Université de Montréal

Préentraînement d'un modèle ELECTRA

par

Simon Théorêt

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique,

December 2, 2024

© Simon Théorêt, 2025

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Préentraînement d'un modèle ELECTRA

présenté par

Simon Théorêt

a été évalué par un jury composé des personnes suivantes :

Nom du président du jury
(président-rapporteur)

Nom du directeur de recherche
(directeur de recherche)

Nom du membre de jury
(membre du jury)

Résumé

...sommaire et mots clés en français...

Abstract

...summary and keywords in english...

Table des matières

Liste des tableaux

Liste des figures

Liste des sigles et des abréviations

MLM	Modélisation de langage avec masque, de l'anglais *Masked Language Modeling*
TAL	Traitement automatique du langage
NER	Reconnaissance d'entitées, de l'anglais *Named-Entity Recognition*

Remerciements

Je tiens remercier Joss, pour sa précieuse aide tout au long de mon stage. Je n'aurais pas pu demander un meilleur superviseur.
Je remercie aussi Momo pour son support moral constant.

Introduction

Le domaine du traitement automatique des langues connaît une explosion fulgurante de techniques, de jeux de données et de modèles permettant de résoudre de nouveaux problèmes. Néanmoins, bon nombre de ces applications restent hors de portée des organisations désirant mettre en application des outils d'apprentissages automatique. En effet, la plupart des modèles de langues récents sont préentraînés sur des corpus majoritairement anglophones, avec des jetoniseurs spécialisés pour traiter le contenu anglophone. Ces deux facteurs limitent les modèles préentraînés disponibles ainsi que leur performance sur des tâches avec un corpus non anglophone.

Druide Inc. est une compagnie basée à Montréal dont le principal produit est Antidote, un logiciel de correction orthographique et grammaticale. Leur logiciel phare fait déjà usage de l'apprentissage profond pour leur moteur de correction en anglais, en plus d'utiliser un correcteur symbolique pour certains types d'erreurs. Le modèle utilisé en production pour la correction en anglais fait près de 2 corrections sur 3 et représente une part importante du moteur de correction. L'équipe de Druide désire mettre en place un modèle de correction similaire, mais adapté à la langue française. En particulier, ils désirent préentraîner un modèle ELECTRA avec un corpus et un jetoniseur francophones pour que le modèle puisse détecter les erreurs grammaticales présentes dans les textes des utilisateurs d'Antidote.

Pour la réalisation du projet, nous disposons d'un jeu de données d'environ 40 GB de données non structuré. De plus, l'entraînement du modèle se fait localement sur une machine ayant accès à 3 NVIDIA RTX A4000, disposant chacune de 16 GB de mémoire VRAM.

Chapitre 1

Druide et ELECTRA

L'équipe de Druide dispose de deux modèles déjà en place pour la correction des erreurs. Néanmoins, leur modèle en anglais corrige une plus grande gamme d'erreurs. Druide désire améliorer leur moteur de correction en français à l'aide de l'apprentissage profond.

1.1. Contraintes

Le modèle doit être intégré dans le logiciel principal de Druide, Antidote. Or, le logiciel Antidote est déployé sur les ordinateurs personnels des usagers. Cela implique d'importantes contraintes quant aux ressources disponibles pour l'exécution du modèle, notamment en ce qui à trait à la consommation de mémoire. De plus, le logiciel Antidote se doit d'être rapide, puisque attendre plusieurs minutes pour la correction d'un texte volumineux dégrade la qualité de l'expérience des utilisateurs. En d'autres mots, le modèle doit être rapide durant l'inférence. Finalement, le logiciel antidote cible deux système d'exploitation: Windows et MacOS. Le déploiement du modèle sur les machines des usagers se fait à l'aide des librairies ONNX[**2**] et CoreML. Il est donc nécessaire que le modèle soit supporté par les deux librairies. En résumé, nous avons des limites quant aux ressources disponibles durant l'inférence ainsi que des contraintes quant aux couches et modèles utilisables.

Ces contraintes ont poussé l'équipe du TAL de Druide à sélectionner des petits modèles Transformers[**4**] avec encodeur. Ces derniers contiennent environ 14 millions de paramètres.

1.2. Méthode de préentraînement ELECTRA

La méthode ELECTRA[**1**] est une méthode inspirée de la modélisation de langage avec masque (Masked Language Modeling; MLM), mais qui se veut plus efficace et rapide que le MLM. La méthode ELECTRA consiste à entraîner deux modèles: un petit modèle, appelé le générateur, et le modèle final, appelé le discriminant. Le générateur reçoit des jetons masqués

et doit prédire quel était le jeton original situé à la position du masque. Les prédictions du modèle sont échantillonnés, de façon à obtenir une nouvelle séquence, potentiellement différente de la séquence originale. Le discriminant reçoit la nouvelle séquence et à pour tâche de prédire quels jetons sont corrompus et lesquels n'ont pas été modifiés par le générateur. Seul le discriminant est réutilisé pour l'affinage. La méthode est visualisée dans la figure

Trois éléments rendent l'entraînement du discriminant plus facile. Premièrement, le générateur dispose de significativement moins de capacité que le discriminant. En effet, ce dernier contient en général 3 à 4 fois plus de paramètres (en excluant les couches de projections *embeddings*) que le générateur. De plus, les entrées du discriminant sont échantillonnées depuis la distribution engendrée par le générateur, au lieu de sélectionner les entrées les plus probables selon la distribution du générateur. Finalement, les poids du générateur sont initialisés aléatoirement et ce dernier est entraîné en même temps que le discriminant, rendant la tâche de plus en plus difficile au fur et à mesure que le générateur s'entraîne. Ces trois facteurs rendent la tâche du discriminant plus facile et permettent de générer des erreurs similaire à ce que le modèle rencontrera en production.

La méthode ELECTRA a été choisie pour deux raisons: c'est une méthode de préentraînement similaire à la correction d'erreurs dans un texte et la méthode ELECTRA permet d'augmenter l'efficacité du préentraînement en atteignant des performances similaires aux performances du MLM en moins d'itérations.

1.3. Affinage pour la détection d'erreurs

Une fois le modèle ELECTRA préentraîné, il est nécessaire d'adapter le modèle pour que celui-ci soit en mesure de détecter efficacement les erreurs dans les textes des utilisateurs. Druide a développé une liste des différents types d'erreurs, permettant de classifier les différents types d'erreurs en de grandes catégories, telles que les erreurs de virgules, les erreurs de mots manquants, les erreurs d'accord du nom, etc. Cette liste contient 750 différents types d'erreurs. Chaque erreur fait partie d'une de ces grandes catégories, et bon nombre de ces erreurs ont une sous-catégorie, précisant encore plus le contexte associée à l'erreur. La détection d'erreur est modélisée comme une tâche de détection d'entité nommée (DEN/NER), dans laquelle chaque jeton dispose d'un classe. Les classes d'erreurs sont représenté avec un identifiant, tandis que la classe représentant l'absence d'erreurs est représenté par l'identifiant O. Le modèle à comme objectif de spécifier la classe de chaque jeton de la séquence. Le schéma $IOB2$ [**3**] est utilisé pour représenter sans ambiguïté les jetons contigus contenus dans la même erreur.

1.4. Infrastructures en place

Notre tâche principale consistait à préentraîner un modèle ELECTRA. Or, un modèle ELECTRA est déjà utilisé pour la tâche de correction en anglais. Ce dernier n'a pas été préentraîné par Druide. En effet, la librairie Transformers[**5**] permet un usage libre de différents modèles ELECTRA préentraînés. De plus, il existe quelques modèles ELECTRA préentraînés sur des corpus francophone. Cependant, aucun d'entre eux ne respectent nos contraintes de tailles et de vitesse. Il est donc nécessaire d'entraîner un modèle à partir d'un initialisation aléatoire.

Nous disposons de deux corpus déjà préparés pour préentraîner et affiner un modèle Electra. Le corpus de préentraînement est une collection de textes non structuré provenant de nombreuses sources, notamment des manuels, des articles de blogues, des livres. Ce corpus de préentraînement est appelé corpus des Combis et représente 40 gigaoctes (Go) de données et 7 milliard de jetons. C'est un corpus deux fois plus grand que le corpus de préentraînement utilisé pour le préentraînement par Google du modèle ELECTRA de même taille. Pour l'affinage, Druide dispose d'un corpus contenant près de 100000 annotations sur des textes francophones. Ces annotation sont fournies par Druide et proviennent d'équipes de linguistes et d'annotateurs corrigeant des textes et classifiant les erreurs qu'ils y trouvent en fonction des types d'erreurs proposés par Druide.

Chapitre 2

Préentraînement d'un modèle initial

Le préentraînement d'un modèle de langue se fait en trois étapes. Il est nécessaire de prétraiter les données, de sélectionner un jetonniseur adapté à la tâche ainsi que d'entraîner le modèle sur la tâche de préentraînement.

2.1. Normalisation des données et entraînement d'un jetoniseur

La normalisation consiste à réduire le nombre de charactèrs différents contenus dans le corpus. C'est une étape importante puisqu'elle permet de réduire la taille du vocabulaire du jetoniseur sans pour autant perdre des éléments syntaxiques. Notre normalisation consistait à tranformer tous les espaces en le même charactère d'espace (espaces insécables, tabulations), de transformer tous les guillements (guillements français, guillemets informatiques, etc.) en guillemets anglais, de retirer les espaces en trop et modifier les types d'apostrophes pour que ceux-ci soient uniformes. La normalisation modifie aussi certains les espacements entre certains mots, de façon à ce que par exemple le texte "11 ème étage" devienne "11ème étage".

Une fois le texte normalisé, il est possible d'entraîner un jetoniseur adapté à la tâche. En l'occurence, nous avons sélectionner le jetoniseur Wordpiece. C'est le jetoniseur choisi par les auteurs de l'article de ELECTRA et est actuellement utilisé en production chez Druide. Pour l'entraînement du jetoniseur, nous utilisons le corpus des combis normalisé, comprenant environ 40 GO de données. Les hyper-paramètres sélectionnés pour le jetoniseur sont dcnnés dans la figure

Chapitre 3

Modèle final

Chapitre 4

Conclusions

Références

[1] Kevin Clark, Minh-Thang Luong, Quoc V. Le, et Christopher D. Manning, *Electra: Pre-training text encoders as discriminators rather than generators*, 2020.

[2] ONNX Runtime developers, *Onnx runtime*, `https://onnxruntime.ai/`, 2021, Version: x.y.z.

[3] Lance A. Ramshaw et Mitchell P. Marcus, *Text chunking using transformation-based learning*, 1995.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, et Illia Polosukhin, *Attention is all you need*, 2023.

[5] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, et Alexander M. Rush, *Transformers: State-of-the-art natural language processing*, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (Online), Association for Computational Linguistics, October 2020, pp. 38–45.

Annexe A

Le titre

A.1. Section un de l'Annexe A

...texte...

Annexe B

Les différentes parties et leur ordre d'apparition

J'ajoute ici les différentes parties d'un mémoire ou d'une thèse ainsi que leur ordre d'apparition tel que décrit dans le guide de présentation des mémoires et des thèses de la Faculté des études supérieures. Pour plus d'information, consultez le guide sur le site web de la facutlé (www.fes.umontreal.ca).

Ordre des éléments constitutifs du mémoire ou de la thèse		
1.	La page de titre	obligatoire
2.	La page d'identification des membres du jury	obligatoire
3.	Le résumé en français et les mots clés français	obligatoires
4.	Le résumé en anglais et les mots clés anglais	obligatoires
5.	Le résumé dans une autre langue que l'anglais ou le français (si le document est écrit dans une autre langue que l'anglais ou le français)	obligatoire
6.	Le résumé de vulgarisation	facultatif
7.	La table des matières, la liste des tableaux, la liste des figures ou autre	obligatoires
8.	La liste des sigles et des abréviations	obligatoire
9.	La dédicace	facultative
10.	Les remerciements	facultatifs
11.	L'avant-propos	facultatif
12.	Le corps de l'ouvrage	obligatoire
13.	Les index	facultatif
14.	Les références bibliographiques	obligatoires
15.	Les annexes	facultatifs
16.	Les documents spéciaux	facultatifs