

Mémoire de fin d'études
Pour l'obtention du diplôme d'Ingénieur d'État en Informatique
Option : Systèmes Informatiques

Création d'un corpus de l'aphasie de Broca et
développement d'un système Speech-to-speech de
réhabilitation de la parole

Réalisé par :
BELGOUMRI Mohammed
Djameleddine
im_belgoumri@esi.dz

Encadré par :
Pr. SMAILI Kamel
smaili@loria.fr
Dr. LANGLOIS David
david.langlois@loria.fr
Dr. ZAKARIA Chahnez
c_zakaria@esi.dz

Table des matières

Page de garde	i
Table des matières	i
Table des figures	ii
Sigles et abréviations	iii
1 Traduction automatique et reconnaissance automatique de la parole	1
1.1 Traduction automatique	1
1.1.1 Traduction automatique à base de transformeur	2
Bibliographie	3

Table des figures

Sigles et abréviations

ASR	reconnaissance automatique de la parole
BPE	byte pair encoding
MT	traduction automatique
NMT	traduction automatique neuronale
RMBT	traduction automatique à base de règle
S2S	séquence-à-séquence

Chapitre 1

Traduction automatique et reconnaissance automatique de la parole

Dans le chapitre ??, nous avons introduit la traduction automatique (MT, de l'anglais : machine translation) et la reconnaissance automatique de la parole (ASR, de l'anglais : automatic speech recognition) comme avenues possibles pour la réhabilitation de la parole chez les patients de l'aphasie de Broca. Ensuite, dans le chapitre ??, nous avons présenté le problème général dont ces deux tâches sont des cas particuliers : celui de la modélisation séquence-à-séquence (S2S). Nous y avons posé formellement le problème et présenté les architectures neuronales majeures qui ont été utilisées pour le résoudre en les comparant. Dans ce chapitre, nous abordons dans plus de détails les aspects spécifiques de ces deux tâches. Nous étudions l'application des architectures présentées (notamment le transformeur) dans leur contexte.

1.1 Traduction automatique

Étant donné un langage source L_S sur un vocabulaire Σ_S , un langage cible L_C sur un vocabulaire Σ_C , et une relation d'équivalence¹ \sim sur $L_S \cup L_C$ la MT de L_S en L_C consiste à trouver une fonction calculable $f : L_S \rightarrow L_C$ qui vérifie

$$\forall x \in L_S, \quad f(x) \sim x \tag{1.1}$$

la relation \sim donne un sens d'identité entre les phrases. Sa définition peut varier dans sa rigueur et sa précision, elle est par exemple mathématiquement définie dans le contexte de compilation² (HADJ, 2015), elle a une définition floue dans le contexte de la MT du langage naturel (CHAN, 2015) et dans notre contexte de MT pour la correction des erreurs³ peut être vu comme un cas intermédiaire (BRYANT et al., 2022).

1. Dans le sens mathématique du terme, c-à-d. une relation réflexive, symétrique et transitive.

2. Qui est bien un exemple de MT où L_S et L_C sont des langages de programmation et \sim est la relation d'équivalence sémantique.

3. Il s'agit encore d'un exemple de MT.

Ce flou dans la définition empêche l'application efficace de la traduction automatique à base de règle (RMBT) dans ce contexte. La plupart des succès ont été eus par la traduction automatique neuronale (NMT) (YANG et al., 2020). Ces méthodes s'inscrivent facilement dans le cadre de l'apprentissage S2S tel que nous l'avons défini dans section ?? . Il est donc naturel de considérer les modèles étudiés dans le chapitre ?? comme des candidats pour la MT. Plus précisément, l'architecture de transformeur est la plus prometteuse en vue de l'analyse comparative effectuée dans le chapitre ?? (voir Table ??). De ce constat, nous consacrons cette section à l'étude de l'utilisation des transformeurs pour la NMT.

1.1.1 Traduction automatique à base de transformeur

Le transformeur tel que nous l'avons présenté dans section ?? peut être utilisé pour la MT. En effet, (VASWANI et al., 2017) l'ont appliqué à cette même tâche. Ayant déjà introduit l'architecture et le principe de fonctionnement du transformeur, nous concentrerons sur les particularités de son application à la MT.

Plongement lexical

Le transformeur — comme tout réseau de neurones — n'opère que sur des vecteurs. Pour lui faire passer des phrases, il faut donc les transformer en vecteurs. La première étape consiste à transformer les mots en vecteurs. C'est ce qu'on appelle un *plongement lexical* (ALMEIDA & XEXÉO, 2019).

La première étape dans le calcul des plongements lexicaux est de diviser le texte en unités lexicales (tokens). Il peut s'agir de mots, d'ensembles de mots ou d'unités plus petites qu'un mot. La sortie de cette étape est appelée un *vocabulaire*. Plusieurs techniques existent pour effectuer ce découpage. La plus simple est d'assimiler chaque mot à un token, mais d'autres techniques comme le byte pair encoding (BPE) proposent la segmentation des mots en sous-éléments. Le choix des sous-mots à garder dans le vocabulaire est le plus souvent une fonction de la fréquence dans le corpus (RAI & BORAH, 2021).

Une fois le vocabulaire construit, il lui faut associer un plongement. L'application qui mappe les tokens sur les vecteurs peut avoir une implémentation arbitraire. Il peut s'agir d'un simple tableau de vecteurs (PASZKE et al., 2019), comme il peut s'agir d'un réseau de neurones (CHURCH, 2017). Une fois qu'on a mis la phrase sous forme vectorielle, le transformeur peut la traiter (voir Section ??).

Bibliographie

- ALMEIDA, F., & XEXÉO, G. (2019). Word Embeddings : A Survey [arXiv :1901.09069 [cs, stat]], (arXiv :1901.09069). <http://arxiv.org/abs/1901.09069>
- BRYANT, C., YUAN, Z., QORIB, M. R., CAO, H., NG, H. T., & BRISCOE, T. (2022). Grammatical Error Correction : A Survey of the State of the Art [arXiv :2211.05166 [cs]], (arXiv :2211.05166). <http://arxiv.org/abs/2211.05166>
- CHAN, S.-w. (2015). *Routledge Encyclopedia of Translation Technology*. Routledge, Taylor & Francis Group.
- CHURCH, K. W. (2017). Word2Vec. *Natural Language Engineering*, 23(1), 155-162. <https://doi.org/10.1017/S1351324916000334>
- HADJ, A. A. e. (2015). *Analyse syntaxique et traduction : outils et techniques cours et exercices résolus*. Ellipses.
- PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., DESMAISON, A., KOPF, A., YANG, E., DEVITO, Z., RAISON, M., TEJANI, A., CHILAMKURTHY, S., STEINER, B., FANG, L., ... GARNETT, R. (2019). PyTorch : An Imperative Style, High-Performance Deep Learning Library. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- RAI, A., & BORAH, S. (2021). Study of Various Methods for Tokenization. In J. K. MANDAL, S. MUKHOPADHYAY & A. ROY (Éd.), *Applications of Internet of Things* (p. 193-200). Springer. https://doi.org/10.1007/978-981-15-6198-6_18
- VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, & POLOSUKHIN, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, 30. <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>
- YANG, S., WANG, Y., & CHU, X. (2020). A survey of deep learning techniques for neural machine translation. *arXiv preprint arXiv :2002.07526*.