100 époques vs. 18h avec OPENNMT pour 100 époques
- différences terminologiques pour la comptabilité différentes des époques (epoch/steps/iterations). Note : OpenNMT 2.2 ne semble plus donner les époques, mais uniquement les steps.
- espace disque : 4,5 Go de données pour un entraînement JOEYNMT/ 3 Go de données pour un entraînement OPENNMT

Reproductibilité et infrastructure

- options de sauvegardes complexifient la mesure de l'espace disque occupé par les entraînements (OPENNMT : 91 Mo, NEMATUS : 1,13 Go) (1 checkpoint)
- les différences de performances matérielles entre V100 et A100 se retrouvent...
- l'entraînement deux fois plus long avec HUMA-NUM (V100) qu'avec la A100 (serveur *Odette* UPCité)
  - 2 GPU A100 sur *Odette*, 87s, 1 000 steps (1 époque)
  - 4 GPU humanum NVIDIA Tesla V100-SW2-32Go, 101s, 1 000 steps (1 époque)

Reproductibilité et infrastructure

Où partager trois entraînements (10 Go) ?

- *github* limité à 50 Mo
- à suivre: corpus multiparallèles des traductions produites (avec scores associés)

Reproductibilité et choix des métriques

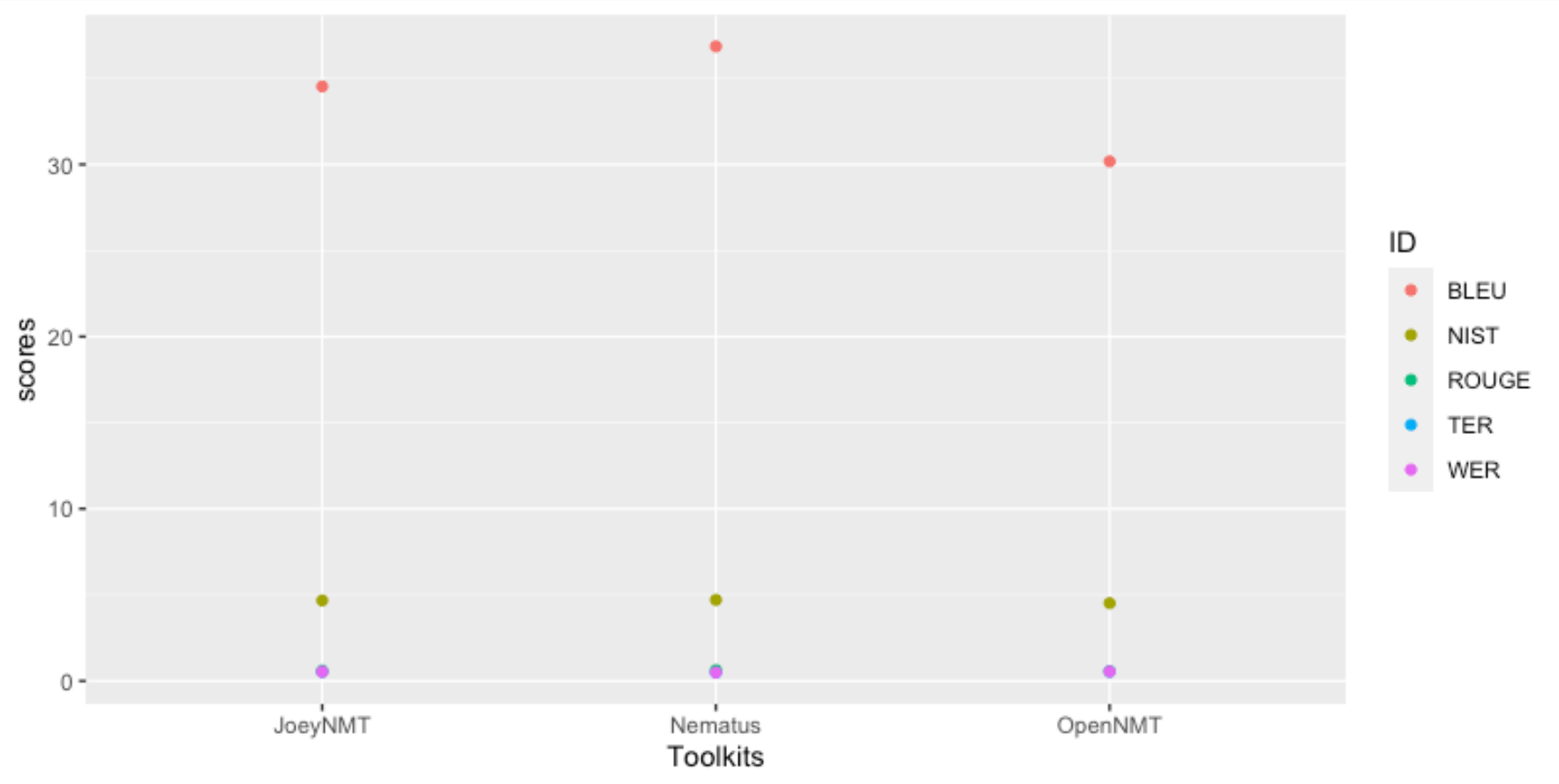


Figure 1: Différentes métriques d'évaluation des traductions JoeyNMT, Nematus et openNMT

Évaluation

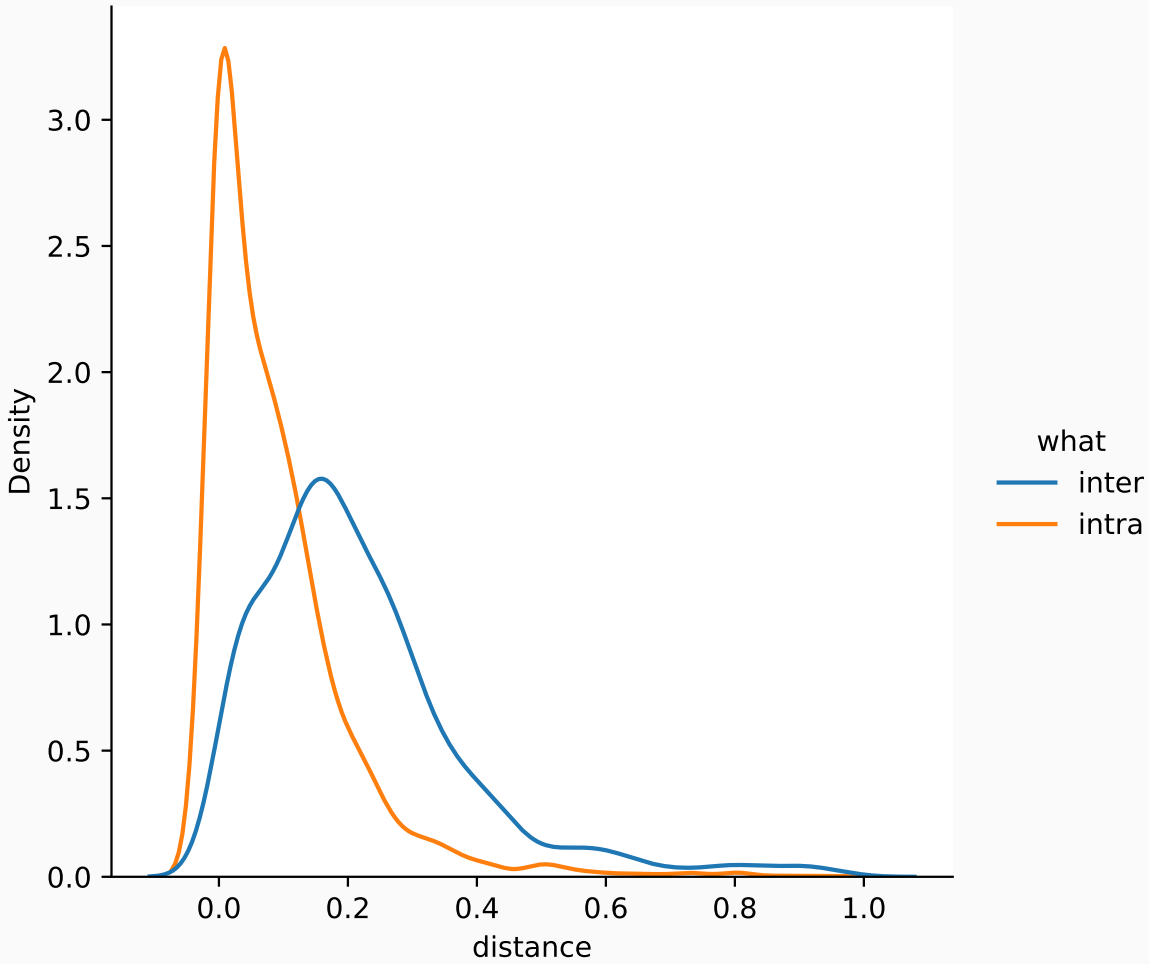


Figure 2: Distribution des distances d'édition intra-système (entre les deux entraînements de JoeyNMT) et inter-système (entre les prédictions de JoeyNMT et de OpenNMT

Discussion

Points de discussion

- Deux tokeniseurs différents sont utilisés (SentencePiece et Subword-NMT).
- Scripts de comparaison en cours pour la sous-tokenisation (pas les mêmes dictionnaires d'unités sous-tokenisées
- Les auteurs indiquent que les trois librairies possèdent un grand nombre d'hyperparamètres qu'il est possible de renseigner. Il serait intéressant d'en faire un tableau synthétique

Bilan

Trois niveaux d'évaluation de la reproductibilité ?

1. contrôler le hasard (les différences sont-elles significatives ?) (ici, un seul corpus d'entraînement),
2. autres approches pas encore faites : analyse textométrique *itrameur*,
3. et le point de vue du traducteur/ingénieur : dans quel toolkit investir ?

Conclusion

Conclusion

- paradoxe : des millions de paramètres mais très peu testés systématiquement : logique des solutions déjà adoptées et qui ”marchent” (cf. *”Attention is all your need” Vaswani et al. [2017]*) mais pas toujours testées
- le statut même de la métrique BLEU est en partie historique (et contes les données sur lesquelles sont fondées les justifications de Papineni et al. [2002] sont-elle (encore) pas disponibles? (voir travaux récents de Benjamin Marie sur SACREBLEU).

Remerciements

- Xinyi pour les tokenisations et analyses du chinois. Behnoosh pour les tokénisations et analyses du farsi.
- Ce travail a été partiellement financé par le projet NeuroViz / Explorations et visualisations d'un système de traduction neuronale, soutenu par la Région Île-de-France dans le cadre d' un financement DIM RFSI 2020 et par le projet SPECTRANS, dans le cadre de l'AAP émergence 2020 (ANR-18- IDEX-0001, Financement IdEx Université Paris Cité). Nous remercions le CNRS/TGIR HUMA-NUM et le Centre de Calcul IN2P3 pour la fourniture des ressources informatiques et de traitement des données nécessaires à ce travail.

Comparatif de quelques hyperparamètres

Feature	Sockeye	Neural Monkey	fair-seq	T2T	XNMT	OpenNMT-py	Joey NMT
Architecture							
RNN encoder	✓	✓	✓	✓	✓	✓	✓
RNN decoder	✓	✓	✓	✓	✓	✓	✓
Transformer encoder	✓	✓	✓	✓	✓	✓	✓
Transformer decoder	✓	✓	✓	✓	✓	✓	✓
ConvS2S encoder	✓	✓	✓			✓	
ConvS2S decoder	✓		✓			✓	
Image Encoder	✓	✓		✓		✓	
Audio Encoder		✓	✓	✓	✓	✓	
CTC		✓					
Attention Mechanisms	✓	✓	✓	✓	✓	✓	✓
Tasks							
Embedding Tying	✓		✓	✓	✓	✓	✓
Softmax Tying	✓	✓	✓	✓	✓	✓	✓
Parameter Freezing	✓		✓				✓
Multi-Source		✓	✓		✓		
Factored Input	✓	✓				✓	
Multi-Task		✓	✓		✓		
Sequence Labeling		✓					
Sequence Classification		✓		✓			
Language Modeling		✓	✓	✓	✓	✓	
Inference							
Segmentation Levels (word/char/bpe)	✓	✓	✓	✓	✓	✓	✓
Beam Search	✓	✓	✓	✓	✓	✓	✓
n-best outputs	✓	✓	✓				
Sampling	✓	✓	✓		✓	✓	
Rescoring	✓	✓			✓		
Checkpoint averaging	✓	✓	✓	✓	✓	✓	✓
Training							
MLE	✓	✓	✓	✓	✓	✓	✓
MRT		✓			✓		
Gradient Clipping	✓	✓	✓	✓	✓	✓	✓
Dropout	✓	✓	✓	✓	✓	✓	✓
Weight Decay	✓	✓	✓	✓	✓	✓	✓
Label Smoothing	✓	✓	✓	✓	✓	✓	✓
Optimizer	✓	✓	✓	✓	✓	✓	✓
Scheduler	✓	✓	✓	✓	✓	✓	✓
Early Stopping	✓	✓	✓	✓	✓	✓	✓
Usage							
CPU/GPU	✓	✓	✓	✓	✓	✓	✓
Monitoring	✓	✓	✓	✓	✓	✓	✓
Attention Visualization	✓	✓		✓	✓		✓

Table 6: Features implemented by popular NMT toolkits in Python as of July 1, 2019.

Figure 3: Comparatif des hyperparamètres dans Kreutzer et al. [2019]

References

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL *https://www.aclweb.org/anthology/P17-4012*.

J. Kreutzer, J. Bastings, and S. Riezler. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-3019. URL *https://aclanthology.org/D19-3019*.

T. Kudo and J. Richardson. 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, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL *https://aclanthology.org/D18-2012*.

B. Marie. Bleu: A misunderstood metric from another age - but still used today in ai research. URL *https://towardsdatascience.com/bleu-a-misunderstood-metric-from-another-age-d434e18f1b37*.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL *https://aclanthology.org/P02-1040*.

R. Sennrich, B. Haddow, and A. Birch. 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)*, pages 1715–1725, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL *https://aclanthology.org/P16-1162*.

R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. Miceli Barone, J. Mokry, and M. Nădejde. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain, Apr. 2017. Association for Computational Linguistics. URL *https://aclanthology.org/E17-3017*.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL *https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf*.