Projet de traduction neuronale OpenNMT

Analyse et évaluation du système de traduction neuroanle OpenNMT

Table des matières

I. Introduction

L'objectif de ce travail est d'analyser et évaluer le système de traduction neuronale OpenNMT pour la traduction de l'anglais vers le français. Nous souhaitons l'installer, l'évaluer et rédiger un rapport décrivant les principaux résultats, points forts et limitations de ce moteur de traduction neuronale.

II. Présentation du moteur de traduction neuronale OpenNMT

OpenNMT (Open Neural Machine Translation) est un système de traduction automatique basé sur le deep learning grâce à un réseau de neurones.

D'après la documentation, nous avons suivi scrupuleusement pas à pas les trois grandes étapes : la préparation des données, l'entraînement du modèle et la traduction.

Afin de fonctionner, OpenNMT utilise un corpus parallèle, soit un ensemble de phrases en langue source et leur traduction en langue cible. Il convient donc de constituer ce corpus parallèle et de le pré-traiter en le tokenisant, le normalisant et en limitant la longueur des phrases par exemple.

Une fois les données prêtes, il faut entraîner le modèle qui est basé sur des réseaux de neurones. Pour ce faire, des paires de phrases sources et cibles sont passées dans le modèle qui, pendant l'entraînement, apprend à associer correctement les sources et les cibles. Plus concrètement, les phrases sources sont converties en vecteurs multi-dimensionnels par un encodeur puis un décodeur génère la traduction grâce à ces vecteurs. En même temps, un mécanisme d'attention permet au modèle de se concentrer sur une certaine partie de la phrase à chaque étape du décodage.

Lorsque le modèle est entraîné, nous pouvons l'utiliser pour traduire de nouveaux textes jamais encore vus par le modèle. Le modèle prend en entrée une phrase en langue source, l'encode et génère une phrase en langue cible grâce au décodeur.

N'ayant pas spécifié d'architecture particulière pour notre modèle dans notre fichier .yaml, OpenNMT utilise par défaut une architecture en Transformer.

III. Evaluation du moteur de traduction neuronale OpenNMT sur un corpus en formes fléchies

Afin d'évaluer les traductions de l'anglais vers le français généré par OpenNMT, nous avons testé OpenNMT sur deux corpus parallèles pour le couple de langues anglais-français, un corpus avec des données du domaine et un autre hors-domaine.

Nous avons utilisé le corpus Europarl pour les textes du domaine. Europarl est un vaste corpus de textes multilingues issus des débats du Parlement européen, souvent utilisé pour entraîner et évaluer des systèmes de traduction automatique. Ce corpus est disponible en ligne et peut être trouvé à l'adresse suivante : https://opus.nlpl.eu/Europarl/corpus/version/Europarl.

Pour les textes hors-domaine, nous avons utilisé le corpus EMEA. EMEA est un corpus multilingue de textes médicaux provenant de l'Agence européenne des médicaments, souvent utilisé pour la traduction automatique dans le domaine médical. Ce corpus est disponible en ligne et peut être trouvé à l'adresse suivante : https://opus.nlpl.eu/EMEA/corpus/version/EMEA.

A. Constitution du corpus du domaine

Tableau 1: Tableau de description du corpus du domaine pour la run1

Corpus		Fichier	Nb phrases
Apprentissage	**Source**	*Europarl_train_10k.tok.true.clean.en*	10k
	Cible	*Europarl_train_10k.tok.true.clean.fr*	10k
Validation	**Source**	*Europarl_dev_1k.tok.true.clean.en*	1k
	Cible	*Europarl_dev_1k.tok.true.clean.fr*	1k
Test	**Source**	*Europarl_test_500.tok.true.clean.en*	500
	Cible	*Europarl_test_500.tok.true.clean.fr*	500

B. Constitution du corpus hors-domaine

Tableau 2: Tableau de description du corpus hors-domaine pour la run 2

Corpus		Fichier	Nb phrases
Apprentissage	**Source**	*Emea_train_1k.tok.true.clean.en*	1k
	Cible	*Emea_train_1k.tok.true.clean.fr*	1k
	Source	*Europarl_train_10k.tok.true.clean.en*	10k
	Cible	*Europarl_train_10k.tok.true.clean.fr*	10k

Validation	**Source**	-	-
	Cible	-	-
	Source	*Europarl_dev_1k.tok.true.clean.en*	1k
	Cible	*Europarl_dev_1k.tok.true.clean.fr*	1k
Test	**Source**	*Emea_test_50.tok.true.clean.en*	50
	Cible	*Emea_test_50.tok.true.clean.fr*	50
	Source	*Europarl_test_500.tok.true.clean.en*	500
	Cible	*Europarl_test_500.tok.true.clean.fr*	500

C. Description des métriques d'évaluation : Score BLEU

Le score BLEU (Bilingual Evaluation Understudy) est une méthode d'évaluation qui compare la traduction générée par une machine avec des traductions de référence humaines en mesurant la précision des n-grammes (séquences de n mots). Bien que le score bleu soit compris entre 0 et 1, il est souvent rapporté en pourcentage, 100% étant le meilleur score possible. Bien que les traductions automatique dont le score bleu dépasse les 50% sont considérées comme d'excellente qualité.

D. Résultats

Référence (nom fichier)	**Traduction Auto (nom fichier)**	**Modèle**	**BLEU (%)**
Europarl_test_500.tok.true.clean.fr	Europarl_pred_500_run1.tok.true.clean.fr	run1	10.84
Europarl_test_500.tok.true.clean.fr	Europarl_pred_500_run2.tok.true.clean.fr	run2	6.91
Emea_test_50.tok.true.clean.fr	Emea_pred_50_run2.tok.true.clean.fr	run2	3.51

On constate que le score BLEU diminue de près de moitié entre le run 1 et le run 2 lorsque l'on compare le fichier de référence au fichier traduit par le run 2. Cela suggère qu'OpenNMT performe mieux lorsqu'il est entraîné et évalué sur des corpus appartenant à un domaine spécifique. De même que, le faible score BLEU de la traduction d'Emea s'explique du fait que le modèle n'a pas été entraîné sur un corpus spécifique au médical mais sur un corpus mixte comprenant du médicale et du politique, ce qui réduit considérablement l'efficacité d'openNMT.

IV. Evaluation du moteur de traduction neuronale OpenNMT sur un corpus en lemmes

Afin de comparer nos résultats pour avoir une évaluation pertinente, nous avons aussi testé OpenNMT sur les mêmes corpus mais avec un pré-traitement différent : nous avons lemmatisé nos textes.

A. Constitution du corpus du domaine

Tableau 3: Tableau de description du corpus du domaine pour la run 3

Corpus		Fichier	Nb phrases
Apprentissage	**Source**	*Europarl_train_10k.lem.true.clean.en*	10k
	Cible	*Europarl_train_10k.lem.true.clean.fr*	10k
Validation	**Source**	*Europarl_dev_1k.lem.true.clean.en*	1k
	Cible	*Europarl_dev_1k.lem.true.clean.fr*	1k
Test	**Source**	*Europarl_test_500.lem.true.clean.en*	500
	Cible	*Europarl_test_500.lem.true.clean.fr*	500

B. Constitution du corpus hors-domaine

Tableau 4: Tableau de description du corpus hors-domaine pour la run 4

Corpus			Fichier	Nb phrases
Apprentissage		**Source**	*Emea_train_1k.lem.true.clean.en*	1k
		Cible	*Emea_train_1k.lem.true.clean.fr*	1k
		Source	*Europarl_train_10k.lem.true.clean.en*	10k
		Cible	*Europarl_train_10k.lem.true.clean.fr*	10k
Validation		**Source**	-	-
		Cible	-	-
		Source	*Europarl_dev_1k.lem.true.clean.en*	1k
		Cible	*Europarl_dev_1k.lem.true.clean.fr*	1k
Test		**Source**	*Emea_test_50.lem.true.clean.en*	50
		Cible	*Emea_test_50.lem.true.clean.fr*	50

		Source	*Europarl_test_500.lem.true.clean.en*	500
		Cible	*Europarl_test_500.lem.true.clean.fr*	500

C. Lemmatisation pour le français

Nous avons utilisé le modèle 'fr_core_news_sm' de la bibliothèque spacy, car nous n'avons pas été capable d'utiliser le module FrenchLefffLemmatizer comme préconisé dans la consigne. La fonction lemmatizer_spacy, transforme le texte en un document spaCy, puis extrait les lemmes de chaque token présent dans le texte. Spacy prend en compte le contexte pour extraire les lemmes, c'est pourquoi la lemmatisation reste assez fidèle au texte d'origine.

D. Lemmatisation pour l'anglais

Pour l'anglais, nous avons utilisé la bibliothèque nltk. La fonction de lemmatisation, lemmatizer_nltk, utilisée sur plusieurs ressources de NLTK, tels que WordNet pour les lemmes et les étiquettes de partie du discours (POS). Le texte est d'abord tokenisé en mots individuels, puis chaque token est étiqueté avec sa partie du discours. La fonction wordnet_pos convertit les étiquettes en un format compatible avec WordNet, ce qui permet une lemmatisation précise en prenant en compte le contexte syntaxique.

E. Résultats

Référence (nom fichier)	**Traduction Auto (nom fichier)**	**Modèle**	**BLEU (%)**
Europarl_test_500.tok.true.clean.fr	Europarl_pred_500_run3.lem.true.clean.fr	run3	3.95
Europarl_test_500.tok.true.clean.fr	Europarl_pred_500_run4.tok.true.clean.fr	run4	2.62
Emea_test_50.tok.true.clean.fr	Emea_pred_50_run4.tok.true.clean.fr	run4	0.40

Idem que pour les tests sur le corpus non Lemmatisé, on constate que le score BLEU diminue entre le run 3 et le run 4 lorsque l'on compare le fichier de référence au fichier traduit par le run 4. Cela suggère qu'OpenNMT performe mieux lorsqu'il est entraîné et évalué sur des corpus appartenant à un domaine unique. En effet, notre modèle ayant été entraîné sur une grande quantité de données Europarl, il semble évident qu'il performe mieux sur la traduction de textes parlementaires que sur la traduction de textes médicaux.

Ce qui est cependant surprenant est que la traduction soit de moins bonne qualité avec le corpus lemmatisé. Avant de commencer les tests, nous avions supposé que la lemmatisation améliorerait la traduction. Cela est peut-être dû au fait que nous n'ayons entraîné notre modèle que sur de petites quantités de données, ou à une erreur de notre part lors de la création du corpus, de sa tokenisation, lemmatisation ou des autres étapes de pré-traitement. Ou peut-être notre hypothèse était-elle simplement fausse.

Dans tous les cas, il serait intéressant dans un travail futur de comprendre pourquoi le modèle est moins performant après la lemmatisation du corpus.

V. Points forts, limitations et difficultés rencontrées

Notre toute première difficulté a été l'installation des dépendances et la configuration de OpenNMT.Une fois l'installation réussie, l'une de nos principales difficultés a été la limitation technique. En effet, OpenNMT nécessite un puissant GPU pour fonctionner et nos machines n'étant pas assez puissantes, l'entraînement des modèles prenait beaucoup trop de temps et faisait surchauffer nos machines. Ainsi, afin d'éviter de les abîmer et pour tout de même obtenir des résultats exploitables, nous avons donc fait le choix d'entraîner notre modèle sur un corpus plus petit. Nous verrons par la suite que notre traduction automatique est très mauvaise, certainement car le modèle n'a pas eu assez de données pour s'entraîner correctement.

De manière générale, comprendre les enjeux et objectifs du projet n'a pas été simple et nous a pris du temps.

Enfin, nous nous sommes jusqu'à la fin questionnés sur la forme que devait prendre notre git, son organisation, son contenu, et la meilleure manière de laisser une trace de notre travail sans mettre en ligne le corpus entier et les modèles entraînés qui sont très lourds, tout en montrant notre travail. Une difficulté a résidé dans le fait que la plupart de notre travail a été fait en ligne de commande et nous ne savions pas comment rendre compte de cela sans prendre de capture d'écran. C'est pourquoi nous avons opté pour des notebooks jupyter qui permettent de rendre compte de notre travail à chaque étape de son avancée. Cependant, nous avons ensuite douté sur la manière d'écrire le Readme puisque cela aurait été une simple répétition de ce qui était écrit sur nos notebooks.

Cependant, une fois ces difficultés surmontées, nous avons pu nous rendre compte que OpenNMT est un outil très facilement modulable que l'on peut personnaliser aisément selon ses besoins. Si nous n'avions pas été aussi limités techniquement, nous aurions pu tester de nombreuses configurations.

VI. Organisation

Puisque nous étions 2 dans notre groupe, nous avons tenté de nous répartir le travail à parts égales :

- Le dépôt Github du projet a été créé par Guilhem et Lucile l'a structuré
- Le preprocessing du corpus a été réalisé par Lucile tandis que Guilhem s'est chargé de mettre en place le code pour le score BLEU
- La lemmatisation a été effectuée par Guilhem tandis que Lucile préparait les notebooks permettant d'exécuter l'entraînement des modèles
- Chacun a rédigé la partie du rapport correspondant au travail qu'il avait effectué
- Idem pour le README dont chacun a rédigé les parties correspondant au travail qu'il avait effectué.

Pour conclure, le travail a été réparti de manière équitable entre les membres du groupe qui savait chacun quoi faire lorsque leur tâche était terminée. Le travail en équipe a été fluide et aucun des deux ne s'est senti lésé. À chaque étape de l'avancée du projet, un point a été fait pour que chacun explique à l'autre ce sur quoi il avait travaillé, afin que chacun comprenne l'entièreté des étapes du projet.