

# Evaluation du moteur de traduction neuronale OpenNMT sur un corpus parallèle anglais-français

WEN, Yu-Chieh  
INALCO, TAL Master1  
yuchieh.wen0621@gmail.com

May 30, 2024

## 1 Introduction

La traduction automatique neuronale transforme le texte d'une langue à une autre en utilisant des réseaux de neurones. Contrairement aux anciens systèmes basés sur des règles, la traduction automatique neuronale apprend des correspondances linguistiques à partir de grandes bases de données bilingues, offrant des traductions plus fluides et précises. Les modèles neuronaux, tels que les RNN, LSTM et Transformers, capturent les nuances contextuelles et grammaticales, améliorant continuellement la qualité des traductions malgré des défis persistants comme les langues rares et les subtilités culturelles.

## 2 Présentation du OpenNMT

OpenNMT(Neuron Machine Translation) est un moteur de traduction automatique neuronale open-source, développé par Harvard NLP et Systran. Il utilise des réseaux de neurones profonds pour traduire des textes, en s'appuyant initialement sur des RNN et des LSTM pour gérer les dépendances temporelles. Récemment, OpenNMT a intégré les Transformers, qui utilisent des mécanismes d'attention pour traiter les séquences en parallèle, offrant ainsi une meilleure efficacité et une traduction de haute qualité.

## 3 Evaluation du moteur de traduction neuronale

### OpenNMT sur un corpus en formes fléchies

#### 3.1 Corpus d'apprentissage et d'évaluation

Pour évaluer le moteur de traduction neuronale OpenNMT, j'ai utilisé deux principaux corpus parallèles : Europarl et EMEA. Ces corpus sont constitués de phrases alignées en anglais et en français.

- **Corpus d'apprentissage :**
  - **Europarl** : 100 000 premières phrases  
(`Europarl_train_100k.en`, `Europarl_train_100k.fr`).
  - **EMEA** : 10 000 phrases (`Ema_train_10k.en`, `Ema_train_10k.fr`).
- **Corpus de développement** : 3 750 phrases supplémentaires du corpus Europarl, de la phrase 100 001 à 103 750 (`Europarl_dev_3750.en`, `Europarl_dev_3750.fr`).
- **Corpus de test :**
  - **In-Domain** : 500 phrases du corpus Europarl, à partir de la phrase 103 751 (`Europarl_test_500.en`, `Europarl_test_500.fr`).
  - **Out-of-Domain** : 500 phrases du corpus EMEA, à partir de la phrase 10 001 (`Ema_test_500.en`, `Ema_test_500.fr`).

#### 3.2 Métriques d'évaluation : Score BLEU

Pour évaluer la qualité des traductions générées par le modèle OpenNMT, j'ai appliqué le score BLEU (Bilingual Evaluation Understudy). Le score BLEU mesure la similarité entre une traduction automatique et des traductions de référence humaines.

La formule du score BLEU peut être résumée comme suit :

$$\text{BLEU} = \text{BP} \cdot \exp \left( \sum_{n=1}^N w_n \log p_n \right)$$

où BP est la pénalité de longueur,  $w_n$  est le poids pour les n-grammes de taille  $n$  (souvent égal pour tous  $n$ ), et  $p_n$  est la précision modifiée des n-grammes de taille  $n$ .

##### Principe du Score BLEU :

- Le score BLEU compare des n-grammes de la traduction générée par la machine avec ceux des traductions de référence. Il calcule la précision des n-grammes et applique une pénalité de longueur pour éviter de favoriser les traductions trop courtes. Plus, les n-grammes de tailles 1 à 4 sont généralement utilisés.

### Calcul du Score BLEU avec sacreBLEU :

- J'ai utilisé l'outil sacreBLEU pour calculer les scores BLEU sur nos corpus de test, ici sont les commandes pour calculer les scores BLEU :

```
sacrebleu Europarl_test_500.fr -i pred_run1_in_domain.txt -m bleu
sacrebleu Emea_test_500.fr -i pred_run1_out_domain.txt -m bleu
sacrebleu Europarl_test_500.fr -i pred_run2_in_domain.txt -m bleu
sacrebleu Emea_test_500.fr -i pred_run2_out_domain.txt -m bleu
```

### 3.3 Tableau des résultats

Description	Run 1	Run 2
Apprentissage (nombre de phrases)	100K (Europarl)	100K+10K (Europarl+Emea)
Tuning (nombre de phrases)	3,75K (Europarl)	3,75K (Europarl)
BLEU Score (Set 1)	19.0	19.7
BLEU Score (Set 2)	0.2	73.1
Verbose Score (Set 1)	42.8/23.2/14.3/9.1	43.8/23.9/15.0/9.5
Verbose Score (Set 2)	5.6/2.2/0.0/0.0	88.1/78.9/68.8/62.3
BP (Set 1)	1.000	1.000
BP (Set 2)	1.000	0.990
Ratio (Set 1)	1.050	1.091
Ratio (Set 2)	1.974	0.990
Hyp Length (Set 1)	16628	17268
Hyp Length (Set 2)	5605	2810
Ref Length (Set 1)	15832	15832
Ref Length (Set 2)	2839	2839

Table 1: Résultats BLEU pour Run 1 et Run 2 sans lemmatisation

## 4 Evaluation du moteur de traduction neuronale OpenNMT sur un corpus en lemmes

### 4.1 Les lemmatiseurs

J'ai utilisé deux lemmatiseur, NLTK pour le corpus anglais et le FrenchLefffLemmatizer pour le français. Le lemmatiseur NLTK (Natural Language Toolkit) utilise le WordNetLemmatizer, basé sur la base de données lexicale WordNet. WordNetLemmatizer fonctionne en réduisant les mots à leur forme canonique ou racine en utilisant des règles

linguistiques et en tenant compte du contexte grammatical. Par exemple, il convertira "running" en "run" et "better" en "good".

Le FrenchLefffLemmatizer est un lemmatiseur spécialement conçu pour le français, intégré dans la bibliothèque TextBlob-fr. Il utilise le dictionnaire Lefff (Lexique des Formes Fléchies du Français), une ressource lexicale riche contenant des formes fléchies de mots français. Le FrenchLefffLemmatizer fonctionne en mappant chaque forme fléchie à sa forme de base ou lemme en se basant sur le dictionnaire Lefff, et il prend en compte les particularités grammaticales et morphologiques du français, comme les accords en genre et en nombre, les conjugaisons verbales, et les formes plurielles. Par exemple, il convertira "courant" en "courir" et "meilleures" en "meilleur".

## 4.2 Tableau des résultats

Description	Run 1	Run 2
Apprentissage (nombre de phrases)	100K (Europarl)	100K+10K (Europarl+Emea)
Tuning (nombre de phrases)	3,75K (Europarl)	3,75K (Europarl)
BLEU Score (Set 1)	18.6	19.9
BLEU Score (Set 2)	0.4	72.4
Verbose Score (Set 1)	43.6/22.7/14.0/8.7	45.2/24.5/15.1/9.4
Verbose Score (Set 2)	4.7/1.5/0.1/0.0	87.3/78.9/70.7/64.7
BP (Set 1)	1.000	1.000
BP (Set 2)	1.000	0.967
Ratio (Set 1)	1.073	1.021
Ratio (Set 2)	2.281	0.968
Hyp Length (Set 1)	16985	16160
Hyp Length (Set 2)	6476	2747
Ref Length (Set 1)	15832	15832
Ref Length (Set 2)	2839	2839

Table 2: Résultats BLEU pour Run 1 et Run 2 avec lemmatisation

## 4.3 Explication des Résultats

### • Résultat sans lemmatisation

- Set 1 : Le score BLEU est de 19.0, avec un ratio longueur hypothèse/référence de 1.050 et une longueur d'hypothèse de 16628 contre une longueur de référence de 15832. Les scores détaillés montrent que la traduction capture assez bien les n-grammes, surtout pour les grands n-grammes (42.8/23.2/14.3/9.1).
- Set 2 : Le score BLEU est extrêmement faible (0.2), avec un ratio de 1.974, suggérant

une hypothèse beaucoup plus longue que la référence. Les scores détaillés montrent une performance médiocre avec presque aucun n-gramme correctement traduit.

- **Résultat après lemmatisation**

- Set 1 : Le score BLEU baisse légèrement à 18.6. Le ratio augmente à 1.073, avec une hypothèse légèrement plus longue (16985). Les scores détaillés sont similaires (43.6/22.7/14.0/8.7), indiquant que la lemmatisation n'améliore pas significativement la traduction pour ce set.
- Set 2 : Le score BLEU double à 0.4, avec un ratio plus élevé de 2.281. La longueur de l'hypothèse augmente à 6476, montrant une hypothèse encore plus longue que la référence. Les scores détaillés montrent une légère amélioration mais restent très faibles (4.7/1.5/0.1/0.0).

## 4.4 Conclusion

La lemmatisation n'a pas significativement amélioré les résultats BLEU. Pour Set 1, le score BLEU a légèrement diminué, tandis que pour Set 2, bien que le score BLEU ait doublé, il reste très faible. Ces résultats suggèrent que d'autres méthodes, telles que des ajustements dans l'architecture du modèle ou des données d'entraînement plus diversifiées, pourraient être nécessaires pour améliorer les performances de traduction.

## 5 Points forts, limitations et difficultés rencontrées

J'ai dû utiliser le GPU payant de Google Colab pour l'entraînement, car l'utilisation de mon propre ordinateur était presque impossible. À chaque tentative d'exécution sur mon machine, un message d'erreur apparaissait, indiquant l'absence de CUDA, ce qui posait de nombreux problèmes.

Plus, une des limitations majeures est que nous devons nous contenter de corpus parallèles préentraînés. Par exemple, le corpus que nous avons utilisé pour l'entraînement n'était pas vraiment à jour, ce qui peut-être affecter la qualité des traductions.

### Points forts

- **Accès à des ressources de calcul puissantes** : Utiliser Google Colab nous permet d'accéder à des GPU puissants sans avoir à investir dans du matériel coûteux. Cela accélère considérablement le processus d'entraînement des modèles.
- **Facilité de configuration** : Google Colab permet l'utilisation de `shell magic`, ce qui simplifie la création de fichiers de configuration YAML et l'exécution de commandes shell directement dans le notebook..

- **Sauvegarde et gestion des versions :** Les notebooks de Colab peuvent être facilement sauvegardés et gérés via Google Drive, offrant ainsi une solution simple pour la gestion des versions et la conservation des travaux.

## Difficultés rencontrées

- **Problèmes de compatibilité logicielle :** Travailler sur mon machine sans NVIDIA CUDA installé a entraîné des erreurs constantes, rendant l'entraînement localement impossible et nécessitant une solution de contournement via des plateformes en ligne.
- **Qualité des données :** L'utilisation de corpus de données non à jour peut limiter les performances et la pertinence des modèles de traduction entraînés, soulignant l'importance de disposer de données récentes et de haute qualité.
- **Coût associé aux ressources :** Bien que Google Colab offre des GPU puissants, l'accès à ces ressources via un abonnement payant peut représenter un coût supplémentaire à considérer pour des projets de grande envergure ou à long terme.
- **Durée de l'entraînement :** L'entraînement du modèle prend environ une heure, ce qui est assez long et contraint de ne pas quitter le PC pendant ce temps en raison des limites de runtime de Google Colab. Cela peut être particulièrement contraignant et inefficace.

En résumé, bien que l'utilisation de Google Colab ait permis de surmonter les limitations matérielles et d'accélérer le processus d'entraînement, les défis liés à la qualité des données, aux coûts associés et aux contraintes de temps d'entraînement restent des facteurs importants à prendre en compte pour l'amélioration continue des modèles de traduction automatique neuronale.

## 6 Organisation

- **Github du projet :** [https://github.com/yuchieh0621mumu/Projet\\_TA\\_NMT](https://github.com/yuchieh0621mumu/Projet_TA_NMT)
- **Code du projet :** WEN, Yu-Chieh
- **Rédaction du rapport :** WEN, Yu-Chieh

## 7 Annexes

### 7.1 Expérimentations en plus

Pour approfondir mon analyse, j'ai réalisé plusieurs expérimentations avant le test du corpus à large échelle. En particulier, j'ai réalisé les expérimentations suivantes :

- **Corpora utilisés :**
  - **Train :** Europarl\_train\_10k.en et Europarl\_train\_10k.fr
  - **Dev :** Europarl\_dev\_1k.en et Europarl\_dev\_1k.fr
  - **Test :** Europarl\_test\_500.en et Europarl\_test\_500.fr
- **Évaluation :** J'ai évalué OpenNMT en utilisant le score BLEU sur le corpus de test.

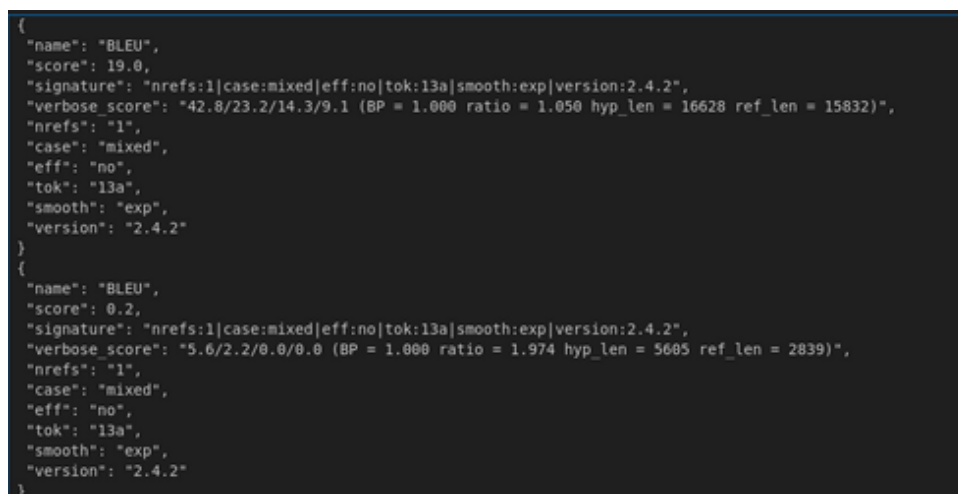
### 7.2 Corrections au niveau des corpus

Des corrections ont été apportées aux corpus de données pour améliorer la qualité des traductions.

- **Nettoyage des données :** Suppression des doublons en utilisant expression régulière, correction des erreurs typographiques et des incohérences linguistiques dans le corpus.

### 7.3 Captures d'écran des résultats

Voici quelques captures d'écran illustrant les résultats obtenus.



```
{
  "name": "BLEU",
  "score": 19.0,
  "signature": "nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.2",
  "verbose_score": "42.8/23.2/14.3/9.1 (BP = 1.000 ratio = 1.050 hyp_len = 16628 ref_len = 15832)",
  "nrefs": "1",
  "case": "mixed",
  "eff": "no",
  "tok": "13a",
  "smooth": "exp",
  "version": "2.4.2"
}
{
  "name": "BLEU",
  "score": 0.2,
  "signature": "nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.2",
  "verbose_score": "5.6/2.2/0.0/0.0 (BP = 1.000 ratio = 1.974 hyp_len = 5605 ref_len = 2839)",
  "nrefs": "1",
  "case": "mixed",
  "eff": "no",
  "tok": "13a",
  "smooth": "exp",
  "version": "2.4.2"
}
```

Figure 1: Capture d'écran des résultats de l'entraînement sans lemmatisation, run1.

```

{
  "name": "BLEU",
  "score": 19.7,
  "signature": "nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.2",
  "verbose_score": "43.8/23.9/15.0/9.5 (BP = 1.000 ratio = 1.091 hyp_len = 17268 ref_len = 15832)",
  "nrefs": "1",
  "case": "mixed",
  "eff": "no",
  "tok": "13a",
  "smooth": "exp",
  "version": "2.4.2"
}
{
  "name": "BLEU",
  "score": 73.1,
  "signature": "nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.2",
  "verbose_score": "88.1/78.9/68.8/62.3 (BP = 0.990 ratio = 0.990 hyp_len = 2810 ref_len = 2839)",
  "nrefs": "1",
  "case": "mixed",
  "eff": "no",
  "tok": "13a",
  "smooth": "exp",
  "version": "2.4.2"
}

```

Figure 2: Capture d'écran des résultats de l'entraînement sans lemmatisation, run2.

```

{
  "name": "BLEU",
  "score": 18.6,
  "signature": "nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.2",
  "verbose_score": "43.6/22.7/14.0/8.7 (BP = 1.000 ratio = 1.073 hyp_len = 16985 ref_len = 15832)",
  "nrefs": "1",
  "case": "mixed",
  "eff": "no",
  "tok": "13a",
  "smooth": "exp",
  "version": "2.4.2"
}
{
  "name": "BLEU",
  "score": 0.4,
  "signature": "nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.2",
  "verbose_score": "4.7/1.5/0.1/0.0 (BP = 1.000 ratio = 2.281 hyp_len = 6476 ref_len = 2839)",
  "nrefs": "1",
  "case": "mixed",
  "eff": "no",
  "tok": "13a",
  "smooth": "exp",
  "version": "2.4.2"
}

```

Figure 3: Capture d'écran des résultats de l'entraînement avec lemmatisation, run1.



```
{
  "name": "BLEU",
  "score": 19.9,
  "signature": "nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.2",
  "verbose_score": "45.2/24.5/15.1/9.4 (BP = 1.000 ratio = 1.021 hyp_len = 16160 ref_len = 15832)",
  "nrefs": "1",
  "case": "mixed",
  "eff": "no",
  "tok": "13a",
  "smooth": "exp",
  "version": "2.4.2"
}
{
  "name": "BLEU",
  "score": 72.4,
  "signature": "nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.2",
  "verbose_score": "87.3/78.9/70.7/64.7 (BP = 0.967 ratio = 0.968 hyp_len = 2747 ref_len = 2839)",
  "nrefs": "1",
  "case": "mixed",
  "eff": "no",
  "tok": "13a",
  "smooth": "exp",
  "version": "2.4.2"
}
```

Figure 4: Capture d'écran des résultats de l'entraînement avec lemmatisation, run2.