Deep Learning and Neural Machine Translation

Valentin Macé

Polytech – 2019

Deep Learning and Neural Machine Translation

Rapport de stage

Valentin Macé

Tuteur et Encadrant

M. Christophe Servan & M. Stéphane Ayache

Remerciements

Je tiens à remercier Christophe Servan pour m'avoir accueilli au sein de son équipe pendant ce stage et permis de travailler sur un sujet passionnant en m'accordant confiance et responsabilités.

Merci à toute l'équipe de Qwant Research pour sa bienveillance à mon égard, avec qui j'ai partagé de très bons moments tout au long de ces six mois.

Je remercie également Stéphane Ayache pour sa confiance, sa disponibilité et ses conseils depuis mon arrivée à Polytech.

Enfin j'aimerais remercier Bernard Espinasse qui m'avait transmis cette offre de stage.

Table des matières

1 Introduction

Dans ce rapport, j'utiliserai de nombreux termes anglais. Il serait difficile et peu pertinent d'effectuer une traduction littérale de ces mots largement acceptés et utilisés par la communauté scientifique.

1.1 Contexte

Ayant effectué mon stage de quatrième année au sein du Laboratoire d'Informatique et Systèmes de Marseille sur une problématique de deep learning appliqué au NLP[1] pour l'extraction et la classification d'entités nommées (Espinasse et al., 2019), je souhaitais poursuivre mon travail dans ce domaine à la frontière entre l'ingénierie et la recherche, qui couvre de nombreux aspects théoriques et techniques à l'instar de la formation d'ingénieur reçue à Polytech.

Le terme « Intelligence Artificielle », souvent repris dans les médias lorsqu'il s'agit d'évoquer les fulgurants progrès apportés par l'apprentissage profond ces dernières années, est loin de faire l'unanimité parmi les spécialistes du domaine. Il a cependant le mérite de nous renvoyer vers le but ultime de comprendre les mécanismes fondamentaux de l'intelligence afin de la recréer et de l'utiliser. La raison pour laquelle le NLP a une place particulière dans ce domaine tient au fait que le meilleur exemple d'intelligence à notre disposition (i.e. l'intelligence humaine) passe essentiellement par le langage naturel. C'est pourquoi, parmi tous les domaines d'application du deep learning, j'étais particulièrement intéressé par le NLP.

Pour des raisons évidentes, un moteur de recherche compétitif ne peut pas subsister sans la mise en place de techniques pour la compréhension du langage naturel. J'ai eu la chance d'être accepté par Qwant Research, filiale de Qwant dédiée à la recherche et l'ingénierie, pour ce stage de fin d'études. Il m'a été proposé d'y étudier et implémenter des approches de traduction automatique neuronale sur diverses paires de langues, d'entrainer de très nombreux modèles de traduction en bénéficiant d'une puissance de calcul compétitive permettant d'obtenir des résultats à l'état de l'art. Un des objectifs principaux de ce stage fut d'explorer le domaine émergent de la traduction neuronale au niveau du document, c'est-à-dire les méthodes permettant de prendre en compte le contexte autour d'une phrase afin de préserver une cohérence dans le choix des mots lors de l'inférence.

[1] Natural Language Processing, « Traitement Automatique du Language Naturel»

Avant de commencer le stage, mon expérience en apprentissage automatique consistait en une série de projets personnels[2], de cours reçus dans le cadre du cursus ingénieur informatique à Polytech Marseille et de différentes formations (MOOCs) disponibles sur internet. Mon stage de quatrième année constituait ma seule expérience professionnelle dans le domaine.

Mon rôle fut dans un premier temps de me familiariser avec l'environnement technique de l'entreprise, de me documenter et de me former sur les nombreux aspects liés à l'apprentissage profond pour la traduction. J'ai souvent échangé avec les chercheurs et ingénieurs de l'entreprise pour avancer dans ma compréhension du sujet et j'ai également pris du temps pour m'intéresser à d'autres aspects du deep learning n'étant pas directement liés à la traduction. Dans un second temps, j'ai pu mettre à profit les compétences acquises pour mener de nombreuses expériences sur lesquelles nous reviendrons.

1.2 Qwant

Qwant[3] est un moteur de recherche français (cf. Figure 1) dont l'existence est basée autour de deux axes principaux : le respect de la vie privée de ses utilisateurs et le besoin de souveraineté technologique au niveau français et européen. L'entreprise ne trace pas ses utilisateurs, elle ne récolte et ne vend pas leurs données personnelles afin de garantir leur vie privée et se veut neutre dans l'affichage des résultats. Aujourd'hui Qwant occupe une place stratégique pour l'indépendance de l'Europe sur le web et constitue une alternative au géant qu'est Google. Le moteur de recherche a été fondé en février 2013 par Jean-Manuel Rozan, Éric Leandri et Patrick Constant et compte aujourd'hui plus de 150 employés répartis à travers la France. Ses locaux sont situés à Nice, Ajaccio, Epinal, Rouen et Paris. C'est dans ces derniers, qui constituent le siège principal, que j'ai été accueilli pour effectuer mon stage.

L'entreprise présente une gamme de produits variée, le moteur de recherche principal se décline en plusieurs versions : Qwant Music est dédié au référencement d'artistes et titres musicaux, Qwant Junior facilite et filtre la recherche pour les enfants de 6 à 12 ans, Qwant School est son équivalent pour les adolescents et Qwant Causes permet de reverser une partie de l'argent des publicités à une œuvre caritative du choix de l'utilisateur. Durant mon stage, j'ai eu l'occasion d'assister à la mise en production de deux services particulièrement intéressants que sont Qwant Maps, équivalent de Google Maps sans récolte d'information sur les trajets effectués, et Qwant Masq qui permet aux utilisateurs de bénéficier de recommandations personnalisées sans qu'aucune information personnelle ne soit récoltée sur

[2] Exemple de projet personnel en apprentissage automatique : https://github.com/valentinmace/snake

[3] Site internet : www.qwant.com

les serveurs de l'entreprise. D'autres services plus classiques sont également proposés sur la page principale, comme la recherche d'images ou les actualités.

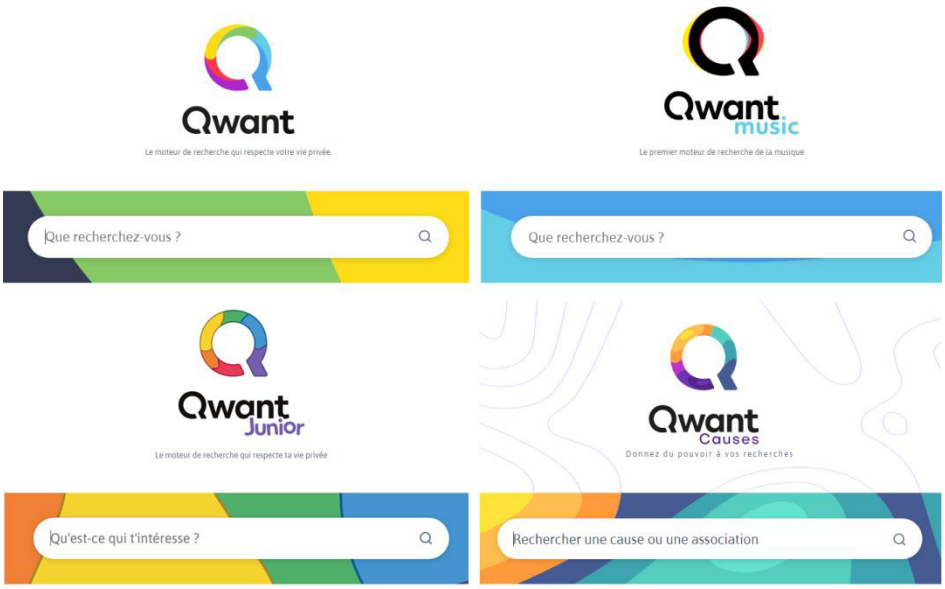

Figure 1- Le moteur de recherche et ses déclinaisons

La filiale recherche de Qwant, dans laquelle j'ai effectué mon stage, s'organise autour de cinq pôles : l'équipe NLP dont j'ai fait partie, le pôle Search en charge de l'amélioration du moteur de recherche principal, le pôle Vision & Image qui travaille à l'élaboration d'outils pour la compréhension automatique du contenu des images du web, l'équipe Maps et l'équipe Masq respectivement à l'origine de Qwant Maps et Qwant Masq. On note également la présence de Tristan Nitot, personnalité du monde des standards du web et fondateur de l'association Mozilla Europe, qui exerce la fonction de Vice-president Advocacy.

1.3 Mon ressenti

Je ne crois pas m'être ennuyé un seul jour dans cette entreprise qui vit au rythme des apparitions dans la presse française et internationale. Durant ces six mois, de nombreux acteurs politiques, médiatiques et même artistiques sont venus interroger les « spécialistes en intelligence artificielle » qui travaillent au troisième étage du bâtiment 7 Rue Spontini. Pour ne citer qu'eux : les chaînes Arte et M6 y ont tourné des reportages peu après mon arrivée, l'écrivain Marc Levy est venu chercher de l'inspiration au pôle NLP pour son prochain

roman, Cédric Villani a essayé la Tesla de Qwant que Gaël Musquet[4] a pris le temps de pirater pour en démontrer la facilité d'accès et la maire du 16ème arrondissement de Paris nous a rendu visite pour s'assurer que le passage à TensorFlow 2.0 se ferait sans encombre.

J'ai eu la chance d'être entouré pendant ces six mois de profils scientifiques, allant de l'ingénieur au docteur, avec qui je partage de nombreux centres d'intérêt. Il va sans dire que je n'ai rien contre les profils commerciaux mais il n'y a que dans ce genre d'environnement que l'on peut voir des gens s'affairer autour d'une vitre pour y dessiner schémas et équations à défaut d'avoir un autre support. Travailler pour un moteur de recherche qui n'en n'est qu'à ses débuts a également quelque chose d'attrayant pour un informaticien, la recherche d'information sur internet constitue un des points clés de la révolution engendrée par l'émergence de l'informatique et ce n'est pas sans rappeler l'histoire des géants américains du web.

Les activités organisées lors des pauses étaient très variées : jeux de plateau, tennis de table, jeux vidéo et même du golf improvisé. L'environnement à Qwant est plus que stimulant, les gens y sont très compétents et donnent envie de s'améliorer. De nombreuses petites conférences scientifiques sont organisées et permettent de découvrir les travaux des chercheurs invités. Le rythme de travail n'est pas imposé (dans la mesure du raisonnable), l'organisation est horizontale beaucoup plus que verticale et les responsables font preuve de flexibilité. Je garderai un très bon souvenir de cette expérience que je recommande aux futurs élèves stagiaires s'ils en ont l'opportunité.

[4] Gaël Musquet est un hacker français parmi les fondateurs de l'association OpenStreetMap France

2 Apprentissage profond et Langage naturel

Avant d'entrer plus en détail dans les activités du stage, il convient d'introduire les notions importantes qui reviendront fréquemment.

2.1 Deep Learning

L'apprentissage profond via réseaux de neurones artificiels ou « deep learning » est un ensemble de méthodes d'apprentissage automatique s'articulant autour de l'idée qu'un réseau constitué de structures atomiques relativement simples (les neurones artificiels) peut abstraire et modéliser des données complexes de manière efficace.

Ces méthodes de deep learning forment un sous-ensemble du machine learning, qui vient lui-même s'inscrire dans l'ensemble plus vaste de l'intelligence artificielle (cf. Figure 2). Il est essentiel de garder à l'esprit que même si l'apprentissage profond est aujourd'hui très en vogue, il n'est qu'un sous-domaine particulier des techniques d'apprentissage automatique et ne permet de traiter qu'un ensemble de problèmes restreint presque totalement disjoint de ceux abordés, entre autres, par les approches symboliques.

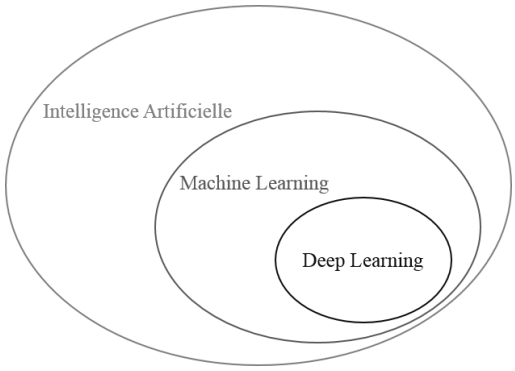

Figure 2 - La place du Deep Learning dans l'Intelligence Artificielle

La Figure 3 est une représentation schématique d'un réseau de neurones profond basique. On distingue de nombreux cercles représentant les neurones artificiels, reliés entre eux par des arcs formant des connexions. Ce réseau contient, de gauche à droite, une couche d'entrée, trois couches cachées et une couche de sortie présentant quatre neurones. Une telle structure peut être entrainée à effectuer n'importe quelle tâche de classification, le socle théorique de l'apprentissage profond est indépendant du domaine d'application mais n'a cependant pas la capacité d'engendrer des modèles transversaux, capables à la fois de discerner le contenu d'une image, de prédire le cours de la bourse et de traduire d'une langue vers une autre. Une

instance correspond généralement à une tâche. Le pouvoir de généralisation de ces modèles à des données encore jamais vues reste cependant très impressionnant et se trouve à l'origine de l'engouement pour ces techniques depuis le début de la décennie.

En pratique, les architectures neuronales ont beaucoup évolué, se sont complexifiées et spécialisées pour répondre au mieux à la tâche sur laquelle elles sont entrainées. Les systèmes portés sur la vision présentent des caractéristiques capables de capturer les invariances au sein d'une image, les systèmes de traduction sont construits de manière à pouvoir prêter plus ou moins d'attention aux différents mots d'une phrase. Cette tendance fait l'objet d'un intéressant débat entre Yann Le Cun[5], défendant que nous devrions restreindre l'implémentation de structures spécifiques découlant de notre intuition et Christopher Manning[6], qui se range du côté opposé.

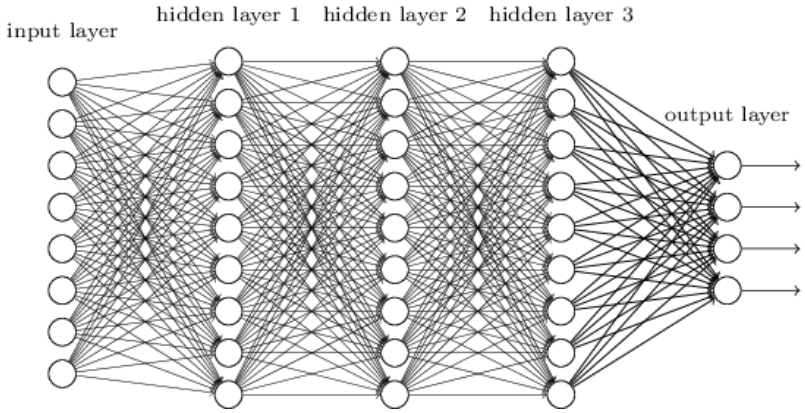

Figure 3 - Réseau de neurones profond

L'architecture d'un réseau de neurones profond, aussi complexe soit-elle, demeure une suite d'opérations simples pouvant être agrégées et modélisées sous la forme d'opérations matricielles afin d'être accélérées par des GPUs.

Pour illustrer ce fonctionnement, nous prendrons l'exemple des images de chiffres manuscrits en noir et blanc du MNIST[7]. Ces images ont une taille de 28x28 soit 784 pixels. On peut raisonnablement imaginer que l'on traduira chaque pixel en un nombre plus ou moins grand selon sa teinte. Pour une image donnée, chacun de ces 784 nombres va être fourni en entrée

[5] Yann Le cun est considéré comme l'un des inventeurs de l'apprentissage prodond

[6] Christopher Manning est un célèbre professeur à l'université de Stanford

[7] Base de données distribuée par Yann LeCun contenant 70 000 images de chiffres manuscrits

du réseau et se propager à travers ses nœuds (opérations). S'il a été préalablement bien entraîné, celui-ci devrait prédire avec une bonne probabilité de réussite le chiffre contenu dans l'image.

Il n'y a aucune raison a priori pour qu'un ensemble de neurones artificiels connectés entre eux donne de meilleurs résultats que du pur hasard. En effet, lorsqu'un réseau n'a pas encore été entrainé, ses prédictions sont aléatoires et ne présentent aucun intérêt. La force de ces structures réside dans le fait qu'elles présentent un grand nombre de paramètres pouvant être ajustés afin de converger vers un objectif.

Lorsque l'on commence à entrainer le modèle et que celui-ci se trompe, on peut mesurer son erreur et la quantifier. Si l'on donne une image du chiffre 4 à notre réseau et que celui-ci prédit 100% de chances que ce soit effectivement un 4 (donc 0% pour tous les autres chiffres), c'est une prédiction parfaite et l'erreur[8] mesurée est nulle. A contrario si la distribution de probabilités est différente, le réseau commet une erreur plus ou moins grande.

Le modèle est entrainé sur un grand nombre d'exemples, chaque fois l'erreur qu'il commet est mesurée et permet de réajuster ses connexions (paramètres) en les renforçant ou en les affaiblissant. Ces modifications, si elles sont faites intelligemment, vont changer sa prédiction pour tendre vers un meilleur résultat.

Ici l'intelligence réside dans les outils utilisés pour effectuer ces modifications, citons la méthode de rétro-propagation du gradient qui permet, grâce à l'utilisation des dérivées partielles, de comprendre dans quelle mesure chaque connexion est impliquée dans l'erreur du réseau. Elle est souvent couplée à l'algorithme de descente du gradient stochastique qui sert à effectuer les modifications en tenant compte de nombreux exemples d'entrainement (batchs) à la fois dans un temps de calcul raisonnable. Autrement dit, les paramètres du modèle ne sont pas modifiés à chaque exemple qui lui est soumis, on attend d'avoir accumulé suffisamment d'informations à travers différents exemples pour ajuster correctement les paramètres.

Si le parallèle entre apprentissage profond et cerveau humain est tentant, il est plus raisonnable de parler de « programmation différentiable[9] », paradigme dans lequel un programme paramétrique est automatiquement dérivé pour répondre de manière conforme à une tâche donnée. Car au final, les réseaux de neurones profonds ne sont ni plus ni moins que des programmes extrêmement complexes entrainés automatiquement, dont les paramètres sont lentement modifiés jusqu'à converger vers un état stable pour finalement demeurer statiques à la manière d'un système qui refroidit jusqu'à se figer.

---

[8] Les techniques pour mesurer l'erreur sont nombreuses et impactent fortement la qualité de l'entrainement

[9] Expression popularisée par Yan Le Cun : « Deep Learning est Mort! Vive Differentiable Programming »

2.2 NLP

Le traitement automatique du langage naturel est un domaine multidisciplinaire impliquant linguistique, informatique et apprentissage automatique. Ses applications sont nombreuses et couvrent différentes tâches, certaines d'entre elles sont directement utiles dans le monde réel (traduction, génération de texte) tandis que d'autres sont des sous-tâches servant de briques de base et pouvant être intégrées à des outils plus complexes.

2.2.1 Plusieurs niveaux de traitement

On distingue plusieurs niveaux de traitement du langage naturel. Les outils utilisés au niveau syntaxique permettent le plus souvent d'agir à l'échelle des mots, sur la manière dont ils sont agencés et sur leurs relations. Des tâches comme la tokénisation, l'étiquetage morphosyntaxique, le parsing de dépendances ou encore la lemmatisation sont très communes et faciles à mettre en place.

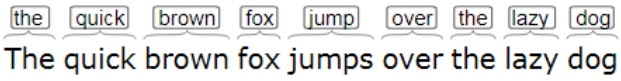

Figure 4 - Lemmatisation

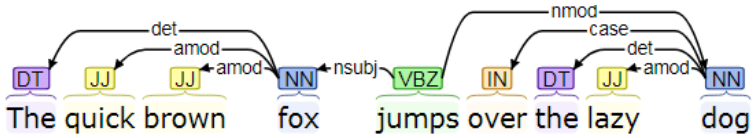

Figure 5 - Parsing de dépendances et étiquetage morphosyntaxique

La Figure 4 représente une phrase lemmatisée, où chaque mot est écrit dans sa forme la plus neutre. La Figure 5 représente une phrase avec son étiquetage morphosyntaxique ainsi que son chemin de dépendances.

Les outils utilisés au niveau sémantique permettent d'étudier la signification du langage, de ce que l'on veut énoncer. Les Figures 6 et 7 illustrent deux tâches de niveau sémantique. Dans la première, les deux entités nommées de la phrase sont reconnues et annotées, on parle de reconnaissance ou d'extraction d'entités nommées. Dans la seconde, la relation qui lie les deux entités est définie, on parle d'extraction de relation.

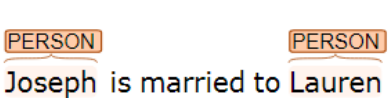

Figure 6 - Reconnaissance d'entités nommées

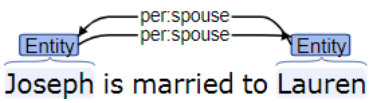

Figure 7 - Extraction de relation entre entités

2.2.2 Word Embedding et Modèle de langage

Le sens des mots est quelque chose qui échappe totalement à un ordinateur, qui les représente comme une suite d'octets arbitraire sans pouvoir en tirer d'information. Le mot « chat » est aussi proche de « félin » que de « tiroir » sous cette représentation et cela n'aide pas à réduire la complexité des tâches qui ont besoin d'un apport sémantique pour fonctionner.

Une première révolution dans le monde du NLP a eu lieu lors de la parution d'un algorithme (Mikolov et al., 2014) permettant de plonger un vocabulaire dans un espace vectoriel à haute dimension capturant la proximité sémantique entre les mots : le word embedding. Autrement dit, chaque mot est représenté sous la forme d'un vecteur à *N* dimensions et la distance entre ces vecteurs nous renseigne sur la similitude entre les mots : « chat » est proche de « félin » mais éloigné de « tiroir ». Cette méthode révolutionnaire capture également la relation d'analogie entre les mots d'un vocabulaire, le vecteur « man » est au vecteur « woman » ce que le vecteur « king » est au vecteur « queen » (cf. Figure 8).

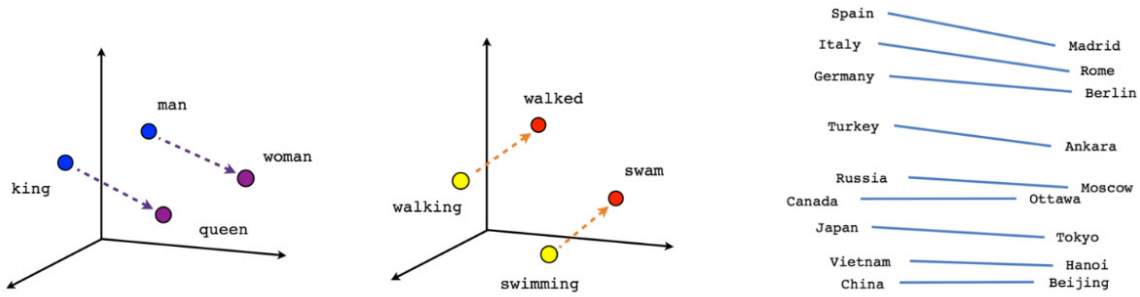

Figure 8 - Word Embedding

L'idée centrale derrière cette prouesse repose sur le concept de modèle de langage, concept qui mènera à une seconde petite révolution très récente dans le domaine du NLP, nous y reviendrons.

Un modèle de langage est un outil statistique permettant d'assigner une probabilité à une séquence de mots. En d'autres termes, c'est un système qui, étant donné une phrase, retourne la probabilité que celle-ci soit réellement employée.

Sous sa forme la plus simple, un modèle de langage peut être défini de la façon suivante :

$$P(w_1, \dots, w_m) = \prod_{i=1}^{m} P(w_i)$$

Équation 1 - Modèle de langage unigram

Ici le modèle, appelé unigram, est simpliste et permet d'assigner une probabilité à la séquence de mots en considérant la probabilité d'apparition de chaque mot pris indépendamment.

Des modèles plus complexes permettent de tenir compte du contexte dans ce calcul, ce sont les modèles dits n-gam. Dans un modèle n-gram, la probabilité d'observer une phrase est approximée par l'Equation 2, pour chaque mot de la séquence, le modèle considère n-1 mots de contexte

$$P(w_1, \dots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \dots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)}, \dots, w_{i-1})$$

Équation 2 - Modèle de langage n-gram

Les modèles de langage peuvent être construits de différentes manières, allant du simple décompte de mots dans un corpus de documents, à l'entrainement d'un modèle neuronal prenant en entrée les n-1 mots de contexte et donnant en sortie une distribution de probabilité sur un vocabulaire entier.

Quoi de mieux pour définir un mot que de connaitre parfaitement les différents contextes dans lesquels il est susceptible d'être employé ? C'est l'idée qui a permis à Thomas Mikolov et son équipe d'aboutir au word embedding. En se basant sur un modèle de langage neuronal simple, ils ont imaginé qu'étant donné un contexte de quelques mots, un réseau de neurones entrainé à prédire le mot manquant à partir de ce contexte capturerait dans sa couche cachée une grande partie de sa signification.

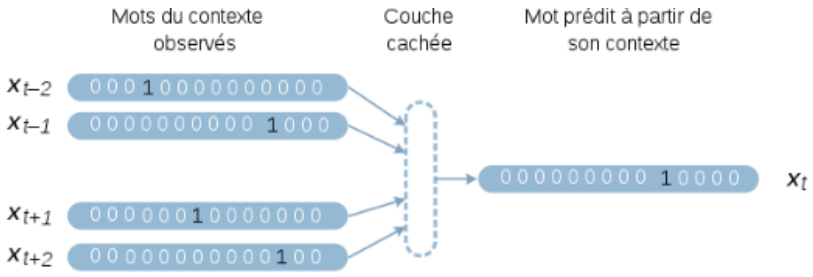

Figure 9 - Word2Vec

La Figure 9 illustre le fonctionnement de cet algorithme, appelé Word2Vec. Il nécessite bien sûr d'être entrainé sur de très grands corpus, mais présente l'avantage d'être quasi non-supervisé. Les vecteurs des mots sont obtenus en récupérant la couche cachée dont on peut choisir la dimension lors de l'entrainement, typiquement celle-ci varie entre 200 et 4000 selon les applications. Cette avancée a considérablement impacté tous les domaines du NLP, bénéficiant d'une représentation légère et puissante du vocabulaire, et constitue un domaine de recherche très actif.

2.2.3 Traduction automatique neuronale

La traduction automatique désigne la traduction d'un texte entièrement réalisée par un programme, sans qu'un traducteur humain n'ait à intervenir. Une avancée récente majeure dans ce domaine est l'adoption des réseaux de neurones profonds permettant une prise en compte plus efficace du contexte et des similarités lexicales. Ces modèles sont entrainés sur de très gros corpus de texte parallèles, c'est-à-dire où chaque phrase dans la langue cible est alignée avec son équivalent dans la langue source. La tâche porte le nom de traduction automatique neuronale, plus connue sous l'appellation anglaise « Neural Machine Translation » (NMT).

Les architectures neuronales pour la NMT sont très variées, elles présentent généralement deux éléments principaux : l'encodeur, chargé de transformer une phrase donnée en une représentation vectorielle compacte et le décodeur, qui transforme à son tour cette représentation en une phrase cible. La Figure 10 représente schématiquement ce type de modèle.

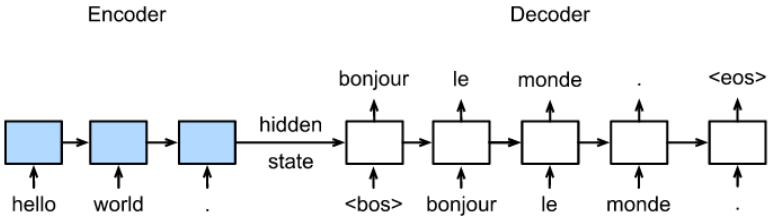

Figure 10 - Encodeur - Décodeur architecture

Historiquement ce sont les LSTMs[10] qui ont le plus longtemps été utilisés pour la traduction car ils présentent la capacité de traiter des séquences de données très efficacement en bénéficiant d'une sorte de mémoire sous la forme de connexions récurrentes. Plus récemment, des architectures novatrices basées sur des mécanismes dits « d'attention » ont pris l'ascendant et sont à l'origine d'une révolution beaucoup plus large dans le monde du NLP. Nous étudierons le modèle référence du Transformer (Vaswani et al., 2017) un peu plus tard dans ce rapport.

[10] Long short-term memory, dérivé des réseaux de neurones récurrents (RNNs)

3 Mon travail

Mon travail chez Qwant m'a permis de découvrir de nombreux aspects autour de l'apprentissage profond et du NLP. Si ma mission était centrée sur la traduction automatique, j'ai eu l'occasion de progresser dans de nombreux autres domaines techniques et théoriques.

3.1 L'environnement

L'environnement technique de Qwant est similaire à celui de nombreuses entreprises. Parmi les outils utilisés en interne pour la communication et le travail en équipe, on retrouve entre autres Jira pour le suivi des tickets et des projets, Slack pour la communication facilitée sous forme de channels, Confluence pour l'organisation du travail d'équipe et Gitlab pour le partage de code. Le VPN de l'entreprise permet aux employés de travailler en dehors du lieu de travail tout en garantissant la sécurité des échanges.

3.1.1 Les ressources de calcul

L'apprentissage profond est réputé pour être très gourmand en ressources de calcul. J'ai eu la chance pendant ces six mois de bénéficier d'un DGX, supercalculateur de chez NVIDIA comportant 8 cartes graphiques Tesla V100[11], un processeur avec 80 cœurs et 512Go de mémoire vive, pour moi tout seul. L'accès au DGX se faisait à distance en SSH, ce qui m'a permis de jouir d'une grande flexibilité durant toute la durée du stage, pouvant travailler à l'extérieur des bureaux tout en bénéficiant de cette puissance de calcul. De telles ressources permettent à la fois de grandement accélérer les nombreuses expériences mises en place mais surtout de pouvoir entrainer des modèles neuronaux à l'état de l'art, qui ne pourraient pas tenir sur des configurations moins performantes. Nous verrons que le nombre de GPUs n'influe pas seulement sur la vitesse d'entrainement, mais également sur les performances finales d'un modèle.

Les droits d'accès au DGX sont restreints et il n'est pas permis d'y exécuter n'importe quoi. La plupart du temps ces machines sont partagées entre les membres d'une équipe et il ne faut pas qu'une personne puisse « casser » le travail des autres. Pour pouvoir agir librement, j'ai été initié à l'utilisation de conteneurs qui permettent d'isoler un environnement virtuel dans lequel je pouvais faire mes expériences. Grâce à la technologie NVIDIA-Docker, il me suffisait de créer des instances d'environnements virtuels à partir de fichiers descriptifs (DockerFiles) pour travailler à l'intérieur sans aucune perte de performance. Cette

[11] Le coût actuel d'une de ces cartes est d'environ 9000€

conteneurisation m'a également permis de facilement gérer le versioning des bibliothèques utilisées sans forcément passer par des outils de gestion des environnements de développement lorsque ce n'était pas nécessaire.

La Figure 11 illustre les couches logicielles et matérielles du DGX. Hormis son prix et sa puissance de calcul, celui-ci est un serveur comme un autre doté de 8 cartes graphiques. Au-dessus de cette couche matérielle se trouve le système d'exploitation sur lequel est installé le driver CUDA, qui permet la pleine exploitation des GPUs. Cette couche CUDA sert à effectuer des calculs génériques sur les cartes graphiques qui sont très utiles lorsqu'il s'agit de paralléliser massivement les nombreuses opérations matricielles inhérentes à l'apprentissage profond. Le Docker Engine se situe au-dessus de cette couche et permet de lancer les différents conteneurs.

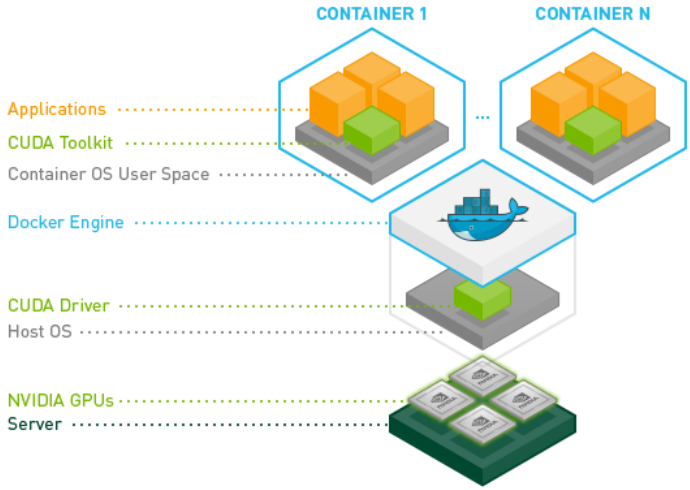

Figure 11 - Architecture du DGX

3.1.2 Les outils utilisés

Durant mon stage j'ai été amené à utiliser de nombreux outils que je ne peux pas citer de manière exhaustive. J'ai découvert la puissance des commandes Bash pour le traitement de datasets contenant des dizaines de millions d'exemples, utilisé de nombreuses bibliothèques Python pour le filtrage de corpus, la parallélisation de scripts ou encore l'extraction et la modification de paramètres dans les modèles de traduction neuronaux, et j'ai également automatisé beaucoup de tâches répétitives.

J'emporte avec moi quelques scripts et commandes génériques qui me seront utiles à l'avenir, mais surtout une approche plus efficace lorsqu'il s'agit de traiter de problèmes en rapport avec le NLP et plus généralement l'apprentissage automatique.

Il y a cependant deux outils qui m'ont suivi tout au long du stage et qui nécessitent d'être détaillés.

TensorFlow

TensorFlow est un outil d'apprentissage automatique open source développé par Google. Il permet à l'utilisateur de créer et d'entrainer des architectures de réseaux de neurones arbitrairement complexes et se veut très optimisé. L'idée sous-jacente au fonctionnement de TensorFlow ainsi qu'à d'autres frameworks similaires consiste à définir un graphe de calcul différentiable constitué d'opérations et de variables (paramètres) qui seront ajustées lors de l'entrainement.

Pour prendre l'exemple le plus simple, supposons que l'on souhaite approximer un nuage de points du plan à l'aide d'un modèle de régression linéaire. Dans ce cas, nous formulons l'hypothèse que le jeu de données (nuages de points) peut être représenté par une droite d'équation $z = Ax + b$ et nous cherchons à prédire les valeurs z en fonction de x. On a donc des données d'entrainement sous la forme d'un ensemble de tuples (x, z). Il est très facile d'utiliser TensorFlow pour construire le graphe de calcul de cette équation comme le montre schématiquement la Figure 12.

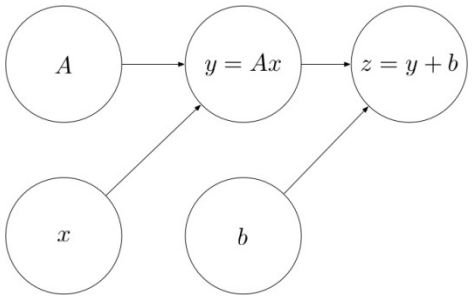

Figure 12 - Graphe de calcul d'une droite

Les paramètres de ce modèle sont les variables que l'on peut ajuster pour modifier la droite et la rendre plus descriptive du jeu de données, ici ce sont A et b. L'utilisation de TensorFlow permet de faciliter et d'accélérer cet ajustement, pour chaque x donné en entrée, le modèle de régression linéaire va effectuer une prédiction quant à sa position sur l'axe z et très probablement se tromper. Cette erreur va être mesurée et rétro-propagée pour identifier la part de responsabilité de chacun des paramètres dans l'erreur commise. Grâce aux gradients calculés automatiquement, le framework va ajuster ces paramètres pour réduire l'erreur et répéter l'action sur les nombreux exemples d'apprentissage.

Tensorflow permet l'automatisation du calcul des gradients qui nous renseignent sur la manière d'ajuster les paramètres. Il nous permet donc de construire des architectures très complexes sans avoir à dériver à la main tous les paramètres du modèle. Un réseau de neurones artificiel n'étant finalement qu'un agencement ingénieux de variables et d'opérations simples, il est naturel d'en faciliter l'entrainement avec ce type d'outil.

La version GPU de TensorFlow permet de paralléliser toutes ces opérations et de les exécuter sur des cartes graphiques beaucoup plus rapides que des processeurs. On notera aussi l'existence de nombreuse extensions à TensorFlow dont l'outil de visualisation TensorBoard (cf. Figure 13), que j'ai beaucoup utilisé durant ce stage, qui permet de visualiser en direct l'entrainement des modèles.

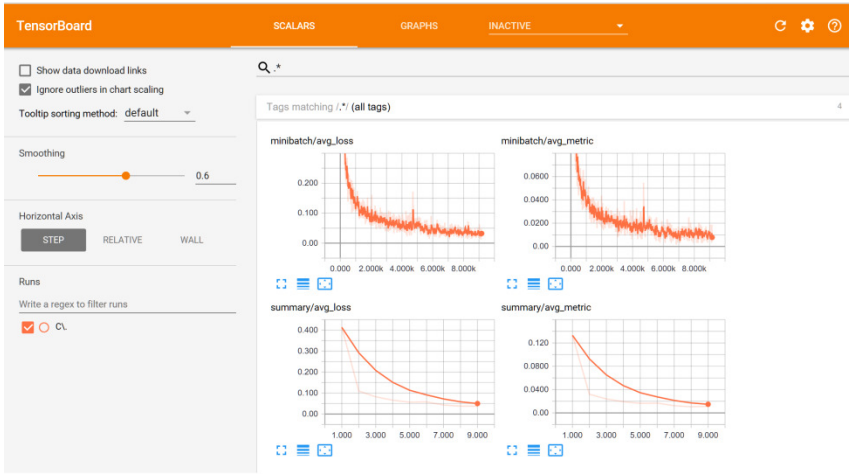

Figure 13 - TensorBoard

OpenNMT

OpenNMT (Klein et al., 2017) est un outil open source pour la traduction automatique qui se veut à la fois flexible et robuste, susceptible d'être utilisé pour la recherche et l'industrie. La bibliothèque se décline en trois versions : OpenNMT-py construite sur le framework PyTorch, OpenNMT-lua et OpenNMT-tf basée Tensorflow. C'est cette dernière que j'ai utilisée durant mon stage, tous les systèmes que j'ai pu entrainer et tester ont été conçus grâce à cet outil qui permet également de facilement déployer les modèles sous forme de serveurs d'inférence.

Le framework se veut généraliste et capable d'aborder un ensemble de problèmes plus vaste que la seule traduction. Il est notamment conçu pour la classification de séquences, la création de modèles de langages ou encore le sequence tagging. OpenNMT a constitué un formidable exemple de bonnes pratiques à mettre en place lorsque l'on construit par-dessus TensorFlow, cependant sa taille et sa complexité m'ont coûté de nombreuses heures de reverse engineering pour arriver à mes fins et m'ont parfois bloqué sur des tâches assez simples.

3.2 Le Transformer

Le Transformer (Vaswani et al., 2017) est le modèle de traduction automatique neuronale que j'ai utilisé tout au long du stage. Il constitue la plus grande avancée pour la traduction ces dernières années et les implications qui en découlent ont eu un impact important sur tous les domaines du NLP.

Son architecture est relativement simple et basée sur un mécanisme lui permettant, contrairement aux précédents modèles de types récurrents, d'être hautement parallélisable. Il est de ce fait plus rapide lors de l'entrainement et de l'inférence.

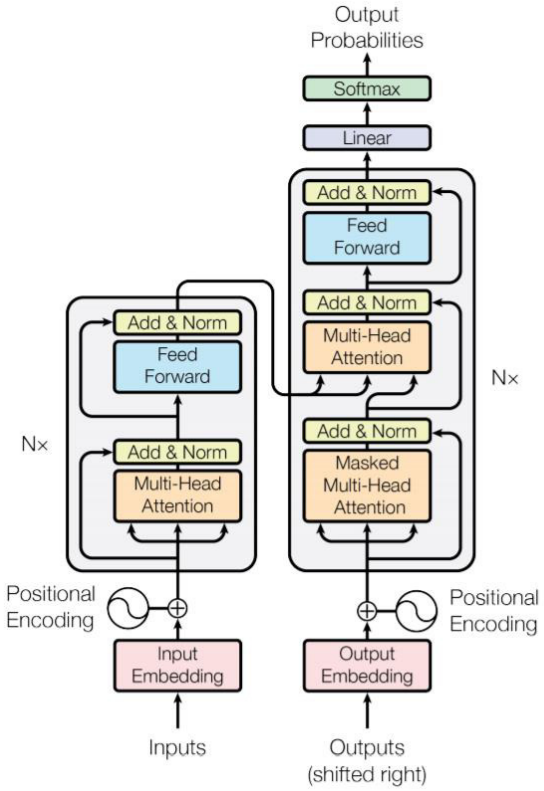

Figure 14 - Architecture du Transformer

La Figure 14 détaille l'architecture du Transformer. Comme ses prédécesseurs, le modèle s'articule autour d'un encodeur, qui transforme une phrase en entrée vers une représentation abstraite, et d'un décodeur qui se charge de traduire cette représentation abstraite en phrase cible.

L'encodeur est constitué d'une pile de couches identiques, chacune de ces couches est divisée en deux parties principales. La première est un mécanisme d'attention élaboré permettant au

modèle d'accorder plus ou moins d'importance aux différents mots d'une phrase pendant l'encodage, c'est ce mécanisme qui est en charge de prendre en compte le contexte de la phrase lors de la traduction. La deuxième partie est un couche de neurones classiques entièrement connectés.

Prenons la phrase « *The animal didn't cross the street because it was too tired* » comme exemple. Pour avoir une représentation pertinente du mot « *it* », un modèle de traduction devrait savoir qu'il correspond à « *The animal* ». C'est précisément ce que permet de faire le mécanisme d'attention, comme le montre la Figure 15, le Transformer prête de l'*attention* au sujet lorsqu'il s'agit d'encoder le mot « *it* ».

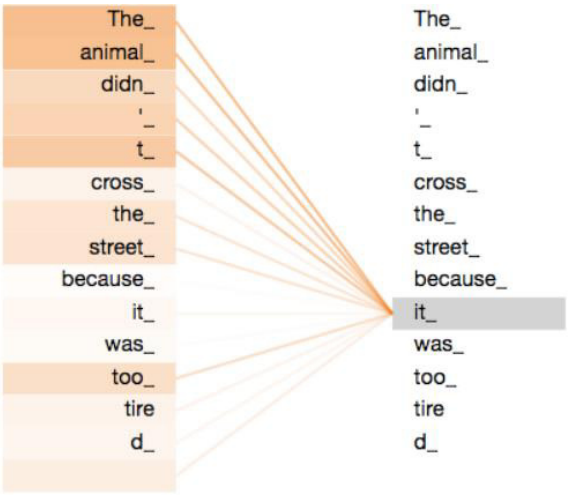

Figure 15 - Visualisation du mécanisme d'attention

Le décodeur du Transformer est constitué d'une pile de couches similaires à celles de l'encodeur, les mécanismes d'attention du décodeur permettent à la fois de prendre en compte la sortie de l'encodeur mais également les mots précédemment traduits.

Contrairement aux modèles de type RNN, le Transformer n'accumule pas l'information de manière séquentielle jusqu'à obtenir une représentation compacte d'une phrase. Le modèle construit cette représentation en considérant tous les mots à la fois, laissant le mécanisme d'attention capturer les interdépendances entre les mots. Il est de ce fait plus facilement parallélisable, et opère beaucoup plus rapidement que les autres architectures séquentielles.

Cette architecture, intuitivement proche des efforts qu'un humain doit mettre en œuvre lors d'une traduction, est à l'origine de progrès dans tous les domaines du NLP. Des modèles de

langage[12] basés sur le Transformer ont permis d'obtenir un socle performant pouvant être adapté à toutes les tâches de traitement du langage naturel et l'impact de cette avancée pourrait être comparable avec l'apparition d'ImageNet dans le domaine de la vision.[13]

3.3 Travaux et Expériences

L'objectif de ce stage était pour moi de mener des travaux afin d'améliorer les modèles de traduction qui seront susceptibles d'être utilisés par Qwant, aussi bien en interne que dans le cadre d'un éventuel service de traduction. Même si ces pratiques sont susceptibles d'évoluer très vite dans les mois à venir, nous avons réussi, avec Christophe Servan mon responsable de stage, à mettre en évidence des résultats intéressants dont l'utilité persistera. J'ai également eu l'occasion de participer activement à des discussions autour de la traduction automatique pour y partager certains résultats et recueillir des conseils.

Dans cette section je résumerai les différentes expériences que nous avons pu mener, et je finirai par inclure l'article scientifique que nous avons soumis au workshop IWSLT 2019, focalisé sur la traduction neuronale automatique au niveau du document, c'est-à-dire la tâche qui consiste à tenir compte du contexte inter-phrases lors de la traduction.

Toutes ces expériences, sauf contre-indication, ont été menées en considérant le corpus de données parallèles de WMT 2019[14] pour la traduction de l'Anglais vers l'Allemand. Les données de test utilisées pour évaluer les modèles correspondent à celles fournies par WMT pour la tâche de traduction de news.

3.3.1 Mesure des performances pour la traduction

Dans ce rapport j'utiliserai principalement deux métriques pour l'évaluation de la performance des modèles de traduction.

Le score BLEU est un algorithme d'évaluation de la qualité d'une traduction. Ici la qualité est considérée comme la correspondance entre la production d'une machine et celle d'un humain : plus une traduction automatique est proche d'une traduction humaine professionnelle, plus le score sera élevé. Les notes sont calculées pour chaque segment traduit, généralement des phrases, en les comparant avec une traduction de référence. La moyenne de ces notes est

[12] github.com/google-research/bert

[13] ruder.io/nlp-imagenet/

[14] www.statmt.org/wmt19/

ensuite calculée sur l'ensemble du corpus pour obtenir une estimation de la qualité globale de la traduction. L'intelligibilité ou l'exactitude grammaticale ne sont pas prises en compte.

Le score TER mesure la quantité de modifications nécessaire qu'un traducteur devrait effectuer pour qu'une traduction automatique corresponde exactement à la traduction de référence. Logiquement, un TER élevé signifie qu'il est nécessaire d'apporter beaucoup de modifications et donc qu'une traduction est mauvaise. A l'inverse, un TER bas est signe d'une bonne traduction.

3.3.2 Exploration des hyper paramètres

Si il y a bien une tâche commune à tous les domaines d'application en apprentissage automatique, c'est certainement celle qui consiste à jouer avec les paramètres structurels du modèle pour en trouver la meilleure configuration.

Les hyper paramètres (par opposition aux paramètres) sont ceux qui permettent de modifier la structure du modèle. On peut par exemple décider qu'une couche donnée comportera deux fois plus de neurones, que la dimension des vecteurs de mots sera moitié moins grande ou encore que le nombre d'exemples utilisés simultanément pour l'apprentissage sera plus important.

Avec une architecture comme celle du Transformer, le choix des hyper paramètres n'est pas anodin. Les auteurs de l'article montrent qu'un modèle plus gros, avec par exemple une dimension interne de 1024 au lieu de 512 et un nombre de têtes d'attention doublé, augmente significativement les performances de traduction. Ce qu'ils ne disent pas en revanche, c'est si il est possible d'obtenir des résultats similaires en considérant un modèle plus petit mais entrainé plus longtemps. Nous avons mené des expériences pour comparer deux architectures de type Transformer, l'une étant identique au modèle de base proposé dans l'article (nous l'appellerons *Base*), l'autre reprenant les hyper paramètres du système plus large (nous l'appellerons *Big*).

La différence majeure entre les deux systèmes réside dans le fait que le modèle de *Base* est entrainé sur un seul GPU tandis que le modèle *Big* est entrainé sur 8 GPUs. Cette répartition du plus gros modèle sur les cartes graphiques se fait de la manière suivante : la section principale du graphe de calcul du modèle est répliquée sur les différents GPUs et les batchs (ensembles d'exemples d'entrainement) sont traités en parallèle. Cette méthode équivaut à multiplier la taille des batchs d'entrainement par le nombre de cartes graphiques utilisées, c'est-à-dire que lorsque le modèle s'entraine, il considère un nombre plus grand d'exemples à chaque étape.

Les deux modèles ont d'abord été entrainés pendant le même nombre d'itérations[15], sans grande surprise le modèle *Big* a largement surpassé le modèle *Base.* Nous avons ensuite poursuivi l'entrainement du modèle *Base* jusqu'à atteindre le triple du nombre d'itérations précédemment utilisé.

Modèle	Itérations	newstest2017		newstest2018	
		BLEU	TER	BLEU	TER
Transformer *Base*	1.5M	27.75	53.7	41.98	39.76
Transformer *Big*	500k	**29.75**	**51.5**	**45.8**	**36.56**

Tableau 1 - Résultats des expériences sur les hyper paramètres

Le Tableau 1 décrit les résultats de cette expérience. On constate que la configuration *Big* dépasse largement les performances du modèle de *Base* ayant bénéficié d'un nombre d'itérations d'entrainement 3 fois supérieur. Ces résultats se confirment sur deux corpus d'évaluation, en utilisant deux métriques différentes. Cette différence s'explique probablement par l'augmentation du nombre de GPUs utilisés lors de l'entrainement, impliquant une multiplication de la taille des batchs d'entrainement. Nous faisons l'hypothèse que des ensembles d'exemples plus grands permettent au modèle d'accumuler des gradients plus précis et plus pertinents pour l'ajustement des paramètres.

3.3.3 Subword units

Une des difficultés les plus importantes en traduction automatique réside dans la nécessité pour un modèle de savoir traduire des mots d'un vocabulaire très large d'une langue source, vers d'autres mots d'un vocabulaire très large d'une langue cible.

Nous l'avons vu, le word embedding permet d'attribuer à chaque mot un vecteur capturant sa signification. Il est cependant difficile de s'assurer que chaque mot d'un vocabulaire très riche sera connu du modèle, qui lui attribuera un vecteur à haute dimension. Non seulement cela demande de stocker une quantité trop grande de données en mémoire, mais cela demande

[15] On utilisera le nombre d'itérations comme mesure pour la durée d'entrainement des modèles, notons qu'une itération ne prend pas le même temps pour s'effectuer sur les différents systèmes

également de plus grandes ressources de calcul lorsque le modèle doit effectuer sa prédiction et choisir un mot dans un vocabulaire cible très large.

Historiquement, les approches de traduction automatique se contentaient d'un vocabulaire réduit, forçant les modèles à traduire les mots inconnus en un token par défaut. Depuis 2016, de nouvelles approches de segmentation des mots (Sennrich et al., 2016) ont permis de résoudre ces problèmes de manière très efficace, augmentant significativement la qualité de la traduction.

L'idée derrière ces méthodes est de découper les mots en de plus petites unités, appelées subword units. Ces subword units agissent comme des briques de base et peuvent être agrégées par un modèle pour former des mots plus longs. Elles permettent ainsi de découper les mots inconnus pour en traduire les sous-unités unes à unes.

Cette approche implique tout d'abord la construction d'un modèle de segmentation pour la langue source et la langue cible. On entraine ce modèle sur les corpus à notre disposition et c'est lui qui décidera automatiquement de la manière de les segmenter. On applique ensuite cette segmentation aux corpus parallèles qui nous serviront pour l'entrainement et l'évaluation des systèmes de traduction.

Obama receives Net@@ any@@ ahu

the relationship between Obama and Net@@ any@@ ahu is not exactly friendly . the two wanted to talk about the implementation of the international agreement and about Teheran 's destabil@@ ising activities in the Middle East . the meeting was also planned to cover the conflict with the Palestinians and the disputed two state solution . relations between Obama and Net@@ any@@ ahu have been stra@@ ined for years . Washington critic@@ ises the continuous building of settlements in Israel and acc@@ uses Net@@ any@@ ahu of a lack of initiative in the peace process . the relationship between the two has further deteriorated because of the deal that Obama negotiated on Iran 's atomic programme . in March , at the invitation of the Republic@@ ans , Net@@ any@@ ahu made a controversial speech to the US Congress , which was partly seen as an aff@@ ront to Obama . the speech had not been agreed with Obama , who had rejected a meeting with reference to the election that was at that time im@@ pending in Israel .

Figure 16 - Exemple de texte segmenté

La Figure 16 présente un texte anglais segmenté en subword units, on remarque que la plupart des mots n'ont pas été segmentés car suffisamment fréquents pour être contenus dans le vocabulaire en un seul morceau. En prêtant attention aux segments choisis par le modèle, on constate que certains d'entre eux font sens pour l'humain tandis que d'autres sont assez originaux.

Nous avons mené des expériences pour tester différents outils implémentant cette approche. Comme toujours, nous comparons les résultats avec un modèle témoin (noté *Baseline)* qui ne

bénéficie pas de cette amélioration. Toutes les expériences sont effectuées avec un Transformer dans sa version de base tournant sur un seul GPU.

Le Tableau 2 présente les résultats des expériences sur la segmentation de mots. Tous les modèles s'entrainent sur des corpus qui ont été préalablement tokénisés, c'est-à-dire où les mots ont été clairement séparés d'autres éléments comme la ponctuation. Ce détail a son importance car même si certains outils sont destinés à être utilisés sur des corpus non tokénisés, nous avons remarqué que cette étape permet quand même d'améliorer les performances des modèles.

On remarque que l'outil le plus performant disponible actuellement pour la génération de subword units est *SentencePiece*[16]. Il est une amélioration de *Subword-nmt* proposée par Google, qui permet d'obtenir des résultats largement supérieurs à ceux du modèle *Baseline*.

Outil de segmentation	newstest2017		newstest2018	
	BLEU	TER	BLEU	TER
Baseline	23.59	57.31	35.35	44.1
Morfessor	23.42	60.33	35.77	47.6
Subword-nmt	26.41	55.03	39.34	41.88
SentencePiece	**27.75**	**53.7**	**41.98**	**39.76**

Tableau 2 - Résultats des expériences sur les subword units

J'ai eu l'occasion de partager ces résultats avec des membres de la communauté d'OpenNMT, qui ont confirmé la pertinence de *SentencePiece* pour cette tâche.

3.3.4 Back-translation et ajout de bruit

Comme dans beaucoup de tâches en apprentissage profond, la quantité de données disponible pour entrainer les modèles de traduction automatique a un impact important sur leurs performances.

La difficulté de créer des datasets d'entrainement pour la traduction réside dans le fait que ceux-ci doivent présenter des phrases alignées dans les deux langues. Chaque phrase dans la langue source doit être minutieusement alignée à la phrase correspondante dans la langue cible, une proportion raisonnable de phrases mal traduites ne pose pas forcément problème mais un seul décalage dans le corpus détruit complètement cet alignement et le rend obsolète.

[16] github.com/google/sentencepiece

Les jeux de données parallèles sont donc relativement restreints par rapport à la quantité presque infinie de corpus monolingues.

Il n'est cependant pas impossible de tirer profit de ces données monolingues. La back-translation est une technique d'augmentation des corpus d'entrainement parallèles avec des données synthétiques. L'idée est simple : si l'on veut entrainer un modèle de traduction qui va de la langue *A* vers la langue *B*, on possède normalement un corpus de phrases parallèles et il est très facile de trouver des données monolingues dans la langue *B*. On peut utiliser un système de traduction inverse qui va de la langue *B* vers la langue *A* pour traduire ces données monolingues et ainsi obtenir de nouvelles données parallèles qui seront ajoutées au corpus d'entrainement.

La Figure 17 illustre le processus de back-translation tel que décrit précédemment.

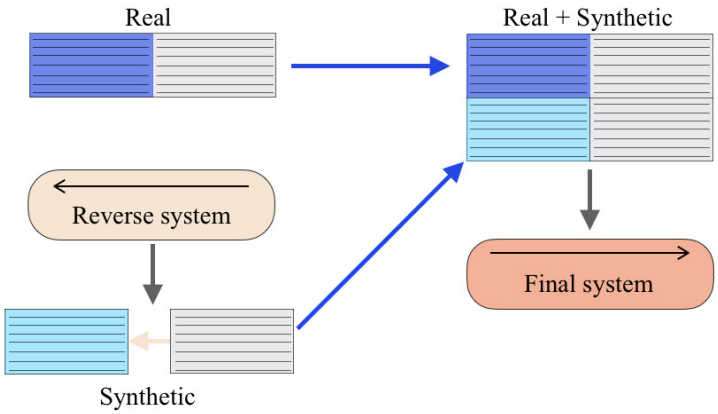

Figure 17 - Back-translation

Cette technique fait plusieurs hypothèses fortes, elle nécessite l'existence d'un système de traduction déjà entrainé qui va de la langue *B* vers la langue *A*. Les performances de ce système impacteront nécessairement la qualité des données synthétiques, qui peuvent s'avérer inutiles pour l'entrainement du modèle qui va de *A* vers *B*. La création du corpus synthétique a un coût important en temps et en ressources de calcul, selon la taille du corpus monolingue, sa traduction peut être aussi chronophage que l'entrainement du modèle lui-même.

Les travaux à l'état de l'art en traduction automatique considèrent cependant une étape de plus dans la constitution du corpus synthétique. Edunov et al. (2018) montrent qu'ajouter du bruit à ces données permet de considérablement améliorer le signal d'entrainement qu'elles offrent. En d'autres termes les données synthétiques, intrinsèquement imparfaites, sont modifiées et altérées de manière à les rendre plus efficaces pour l'entrainement des modèles de traduction.

Ces modifications sont très simples, elles consistent par exemple à changer l'ordre des mots d'une phrase, à supprimer certains de ses mots ou encore à les remplacer par un token par défaut. Il apparait que l'ajout de ce bruit aux données synthétiques permet de casser la trop grande régularité avec laquelle elles sont obtenues et ainsi d'améliorer le signal d'entrainement.

Durant mon stage j'ai eu l'occasion d'implémenter[17] et de tester cette idée. Le Tableau 3 présente les résultats obtenus pour les expériences sur la back-translation et l'ajout de bruit. Tous les modèles ayant bénéficié d'un corpus augmenté par des données synthétiques obtiennent de meilleurs résultats que le modèle *Baseline*. Le système qui présente les meilleurs scores est le *Back-translation+Noise (2)* qui a été entrainé sur un corpus présentant 50% de données synthétiques bruitées.

Modèle	Proportion of Synthetic data	newstest2017		newstest2018	
		BLEU	TER	BLEU	TER
Baseline	0%	25.77	55.94	39.58	41.98
Back-translation	50%	27.06	54.69	40.06	41.46
Back-translation+Noise (1)	25%	27.35	54.21	40.07	41.55
Back-translation+Noise (2)	50%	**27.88**	**53.88**	**41.92**	**39.88**

Tableau 3- Résultats des expériences sur la back-translation et l'ajout de bruit

Comprendre l'apport du bruit dans l'entrainement des modèles neuronaux m'a également permis d'interpréter des résultats qui jusque là me paraissaient surprenants. Les corpus d'entrainement pour la paire de langue Anglais-Allemand sont conséquents et contiennent de nombreuses phrases n'appartenant à aucune des deux langues. Mes premières expériences furent de filtrer les données pour éliminer ces phrases et d'entrainer des modèles pour mesurer l'effet de ce prétraitement. A chaque fois j'étais surpris de constater que les modèles s'entrainant sur des corpus non filtrés obtenaient de meilleurs scores, ce qui peut s'expliquer par l'apport en bruit des phrases indésirables.

3.3.5 Agrégation des améliorations dans un modèle final

Toutes les expériences évoquées, ainsi que de nombreuses autres m'ont permis d'améliorer peu à peu la qualité des systèmes de traduction. Durant mon dernier mois de stage, j'ai fini par

[17] github.com/valentinmace/noisy-text

implémenter un modèle bénéficiant de ces améliorations qui obtient des performances à l'état de l'art.

Le Tableau 4 présente les résultats obtenus par le modèle final (noté *Qwant Research*), il surpasse légèrement le score obtenu par le modèle de traduction de l'entreprise *DeepL* et performe moins bien que celui proposé par Edunov et al. (2018). Ce dernier, presque identique au notre, a été entrainé sur 128 GPUs Tesla V100 alors que nous en utilisons 8.

Modèle	newstest2014 BLEU
DeepL	33.3
Qwant Research	33.41
Edunov et al. (2018)	**34.46**

Tableau 4 - Résultats du model final et comparaison à l'état de l'art

3.3.6 Traduction au niveau du document

Les expériences menées sur la traduction au niveau du document constituent le sujet principal de ce stage, elles font l'objet d'une publication scientifique que j'ai rédigée en collaboration avec Christophe Servan. L'article a été soumis au workshop IWSLT 2019 et se trouve en annexe.

4 Conclusion

Ce stage de fin d'études chez Qwant constitue ma plus belle expérience professionnelle, j'ai atteint les objectifs que je m'étais fixés en arrivant et j'espère que ma contribution au projet porté par Eric Léandri et ses équipes pourra être utile. J'espère également avoir l'occasion de retrouver un environnement de travail aussi stimulant et bienveillant, je souhaite à toute l'équipe de l'étage Research ainsi qu'aux autres membres de l'entreprise une excellente continuation.

Mon objectif à présent est de trouver un sujet de thèse et de continuer à explorer le domaine fascinant de l'apprentissage automatique. Ces trois années passées à Polytech Marseille m'ont permis d'avancer dans cette direction, ce pourquoi je remercie tous les professeurs et professeures qui m'ont eu en tant qu'élève.

Using Whole Document Context in Neural Machine Translation

Valentin Mace, Christophe Servan

valentin.mace@etu.univ-amu.fr
c.servan@qwant.com

Abstract

In Machine Translation, considering the document as a whole can help to resolve ambiguities and inconsistencies. In this paper, we propose a simple yet promising approach to add contextual information in Neural Machine Translation. We present a method to add source context that capture the whole document with accurate boundaries, taking every word into account. We provide this additional information to a Transformer model and study the impact of our method on three language pairs. The proposed approach obtains promising results in the English-German, English-French and French-English document-level translation tasks. We observe interesting cross-sentential behaviors where the model learns to use document-level information to improve translation coherence.

1. Introduction

Neural machine translation (NMT) has grown rapidly in the past years [1, 2]. It usually takes the form of an encoder-decoder neural network architecture in which source sentences are summarized into a vector representation by the encoder and are then decoded into target sentences by the decoder. NMT has outperformed conventional statistical machine translation (SMT) by a significant margin over the past years, benefiting from gating and attention techniques. Various models have been proposed based on different architectures such as RNN [1], CNN [3] and Transformer [2], the latter having achieved state-of-the-art performances while significantly reducing training time. However, by considering sentence pairs separately and ignoring broader context, these models suffer from the lack of valuable contextual information, sometimes leading to inconsistency in a translated document. Adding document-level context helps to improve translation of context-dependent parts. Previous study [4] showed that such context gives substantial improvement in the handling of discourse phenomena like lexical disambiguation or co-reference resolution.

Most document-level NMT approaches focus on adding contextual information by taking into account a set of sentences surrounding the current pair [5, 6, 7, 8, 9, 10]. While giving significant improvement over the context-agnostic versions, none of these studies consider the whole document with well delimited boundaries. The majority of these approaches also rely on structural modification of the NMT model [7, 8, 9, 10]. To the best of our knowledge, there is no existing work considering whole documents without structural modifications.

Contribution: We propose a preliminary study of a generic approach allowing any model to benefit from document-level information while translating sentence pairs. The core idea is to augment source data by adding document information to each sentence of a source corpus. This document information corresponds to the belonging document of a sentence and is computed prior to training, it takes every document word into account. Our approach focuses on pre-processing and consider whole documents as long as they have defined boundaries. We conduct experiments using the Transformer base model [2]. For the English-German language pair we use the full WMT 2019 parallel dataset. For the English-French language pair we use a restricted dataset containing the full TED corpus from MUST-C [11] and sampled sentences from WMT 2019 dataset. We obtain important improvements over the baseline and present evidences that this approach helps to resolve cross-sentence ambiguities.

2. Related Work

Interest in considering the whole document instead of a set of sentences preceding the current pair lies in the necessity for a human translator to account for broader context in order to keep a coherent translation. The idea of representing and using documents for a model is interesting, since the model could benefit from information located before or after the current processed sentence.

Previous work on document-level SMT started with cache based approaches, [12] suggest a conjunction of dynamic, static and topic-centered cache. More recent work tend to focus on strategies to capture context at the encoder level. Authors of [6] propose an auxiliary context source with a RNN dedicated to encode contextual information in addition to a warm-start of encoder and decoder states. They obtain significant gains over the baseline. A first extension to attention-based neural architectures is proposed by [7], they add an encoder devoted to capture the preceding source sentence. Authors of [8] introduce a hierarchical attention network to model contextual information from previous sentences. Here the attention allows dynamic access to the context by focusing on different sentences and words. They show significant improvements over a strong NMT baseline. More recently,

SOURCE	*TARGET*
<doc1> Pauli is a theoretical physicist	Pauli est un physicien théoricien
<doc1> He received the Nobel Prize	Il a reçu le Prix Nobel
<doc2> Bees are found on every continent	On trouve des abeilles sur tous les continents
<doc2> They feed on nectar using their tongue	Elles se nourrissent de nectar avec leur langue
<doc2> The smallest bee is the dwarf bee	La plus petite abeille est l'abeille naine

Table 1: Example of augmented parallel data used to train the *Document* model. The source corpus contains document tags while the target corpus remains unchanged.

[10] extend Transformer architecture with an additional encoder to capture context and selectively merge sentence and context representations. They focus on co-reference resolution and obtain improvements in overall performances.

The closest approach to ours is presented by [5], they simply concatenate the previous source sentence to the one being translated. While they do not make any structural modification to the model, their method still does not take the whole document into account.

3. Approach

We propose to use the simplest method to estimate document embeddings. The approach is called SWEM-aver (Simple Word Embedding Model – average) [13]. The embedding of a document k is computed by taking the average of all its N word vectors (see Eq. 1) and therefore has the same dimension. Out of vocabulary words are ignored.

$$Doc_k = \frac{1}{N} \sum_{i=1}^{N} w_{i,k} \tag{1}$$

Despite being straightforward, our approach raises the need of already computed word vectors to keep consistency between word and document embeddings. Otherwise, fine-tuning embeddings as the model is training would shift them in a way that totally wipes off the connection between document and word vectors.

To address this problem, we adopt the following approach: First, we train a baseline Transformer model (noted *Baseline* model) from which we extract word embeddings. Then, we estimate document embeddings using the SWEM-aver method and train an enhanced model (noted *Document* model) benefiting from these document embeddings and the extracted word embeddings. During training, the *Document* model does not fine-tune its embeddings to preserve the relation between words and document vectors. It should be noted that we could directly use word embeddings extracted from another model such as Word2Vec [14], in practice we obtain better results when we get these vectors from a Transformer model. In our case, we simply extract them from the *Baseline* after it has been trained.

Using domain adaptation ideas [15, 16, 17], we associate a tag to each sentence of the source corpus, which represents the document information. This tag takes the form of an additional token placed at the first position in the sentence and corresponds to the belonging document of the sentence (see Table 1). The model considers the tag as an additional word and replace it with the corresponding document embedding. The *Baseline* model trains on a standard corpus that does not contain document tags.

The proposed approach requires strong hypotheses about train and test data. The first downfall is the need for well defined document boundaries that allow to mark each sentence with its document tag. The second major downfall is the need to compute an embedding vector for each new document fed in the model, adding a preprocessing step before inference time.

4. Experiments

We consider two different models for each language pair: the *Baseline* and the *Document* model. We evaluate them on 3 test sets and report BLEU and TER scores. All experiments are run 8 times with different seeds, we report averaged results and p-values for each experiment. Translation tasks are English to German, proposed in the first document-level translation task at WMT 2019 [18], English to French and French to English, following the IWSLT translation task [19].

4.1. Training and test sets

Table 2 describes the data used for the English-German language pair. These corpora correspond to the WMT 2019 document-level translation task. Table 3 describes corpora for the English-French language pair, the same data is used for both translation directions.

For the English-German pair, only 10.4% (3.638M lines) of training data contains document boundaries. For English-French pair, we restricted the total amount of training data in order to keep 16.1% (602K lines) of document delimited corpora. To achieve this we randomly sampled 10% of the ParaCrawl V3. It means that only a fraction of the source training data contains document context. The enhanced model learns to use document information only when it is available. All test sets contain well delimited documents, *Baseline* models are evaluated on standard corpora while *Document* models are evaluated on corpora that have been augmented with document context. We evaluate the English-German systems on newstest2017, newstest2018 and newstest2019 where documents consist of newspaper articles

Corpora	#lines	# EN	# DE
Common Crawl	2.2M	54M	50M
Europarl V9†	1.8M	50M	48M
News Comm. V14†	338K	8.2M	8.3M
ParaCrawl V3	27.5M	569M	527M
Rapid 19†	1.5M	30M	29M
WikiTitles	1.3M	3.2M	2.8M
Total Training	34.7M	716M	667M
newstest2017†	3004	64K	60K
newstest2018†	2998	67K	64K
newstest2019†	1997	48K	49K

Table 2: Detail of training and evaluation sets for the English-German pair, showing the number of lines, words in English (EN) and words in German (DE). Corpora with document boundaries are denoted by †.

Corpora	#lines	# EN	# FR
News Comm. V14†	325K	9.2M	11.2M
ParaCrawl V3 (sampled)	3.1M	103M	91M
TED†	277K	7M	7.8M
Total Training	3.7M	119.2M	110M
tst2013†	1379	34K	40K
tst2014†	1306	30K	35K
tst2015†	1210	28K	31K

Table 3: Detail of training and evaluation sets for the English-French pair in both directions, showing the number of lines, words in English (EN) and words in French (FR). Corpora with document boundaries are denoted by †.

to keep consistency with the training data. English to French and French to English systems are evaluated over IWSLT TED tst2013, tst2014 and tst2015 where documents are transcriptions of TED conferences (see Table 3).

Prior to experiments, corpora are tokenized using Moses tokenizer [20]. To limit vocabulary size, we adopt the BPE subword unit approach [21], through the SentencePiece toolkit [22], with 32K rules.

4.2. Training details

We use the OpenNMT framework [23] in its TensorFlow version to create and train our models. All experiments are run on a single NVIDIA V100 GPU. Since the proposed approach relies on a preprocessing step and not on structural enhancement of the model, we keep the same Transformer architecture in all experiments. Our Transformer configuration is similar to the baseline of [2] except for the size of word and document vectors that we set to $d_{model} = 1024$, these vectors are fixed during training. We use $N = 6$ as the number of encoder layers, $d_{ff} = 2048$ as the inner-layer dimensionality, $h = 8$ attention heads, $d_k = 64$ as queries and keys dimension and $Pdrop = 0.1$ as dropout probability. All experiments, including baselines, are run over 600k training steps with a batch size of approximately 3000 tokens.

For all language pairs we trained a *Baseline* and a *Document* model. The *Baseline* is trained on a standard parallel corpus and is not aware of document embeddings, it is blind to the context and cannot link the sentences of a document. The *Document* model uses extracted word embeddings from the *Baseline* as initialization for its word vectors and also benefits from document embeddings that are computed from the extracted word embeddings. It is trained on an augmented corpus (see Table 1) and learns to make use of the document context. The *Document* model does not consider its embeddings as tunable parameters, we hypothesize that fine-tuning word and document vectors breaks the relation between them, leading to poorer results. We provide evidence of this phenomena with an additional system for the French-English language pair, noted *Document+tuning* (see Table 5) that is identical to the *Document* model except that it adjusts its embeddings during training.

The evaluated models are obtained by taking the average of their last 6 checkpoints, which were written at 5000 steps intervals. All experiments are run 8 times with different seeds to ensure the statistical robustness of our results. We provide *p-values* that indicate the probability of observing similar or more extreme results if the *Document* model is actually not superior to the *Baseline*.

4.3. Results

Table 4 presents results associated to the experiments for the English to German translation task, models are evaluated on the newstest2017, neswtest2018 and newstest2019 test sets. Table 5 contains results for both English to French and French to English translation tasks, models are evaluated on the tst2013, tst2014 and tst2015 test sets.

En→De: The *Baseline* model obtained State-of-The-Art BLEU and TER results according to [24, 25]. The *Document* system shows best results, up to 0.85 BLEU points over the *Baseline* on the newstest2019 corpus. It also surpassed the *Baseline*e by 0.18 points on the newstest2017 with strong statistical significance, and by 0.15 BLEU points on the newstest2018 but this time with no statistical evidence. These encouraging results prompted us to extend experiments to another language pair: English-French.

En→Fr: The *Document* system obtained the best results considering all metrics on all test sets with strong statistical evidence. It surpassed the *Baseline* by 1.09 BLEU points and 0.85 TER points on tst2015, 0.75 BLEU points and 0.76 TER points on tst2014, and 0.48 BLEU points and 0.68 TER points on tst2013.

Fr→En: Of all experiments, this language pair shows the most important improvements over the *Baseline*. The *Document* model obtained substantial gains with very strong statistical evidence on all test sets. It surpassed the *Baseline* model by 1.81 BLEU points and 1.02 TER points on tst2015, 1.50 BLEU points and 0.96 TER points on tst2014, and 1.29 BLEU points and 0.83 TER points on tst2013. The *Docu-*

Model	newstest2017		newstest2018		newstest2019	
En→De	BLEU	TER	BLEU	TER	BLEU	TER
Baseline	26.78	54.82	40.61	41.02	35.67	46.80
Document	**26.96****	**54.76**	**40.77**	**40.97**	**36.52***	**46.36***

Table 4: Results obtained for the English-German translation task, scored on three test sets using BLEU and TER metrics. p-values are denoted by * and correspond to the following values: $^{*}< .05$, $^{**}< .01$, $^{***}< .001$.

Translation direction	**Model**	tst2013		tst2014		tst2015	
		BLEU	TER	BLEU	TER	BLEU	TER
En→Fr	Baseline	46.05	37.83	43.38	39.71	41.41	42.18
	Document	**46.53***	**37.15****	**44.14****	**38.95****	**42.50*****	**41.33*****
Fr→En	Baseline	45.99	34.64	42.96	37.30	39.91	39.06
	Document+tuning	45.94	34.42	43.16	36.93	40.14	38.70
	Document	**47.28*****	**33.80*****	**44.46*****	**36.34*****	**41.72*****	**38.04*****

Table 5: Results obtained for the English-French and French-English translation tasks, scored on three test sets using BLEU and TER metrics. p-values are denoted by * and correspond to the following values: $^{*}< .05$, $^{**}< .01$, $^{***}< .001$.

ment+tuning system, which only differs from the fact that it tunes its embeddings, shows little or no improvement over the *Baseline*, leading us to the conclusion that the relation between word and document embeddings described by Eq. 1 must be preserved for the model to fully benefit from document context.

4.4. Manual Analysis

In this analysis we present some of the many cases that suggest the *Document* model can handle ambiguous situations. These examples are often isolated sentences where even a human translator could not predict the good translation without looking at the document, making it almost impossible for the *Baseline* model which is blind to the context. Table 6 contains an extract of these interesting cases for the French-English language pair.

Translation from French to English is challenging and often requires to take the context into account. The personal pronoun *"lui"* can refer to a person of feminine gender, masculine gender or even an object and can therefore be translated into *"her"*, *"him"* or *"it"*. The first example in Table 6 perfectly illustrate this ambiguity: the context clearly indicates that *"lui"* in the source sentence refers to *"ma fille"*, which is located three sentences above, and should be translated into *"her"*. In this case, the *Baseline* model predict the personal pronoun *"him"* while the *Document* model correctly predicts *"her"*. It seems that the *Baseline* model does not benefit from any valuable information in the source sentence. Some might argue that the source sentence actually contains clues about the correct translation, considering that *"robe à paillettes"* (*"sparkly dress"*) and *"baguette magique"* (*"magic wand"*) probably refer to a little girl, but we will see that the model makes similar choices in more restricted contexts. This example is relevant mainly because the actual reference to the subject *"ma fille"* is made long before the source sentence.

The second example in Table 6 is interesting because none of our models correctly translate the source sentence. However, we observe that the *Baseline* model opts for a literal translation of *"je peux faire le poirier"* (*"I can stand on my head"*) into *"I can do the pear"* while the *Document* model predicts *"I can wring"*. Even though these translations are both incorrect, we observe that the *Document* model makes a prediction that somehow relates to the context: a woman talking about her past disability, who has become more flexible thanks to yoga and can now twist her body.

The third case in table 6 is a perfect example of isolated sentence that cannot be translated correctly with no contextual information. This example is tricky because the word *"Elle"* would be translated into *"She"* in most cases if no additional information were provided, but here it refers to *"la conscience"* (*"consciousness"*) from the previous sentence and must be translated into *"It"*. As expected the *Baseline* model does not make the correct guess and predicts the personal pronoun *"She"* while the *Document* model correctly predicts *"It"*. This example present a second difficult part, the word *"son"* from the source sentence is ambiguous and does not, in itself, inform the translator if it must be translated into *"her"*, *"his"* or *"its"*. With contextual information we know that it refers to *"[le] monde physique"* (*"[the] physical world"*) and that the correct choice is the word *"its"*. Here the *Baseline* incorrectly predicts *"her"*, possibly because of its earlier choice for *"She"* as the subject. The *Document* model makes again the correct translation.

According to our results (see Table 5), the English-French language pair also benefits from document-level information but to a lesser extent. For this language pair, ambiguities about personal pronouns are less frequent. Other ambiguous phenomena like the formal mode (use of *"vous"* instead of *"tu"*) appear. Table7 presents an example of this kind of situation where the word *"You"* from the source sentence does not indicate if the correct translation is *"Vous"* or *"Tu"*.

Fr-En	
Context	[...] et quand ma fille avait quatre ans, nous avons regardé "Le Magicien d'Oz" ensemble. Ce film a complètement captivé son imagination pendant des mois. Son personnage préféré était Glinda, bien entendu.
Source	Ça lui donnait une bonne excuse pour porter une robe à paillettes et avoir une baguette magique.
Ref.	It gave her a great excuse to wear a sparkly dress and carry a wand.
Baseline	It gave **him** a good excuse to wear a glitter dress and have a magic wand.
Document	It gave **her** a good excuse to wear a glitter dress and have a magic wand.
Context	Mon père passait souvent les grandes vacances à essayer de me guérir ... Mais nous avons trouvé un remède miracle : le yoga. [...] j'étais une comique de stand-up qui ne tenait pas debout.
Source	Maintenant, je peux faire le poirier.
Ref.	And now I can stand on my head.
Baseline	Now I can **do the pear**.
Document	Now, I can **wring**.
Context	C'est le but ultime de la physique : décrire le flux de conscience. Selon cette idée, c'est donc la conscience qui met le feu aux équations. Selon cette idée, la conscience ne pendouille pas en dehors du monde physique ...
Source	Elle siège bien en son cœur.
Ref.	It's there right at its heart.
Baseline	**She** sits well in **her** heart.
Document	**It** sits well in **its** heart .

Table 6: Translation examples for the French-English pair. We took the best models of all runs for both the *Baseline* and the *Document* enhanced model

En-Fr	
Context	[The speaker in this example is an old police officer saving a man from suicide] But I asked him, "What was it that made you come back and give hope and life another chance ?" And you know what he told me ?
Source	He said "You listened."
Ref.	Il a dit : "Vous avez écouté."
Baseline	Il a dit : "**Tu** as écouté."
Document	Il a dit : "**Vous** avez écouté."

Table 7: Translation example for the English-French pair.

However it refers to the narrator of the story who is an old police officer. In this case, it is very likely that the use of formal mode is the correct translation. The *Baseline* model incorrectly predicts *"Tu"* and the *Document* model predicts *"Vous"*.

5. Conclusion

In this work, we presented a preliminary study of a simple approach for document-level translation. The method allows to benefit from the whole document context at the sentence level, leading to encouraging results. In our experimental setup, we observed improvement of translation outcomes up to 0.85 BLEU points in the English to German translation task and exceeding 1 BLEU point in the English to French and French to English translation tasks. Looking at the translation outputs, we provided evidence that the approach allows NMT models to disambiguate complex situations where the context is absolutely necessary, even for a human translator.

The next step is to go further by investigating more elaborate document embedding approaches and to bring these experiments to other languages (e.g.: Asian, Arabic, Italian, Spanish, etc.). To consider a training corpus with a majority of document delimited data is also very promising.

6. References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[3] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.

[4] R. Bawden, R. Sennrich, A. Birch, and B. Haddow, "Evaluating discourse phenomena in 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)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018.

[5] J. Tiedemann and Y. Scherrer, "Neural machine translation with extended context," in *Proceedings of the Third Workshop on Discourse in Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 82–92.

[6] L. Wang, Z. Tu, A. Way, and Q. Liu, "Exploiting cross-sentence context for neural machine translation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2826–2831.

[7] S. Jean, S. Lauly, O. Firat, and K. Cho, "Does neural machine translation benefit from larger context?" *arXiv preprint arXiv:1704.05135*, 2017.

[8] L. Miculicich, D. Ram, N. Pappas, and J. Henderson, "Document-level neural machine translation with hierarchical attention networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2947–2954.

[9] J. Zhang, H. Luan, M. Sun, F. Zhai, J. Xu, M. Zhang, and Y. Liu, "Improving the transformer translation model with document-level context," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 533–542.

[10] E. Voita, P. Serdyukov, R. Sennrich, and I. Titov, "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)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 1264–1274.

[11] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "MuST-C: a Multilingual Speech Translation Corpus," 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 2012–2017. [Online]. Available: https://www.aclweb.org/anthology/N19-1202

[12] Z. Gong, M. Zhang, and G. Zhou, "Cache-based document-level statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 909–919.

[13] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.

[15] R. Sennrich, B. Haddow, and A. Birch, "Controlling politeness in neural machine translation via side constraints," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 35–40.

[16] C. Chu, R. Dabre, and S. Kurohashi, "An empirical comparison of domain adaptation methods for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 385–391.

[17] C. Chu and R. Wang, "A survey of domain adaptation for neural machine translation," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1304–1319.

[18] L. Barrault, O. Bojar, M. R. Costa-jussà, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, S. Malmasi, C. Monz, M. Müller, S. Pal, M. Post, and M. Zampieri, "Findings of the 2019 conference on machine translation (wmt19)," in *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, August 2019, pp. 1–61.

[19] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, and F. Marcello, "The iwslt 2015 evaluation campaign," in *Proceedings of the twelfth International Workshop on Spoken Language Translation*, Da Nang, Vietnam, December 2015, pp. 2–10.

[20] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, 2007, pp. 177–180.

[21] 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)*, vol. 1, 2016, pp. 1715–1725.

[22] 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*, 2018, pp. 66–71.

[23] 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*, Vancouver, Canada, July 2017, pp. 67–72.

[24] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia, and M. Turchi, "Findings of the 2017 conference on machine translation (wmt17)," in *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 169–214.

[25] O. Bojar, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, and C. Monz, "Findings of the 2018 conference on machine translation (wmt18)," in *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, October 2018, pp. 272–307.