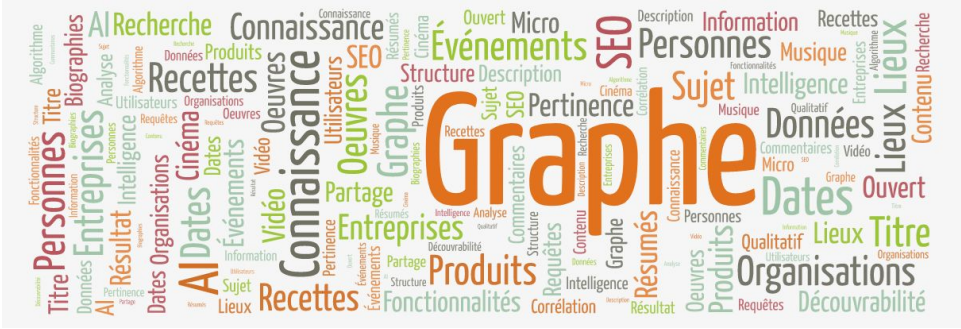
Franchir le « mur des données massives »:

étude de la faisabilité de quelques techniques d'amplification textuelle simples, pratiques et robustes.

Claude COULOMBE
candidat au doctorat en informatique cognitive - TÉLUQ / UQAM
Consultant - Lingua Technologies inc.
directeur scientifique - DataFranca

Colloque 2019
Le web sémantique au Québec
6 juin 2019 - TÉLUQ



Un secret bien gardé de l'AP

Il y a un prix à payer pour l'apprentissage profond (AP) dont on ne parle pas beaucoup.

Un secret bien gardé de l'apprentissage profond
L'apprentissage profond exige d'énormes quantités de données, l'accès à d'importantes infrastructures de calcul avec des processeurs graphiques et une bonne dose de savoir-faire pour préparer les données, bâtir les architectures et entraîner les réseaux de neurones profonds.

Le mur des données massives

Il n’est pas rare de se retrouver avec des quantités nettement insuffisantes de données pour entraîner un modèle profond.

Ce « *mur des données massives* » représente un défi pour les communautés linguistiques minoritaires sur la Toile, les organisations, les laboratoires et les entreprises qui rivalisent avec les géants du GAFAM.

La nécessité d'une grande quantité de données n'est pas particulière à l'AP, elle est liée à la complexité de la tâche à résoudre.

Franchir le « mur des données massives »

Le mur des données massives (2)

Règles empiriques sur la quantité de données en apprentissage supervisé profond
Number of training cases ~ number of trainable parameters [Sutskever, 2013]
… a supervised deep learning algorithm will generally achieve acceptable performance with around 5,000 labeled examples per category [Goodfellow, Bengio & Courville, 2016]
… and will match or exceed human performance when trained with a dataset containing at least 10 million labeled examples [Goodfellow, Bengio & Courville, 2016]
If you only have 10 examples of something, it's going to be hard to make deep learning work. If you have 100,000 things you care about, records or whatever, that's the kind of scale where you should really start thinking about these kinds of techniques. Jeff Dean, Google Brain [Frank, 2017]

Augmentation/amplification des données

L'idée est de créer de nouvelles données à partir des données existantes.

Par abus de langage on parle d'augmentation des données, mais il s'agit plutôt d'une amplification, car on part de données existantes pour en créer de nouvelles, tout en préservant le sens qui doit demeurer invariant. L'idée de « *transformation sémantiquement invariante* » est au coeur du processus d'amplification des données

On parle aussi de données synthétiques, de données générées, de données artificielles ou même de fausses données.

Travaux précurseurs

En vision par ordinateur, c'est une pratique courante de créer de nouvelles images par des transformations géométriques qui préservent la similitude [Simard, Steinkraus & Platt, 2003], [Ha & Bunke, 1997]. Ce type d’amplification de données a été utilisé pour remporter ImageNet en 2012 [Krizhevsky, Sutskever & Hinton, 2012].

En reconnaissance vocale, l'amplification des données est obtenue en manipulant le signal en le ralentissant ou l’accélérant [Ko et al, 2015], par injection de bruit et par modification du spectrogramme [Jaitly & Hinton, 2013].

État de l’art en amplification de données textuelles

L’amplification textuelle est peu répandue
However, usage of data augmentation for NLP has been limited. [Kobayashi, 2018]

Jusqu'à tout récemment, la seule technique d'amplification des données textuelles vraiment répandue était la substitution lexicale qui consiste à remplacer un mot par son synonyme à l'aide d'un thésaurus [Zhang & LeCun, 2015].

C’est un secret de polichinelle que les règles codées à la main, incluant l’injection de bruit, sont utilisées. Un exemple concret est la bibliothèque NoiseMix [Bittlingmayer, 2018].

Pourquoi?

1- Les données textuelles sont difficiles à traiter car elles sont symboliques, discrètes, compositionnelles et dispersées [Goldberg, 2017] & hiérarchiques, bruitées, bourrées d'exceptions, ambigües;

2- Les techniques basées sur la descente de gradient ne s'appliquent pas directement à des données discrètes [Goodfellow, 2016];

3- Il est difficile de générer des données textuelles réalistes. Par exemple, les auto-encodeurs classiques ne contraignent pas l'espace latent et travaillent mot par mot [Bowman et al, 2015];

4- Tout le monde le fait, mais personne n'en parle.

Pourquoi? (2)

Peut-être que l'explication est plus sociologique que technique. La plupart des travaux de recherche actuels sur l’amplification des données textuelles cherchent des solutions d’apprentissage de bout en bout, ce qui revient à « *utiliser des réseaux de neurones pour nourrir des réseaux de neurones* ».

Ce sont des travaux de recherche à long terme, brillants et essentiels, mais il est urgent de trouver des solutions pratiques et à court terme.

Objectif

Le présent travail a pour objectif l'étude de la faisabilité de techniques d'amplification textuelle par prétraitement des données qui soient pratiques, robustes et simples à mettre en oeuvre;

Certaines technique existantes ont été testées pour fin de comparaisons comme l'injection de bruit ou l'emploi d'expressions régulières. D'autres ont été modifiées ou améliorées comme la substitution lexicale. Enfin, des techniques plus innovatrices, comme la génération de paraphrases par la rétrotraduction et la transformation d'arbres syntaxiques, font appel à des services en ligne pratiques, robustes et faciles à utiliser.

Quelques règles

Règle du respect de la distribution statistique

Les données amplifiées doivent suivre une distribution statistique similaire à celle des données originales.

Règle d'or de plausibilité

Un être humain ne devrait pas pouvoir distinguer entre les données amplifiées et les données originales.

[Géron, 2017b]

Règle d'invariance sémantique

L'amplification des données implique des transformations sémantiquement invariantes.

Injection de bruit textuel

Injection de bruit textuel
L'injection de bruit textuel faible est une transformation sémantiquement invariante.
L'injection de bruit textuel fort n'est pas une transformation sémantiquement invariante.

L'injection de bruit textuel consiste à retirer, remplacer ou ajouter au hasard un caractère alphabétique ou à une table d'équivalences pré-établies. L'algorithme procède en 3 étapes. 1) Choix aléatoire d'un caractère à remplacer dans le texte. 2) Tirage aléatoire d'un opérateur (retrait, remplacement, ajout, équivalence). 3) Choix aléatoire du caractère ou de la séquence de remplacement [Roquette, 2018].

Injection de fautes d'orthographe

Injection de fautes d'orthographe

L'injection de fautes d'orthographe est une transformation sémantiquement invariante.

L'idée est de générer des textes contenant des fautes d'orthographe courantes afin d'entraîner nos modèles qui deviendront ainsi plus robustes à ce type particulier de bruit textuel.

L'algorithme d'injection de fautes d'orthographe se base sur une liste des erreurs d'orthographe les plus courantes en anglais. Cette liste a été compilée par l'éditeur des dictionnaires Oxford [Oxford Dictionaries, 2018].

Substitution lexicale avec dictionnaire

Règles pour la substitution lexicale
La substitution d'un mot par un vrai synonyme est une transformation sémantiquement invariante.
La substitution d'un mot par un hyperonyme (mot plus général) est une transformation sémantiquement invariante.
La substitution d'un mot par un hyponyme (mot plus spécifique) n'est généralement pas une transformation sémantiquement invariante.
La substitution d'un mot par un antonyme n'est pas une transformation sémantiquement invariante.

La substitution lexicale consiste à proposer des mots qui pourront se substituer à un mot donné [Zhang & LeCun, 2015]. Ces mots sont typiquement des synonymes tirés d'un dictionnaire comme Wordnet [Miller & al, 1990]. Nous avons également expérimenté l'utilisation de vecteurs de mots qui tiennent compte de la polysémie avec l'algorithme AdaGram [Bartunov et al, 2016].

Génération de paraphrases

La génération de paraphrases, irréaliste et coûteuse

Therefore, the best way to do data augmentation would have been using human rephrases of sentences, but this is unrealistic and expensive due the large volume of samples in our datasets. As a result, the most natural choice in data augmentation for us is to replace words or phrases with their synonyms.

[Zhang & LeCun, 2015]

Définition de la paraphrase idéale

In addition to being meaning-preserving, an ideal paraphrase must also diverge as sharply as possible in form from the original while still sounding natural and fluent.

[Chen & Dolan, 2011]

Génération de paraphrases avec exp. rég.

Exemples d'une transformation textuelle de surface

Le passage d'une forme verbale vers une forme contractée et vice versa est une transformation de surface sémantiquement invariante à la condition que l'on préserve les ambiguïtés.

```
I am => I'm, you are => you're, he is => he's, it is => it's, she is => she's, we are => we're, they
 are => they're,, I have => I've, you have => you've, we have => we've, they have => they've, he has
=> he's, it has => it's, she has => she's, I will => I'll, you will => you'll, he will => he'll, are
 not => aren't, is not => isn't, was not => wasn't, ..., I'm => I am, I'll => I will, you'll => you
 will, he'll => he will, aren't => are not, isn't => is not, wasn't => was not, weren't => were not,
couldn't => could not, don't => do not, doesn't => does not, didn't => did not, mustn't => must not,
 shouldn't => should not, can't => can not, can't => cannot, won't => will not, shan't => shall not
```

Dans un premier temps, les transformations de surface qui peuvent être produites avec des expressions régulières sont à privilégier car elles sont simples et très performantes en terme de calculs.

Génération de paraphrases avec exp. rég. (2)

Exemple de transformations interdites car elles lèvent une ambiguïté sans justification

she's => she is
she's => she has

Règle empirique du « respect de l'ambiguïté »

Une transformation qui génère une ambiguïté ou une imprécision est généralement considérée comme sémantiquement invariante.

Une transformation qui lève une ambiguïté, en précisant une information, ne peut pas être considérée comme une transformation sémantiquement invariante, à moins que l'information précisée ne soit motivée par le contexte.

Génération de paraphrases par des arbres syntaxiques

La génération de paraphrases par transformation d'arbres syntaxiques est directement inspirée des travaux de Michel Gagnon de Polytechnique Montréal [Gagnon & Da Sylva, 2005], [Zouaq, Gagnon & Ozell, 2010].

Le résultat de l'analyse d'une phrase selon un formalisme en grammaire de dépendances est un arbre dont les nœuds sont les mots de la phrase et les arêtes (liens) les dépendances syntaxiques entre les mots.

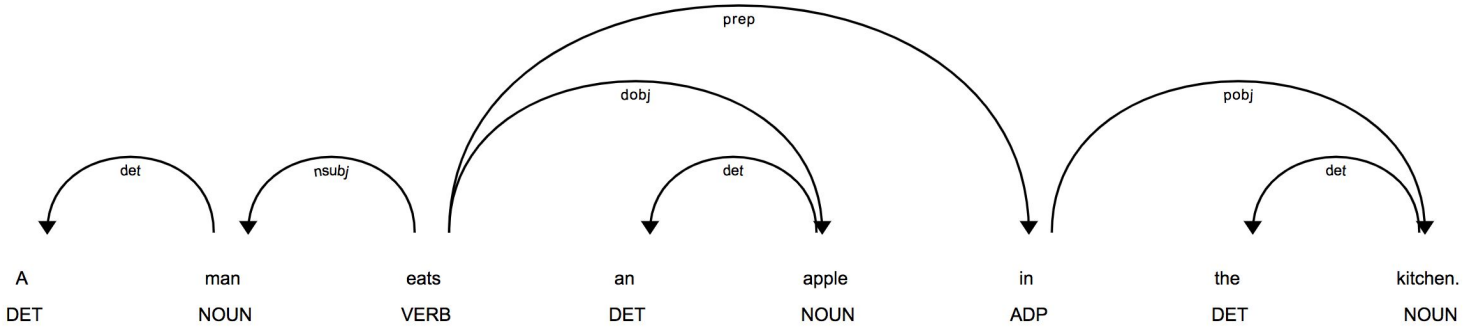
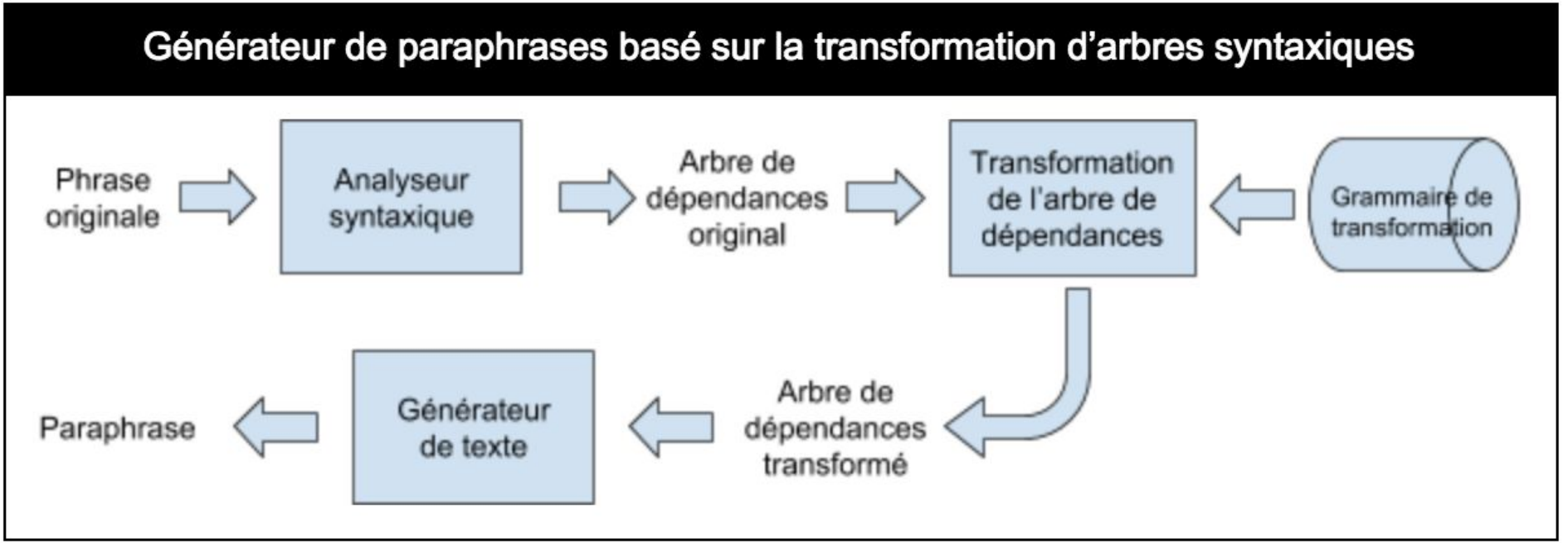


Schéma produit avec l'aide de spaCy [Honnibal & Montani, 2017]

Génération de paraphrases par des arbres syntaxiques (2)



Chaque phrase est soumise à l'analyseur SyntaxNet [Petrov, 2016], [Kong et al, 2017] qui se base sur des techniques d'apprentissage profond et la bibliothèque TensorFlow. Le code est exécuté dans l'infrastructure infonuagique de Google via l'IPA Cloud Natural Language [Google, 2018a].

Génération de paraphrases par des arbres syntaxiques (3)

Exemples de transformations sémantiquement invariantes d'arbres syntaxiques
Le passage de la forme passive à la forme active et vice versa est une transformation sémantiquement invariante.
Le remplacement d'un nom ou d'un syntagme nominal par un pronom est une transformation sémantiquement invariante.
Le retrait d'un adjectif, d'un adverbe, d'un groupe adjectival ou d'un groupe adverbial est une transformation sémantiquement invariante.

Les règles de transformations sont construites manuellement selon la typologie des paraphrases de [Vila, Martí & Rodríguez, 2014] conformément au principe d'ingénierie de Pareto 20/80. Marie Bourdon, linguiste informaticienne chez Coginov inc Montréal, a apporté une aide précieuse à ce travail.

Génération de paraphrases par des arbres syntaxiques (4)

Illustration du passage de la voix active à la voix passive

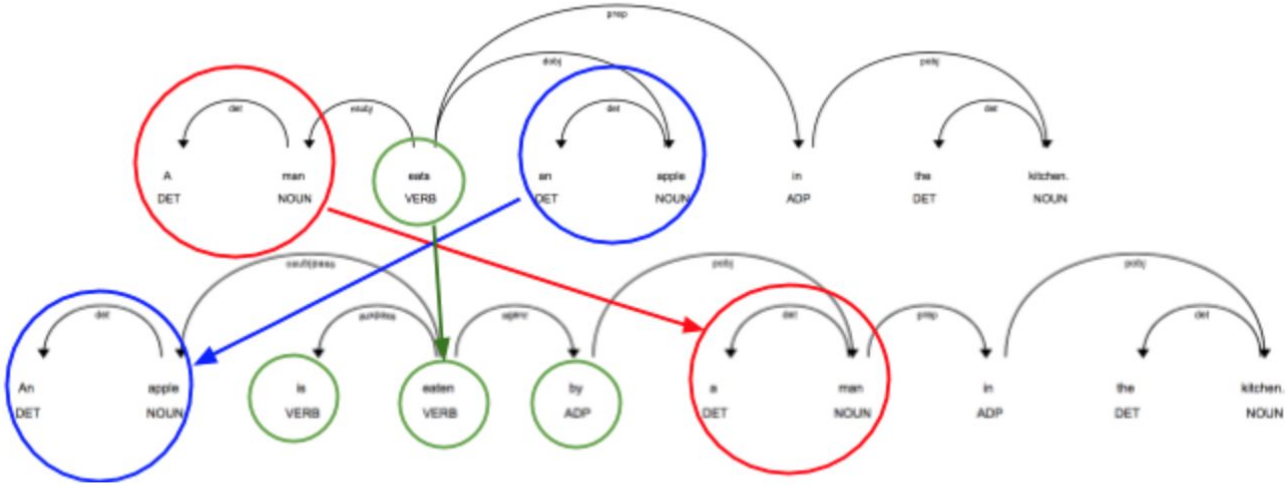


Schéma produit avec l'aide de spaCy [Honnibal & Montani, 2017]

Passage à la voix passive de la phrase « A man eats an apple in the kitchen.» La tête de la structure de dépendances est le verbe « eat ». La régle de transformation commence par échanger le groupe sujet « man » (en rouge) et le groupe objet « apple » (en bleu) puis l'accord du verbe « eat » est modifié (en vert) pour donner une nouvelle structure de dépendances qui une fois aplatie génère la phrase « An apple is eaten by a man in the kitchen. ».

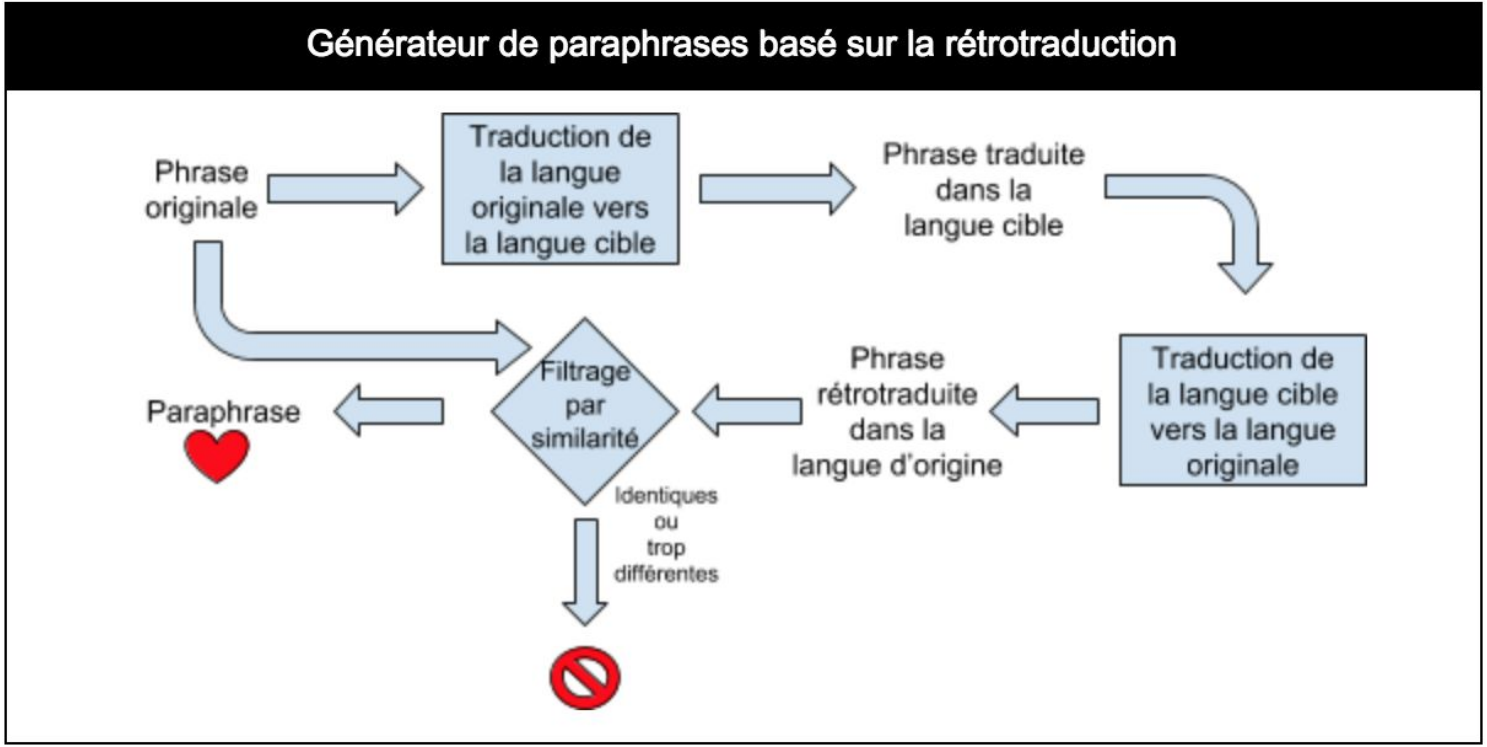
Génération de paraphrases par rétrotraduction

L'étude de la transformation par rétrotraduction (back-translation) a été suggérée par le physicien Antoine Saucier à l'occasion d'une «discussion de corridor» à la Polytechnique Montréal au printemps 2015.

La rétrotraduction est un vieux truc utilisé pour tester la qualité d'un programme de traduction automatique qui consiste à traduire vers la langue d'origine un texte déjà traduit depuis cette langue. Or il existe toujours un grand nombre de traductions correctes à cause de l'immense productivité combinatoire de la langue naturelle. Par définition, toutes ces traductions équivalentes sont des paraphrases.

Historiquement, la première mention de l'emploi de la rétrotraduction sous le terme «round trip machine translation» pour introduire des variantes dans les données textuelle se trouve dans un article d'une équipe du King's College London présenté à la conférence ISCOL 2015 [Lau, Clark & Lappin, 2015].

Génération de paraphrases par rétrotraduction (2)



La phrase originale est traduite par Google Translate, un système de traduction automatique neuronale. Un second appel produit une rétro-traduction. Ensuite, la phrase "rétro-traduite" est filtrée pour être retenue ou écartée.

Génération de paraphrases par rétrotraduction (3)

Amplification de données textuelles en utilisant la rétrotraduction
La rétrotraduction de bonne qualité est une transformation sémantiquement invariante.
La rétrotraduction de piètre qualité n'est pas une transformation sémantiquement invariante.

Pour le filtrage basé sur la similitude, nous avons d'abord opté pour la simple différence de longueur entre le texte d'origine et le texte traduit comme [Wieting, Mallinson & Gimpel, 2017], puis nous avons utilisé la métrique BLEU et un modèle logistique entraîné sur des données étiquetées manuellement..

Nous utilisons les services de traduction en ligne de Google via Google Translate API. Une fois inscrit et la clé d'accès au service Google Translate obtenue, tout le code requis est contenu dans une petite cellule d'un carnet iPython.

Expérimentation - choix de la tâche et du jeu de données

Pour valider et comparer les différentes techniques d'amplification textuelle, nous avons choisi un problème simple qui fait intervenir un jeu de données normalisé et des architectures courantes de réseaux de neurones profonds. Ainsi, nous arriverons plus facilement à isoler l'effet de l'amplification des données textuelles.

Nous avons opté pour la tâche de prédiction de la polarité positive ou négative de critiques de film contenues dans la base IMDB [Pang, Lee & Vaithyanathan, 2002]. Plus spécifiquement nous avons utilisé le jeu de données «polarity dataset v2.0» qui comporte 1000 critiques positives et 1000 critiques négatives extraites de la base de données IMDB.

Pour cette tâche et avec ce corpus précis, les performances s'échelonnent de 70% pour les algorithmes d'apprentissage classiques jusqu'à plus de 90% pour des réseaux de neurones profonds finement ajustés.

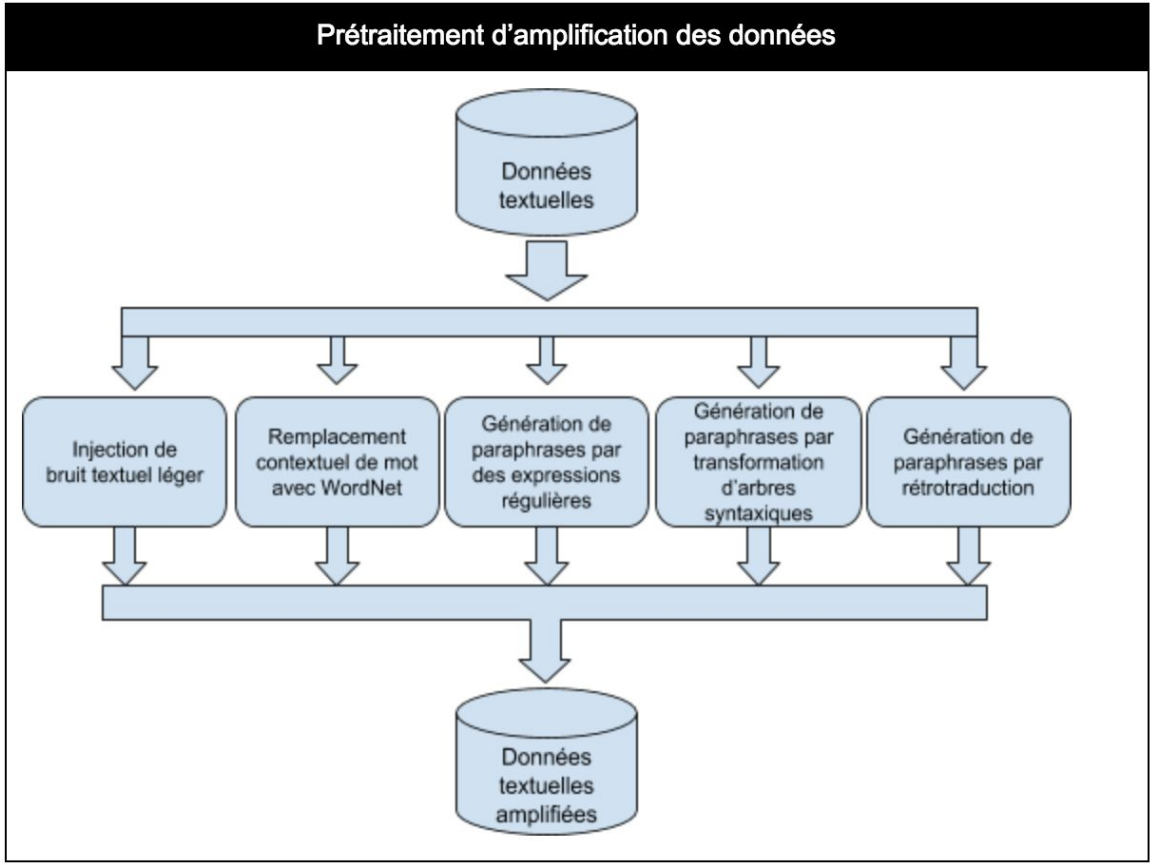
Expérimentation - protocole

L’expérimentation se divise en deux (2) phases: 1) une phase de prétraitement où se réalise l’amplification des données textuelles selon différentes techniques 2) la phase d’entraînement des modèles avec différentes architectures de réseaux de neurones profonds sur les données originales et les données amplifiées.

Les modèles sont ensuite comparés pour voir s’il y a amélioration ou dégradation des performances au niveau de la prédiction (exactitude) des modèles. Les erreurs d’entraînement et les mesures F1 ont également été calculées.

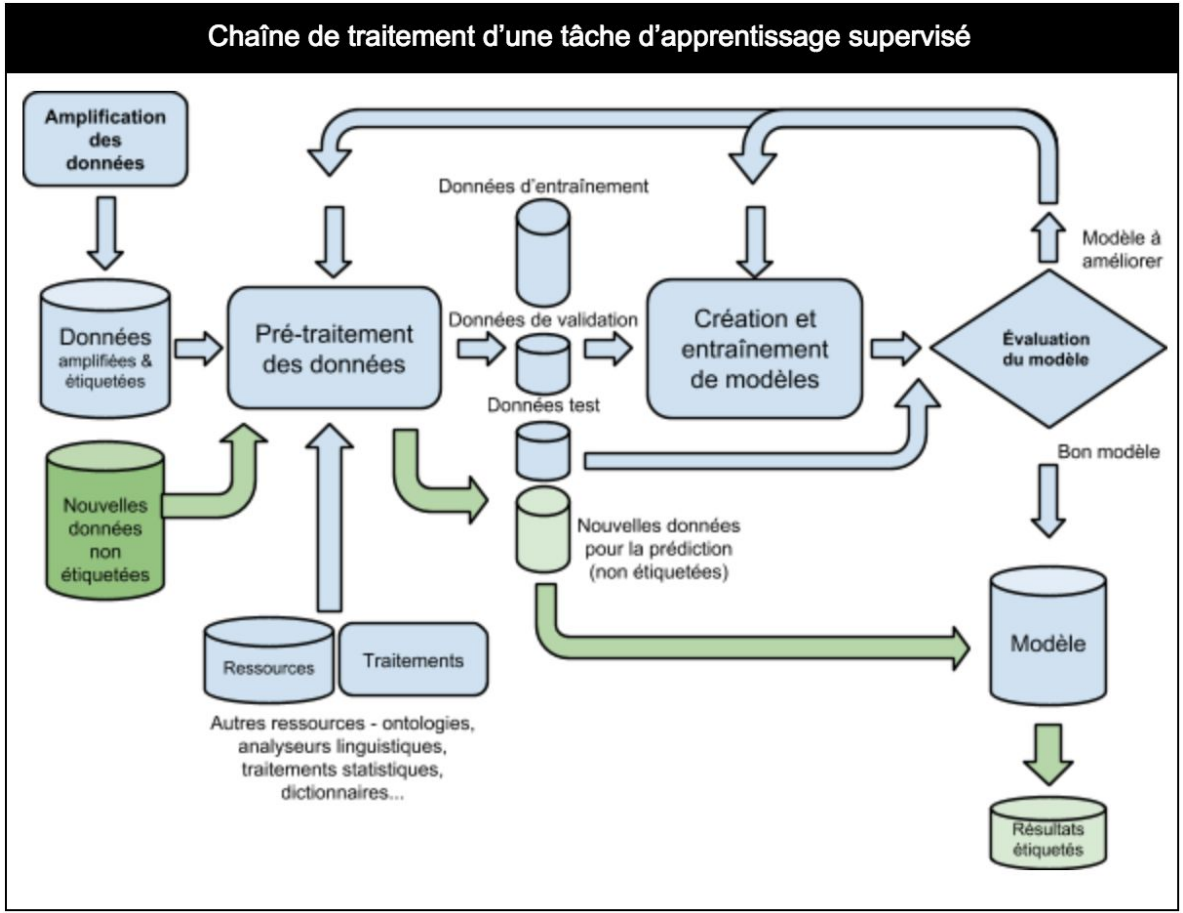
Étant donné nos ressources de calcul très limitées, nous avons restreint nos expériences à la seule preuve de concept. Beaucoup de travail reste à faire pour explorer chaque méthode d’amplification textuelle dans le détail.

Expérimentation - prétraitement



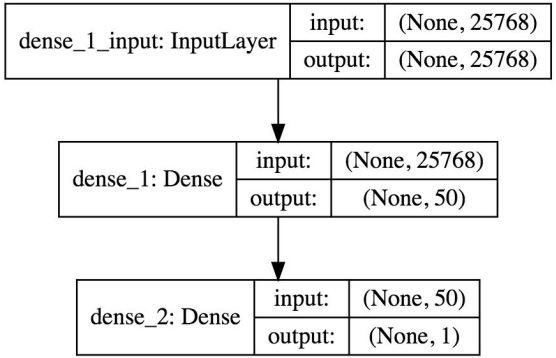
Amplification des données textuelles pour l'apprentissage profond

Expérimentation - entraînement des modèles

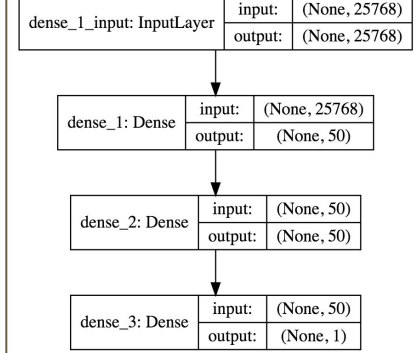


Expérimentation - entraînement des modèles (2)

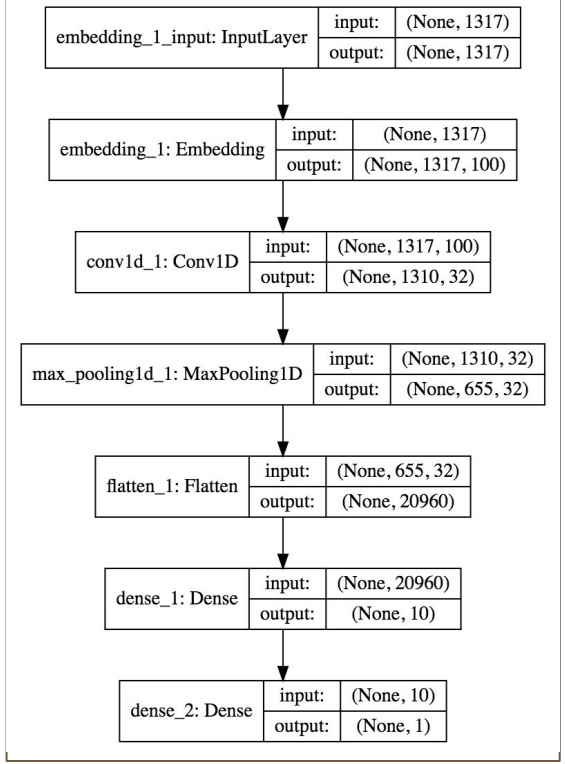
Perceptron multicouche simple
(PMC simple)



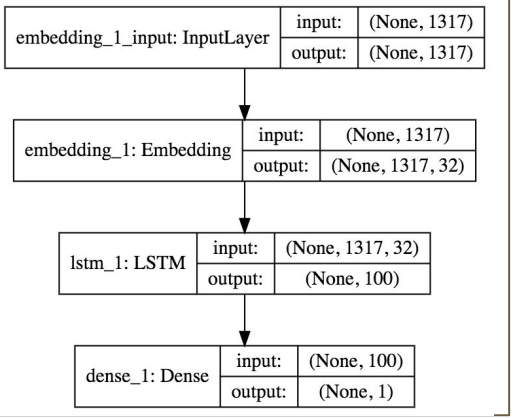
Perceptron multicouche
à deux couches cachées (PM2C)



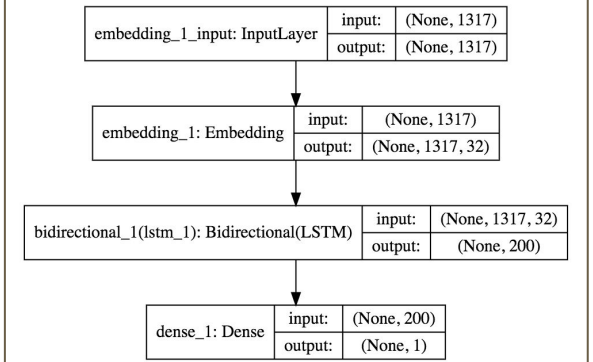
Réseau à convolutions 1D



Réseau récurrent à
longue mémoire court terme (LMCT)



Réseau récurrent à longue mémoire
court terme bidirectionnel (biLMCT)



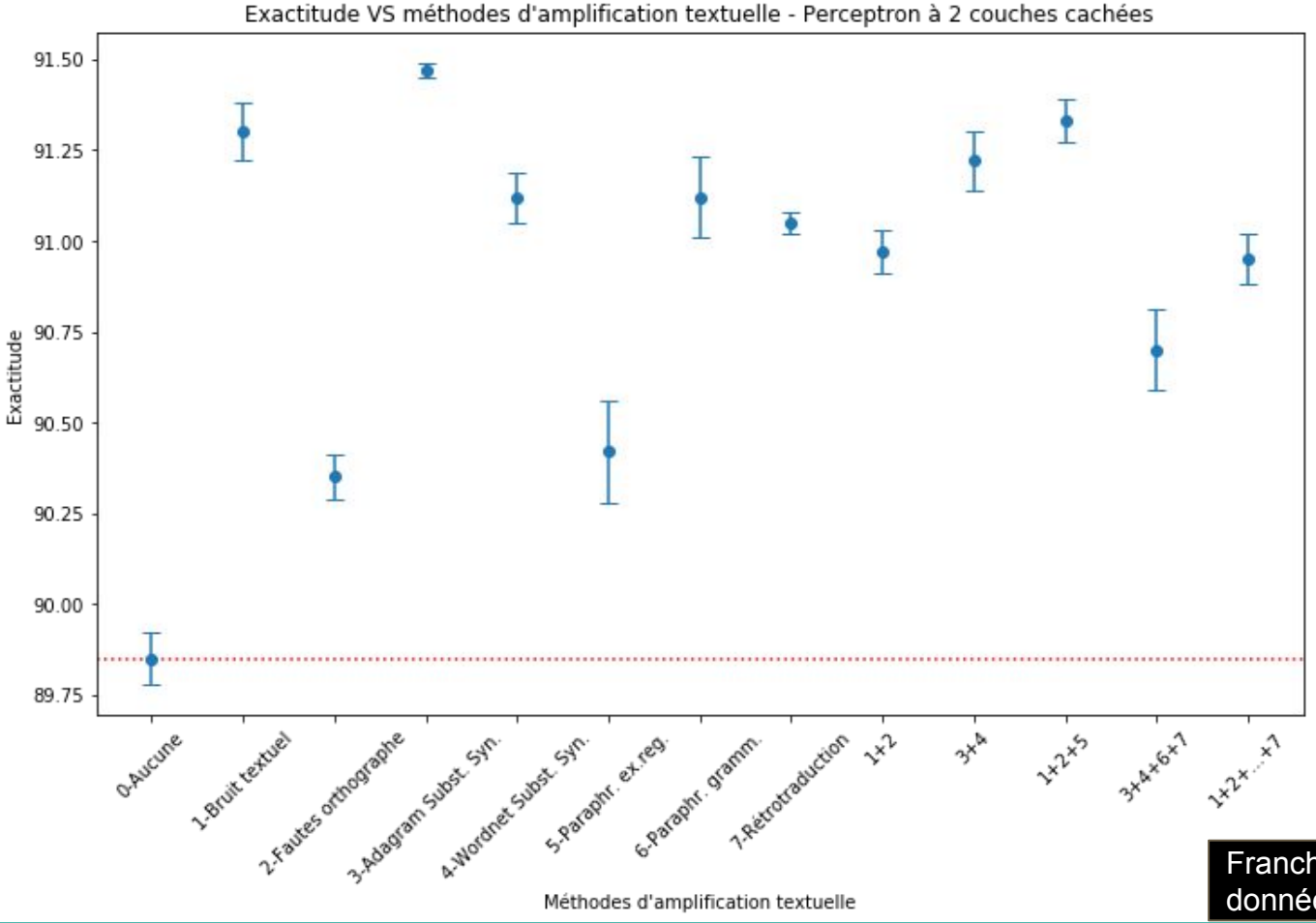
Résultats

Les résultats montrent que les différentes techniques d'amplification textuelle étudiées améliorent les performances des modèles, dans une fourchette 4.3% à 21.6%, par rapport à la base de référence qui est le jeu de données initial, sans amplification des données. On constate également d'importantes fluctuations statistiques.

Les résultats sont présentés dans les prochaines diapositives avec des graphiques comportant des barres d'erreur qui représentent l'écart-type. Cette façon de présenter l'information a l'avantage de faire ressortir l'effet comparé des différentes techniques d'amplification textuelle.

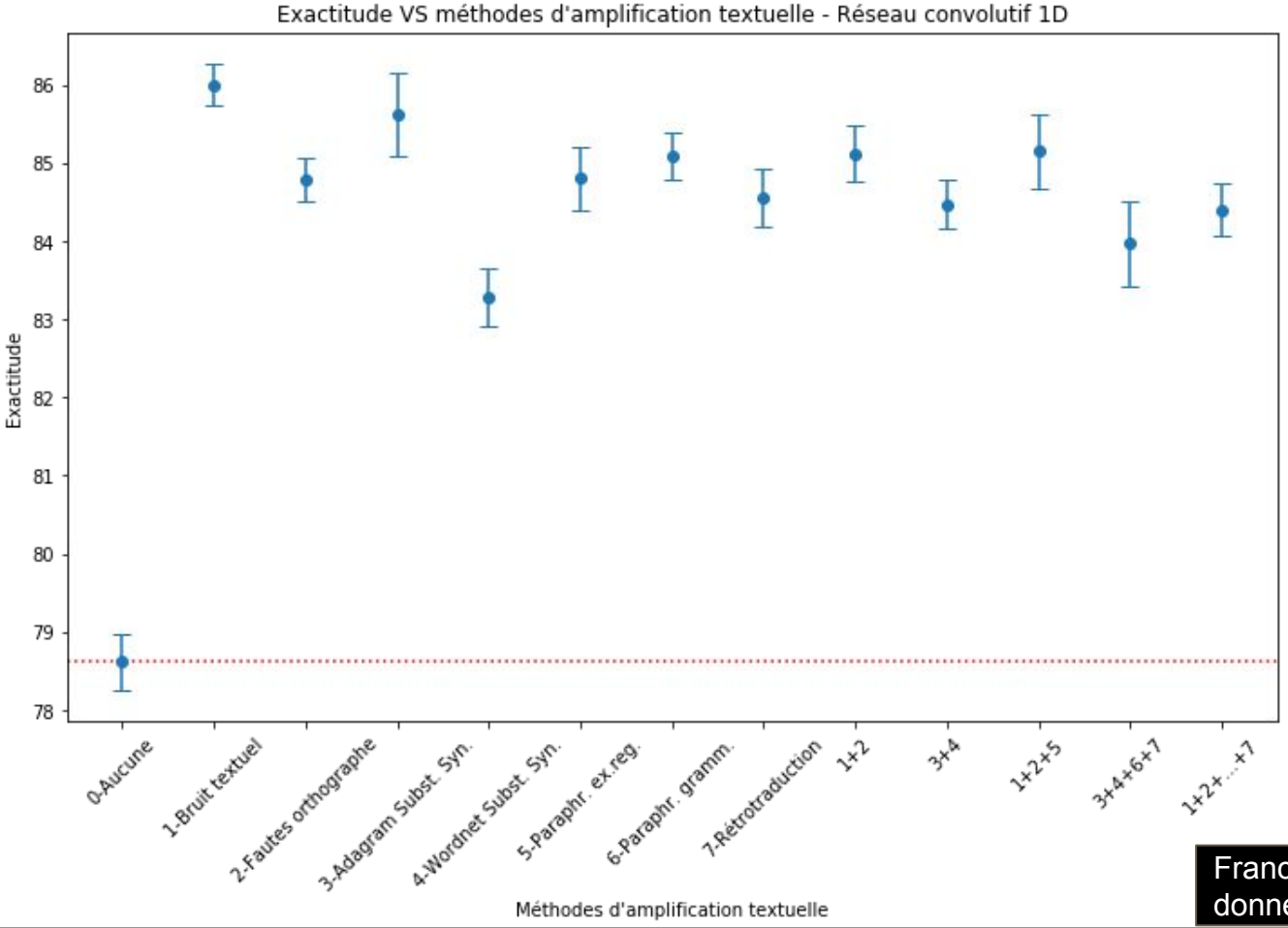
il est important de comprendre que l'objectif n'était pas d'obtenir la meilleure performance en terme d'exactitude mais d'isoler l'effet de l'amplification textuelle.

Résultats - Perceptron multicouche



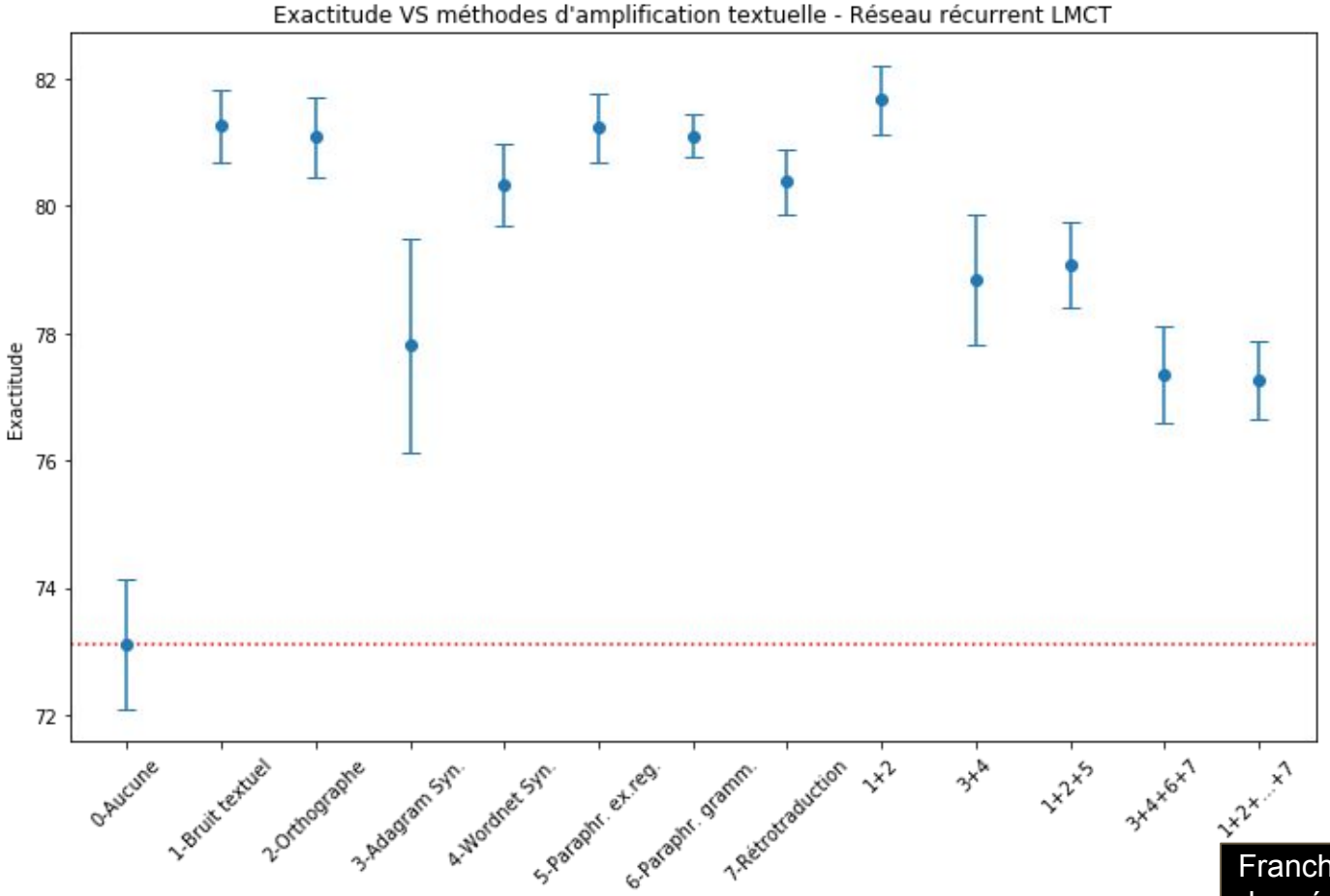
Franchir le « mur des données massives »

Résultats - Réseau convolutif 1D

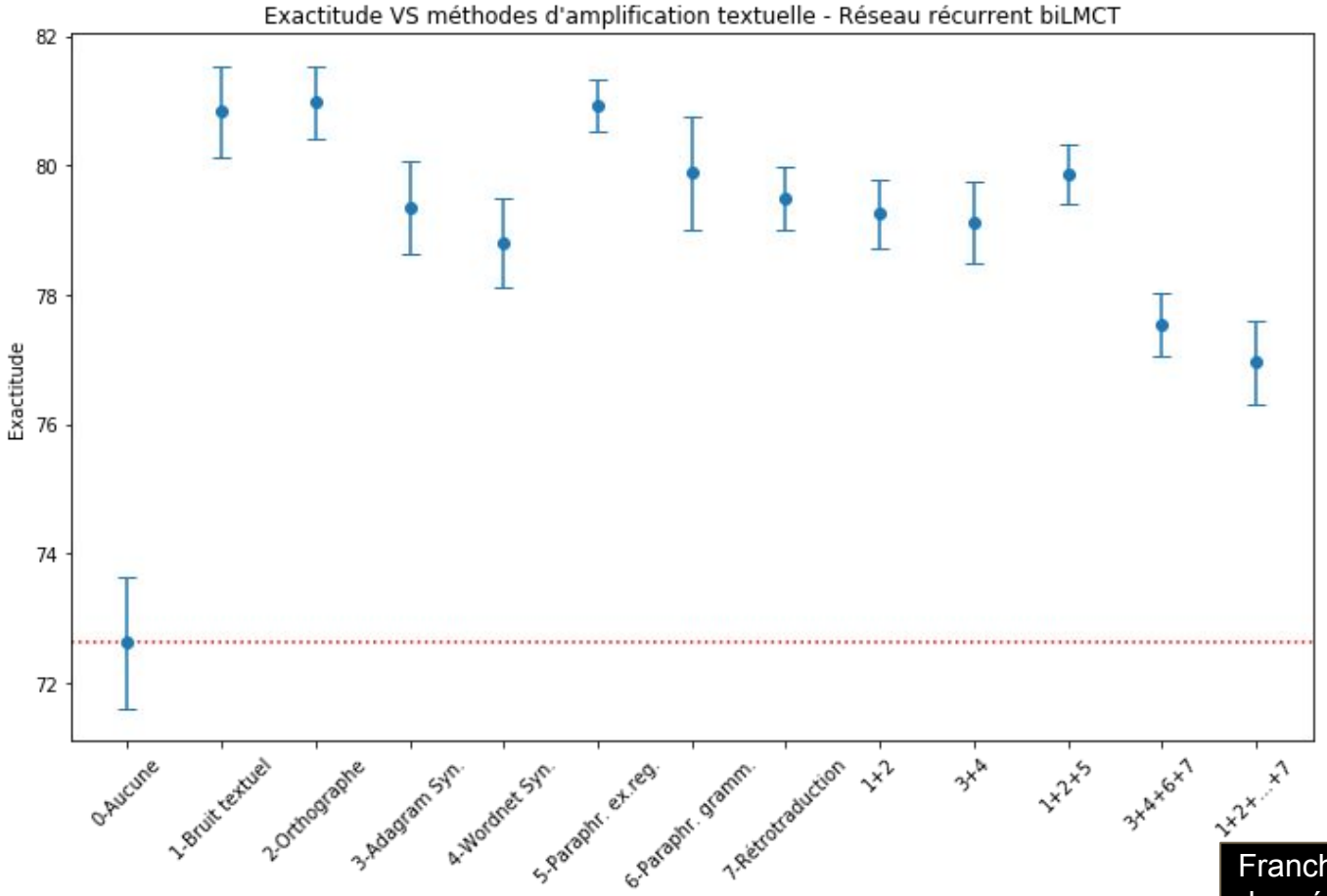


Franchir le « mur des données massives »

Résultats - Réseau récurrent LMCT



Résultats - Réseau récurrent biLMCT



Franchir le « mur des données massives »

Analyse - Avantages / Inconvénients

Les principaux avantages des techniques d'augmentation de texte qui viennent d' être brièvement explorées sont d'un point de vue pratique. Ils sont faciles à mettre en œuvre et à utiliser.

Tirer parti des services en ligne de traitement du langage naturel de fournisseurs bien établis offre de nombreux avantages concrets et immédiats: disponibilité, robustesse, fiabilité, capacité de monter en charge. En outre, il existe des solutions bon marché, prêtes à l'emploi et disponibles dans un grand nombre de langues.

Le principal inconvénient des techniques d'amplification textuelle explorées demeure la quantité de traitement et leur aspect massif. Aussi, plusieurs dépendent de services en ligne comme les services de traduction et d'analyse syntaxique qui sont loués à des entreprises privées.

Le mur des « modèles pré-entraînés »

Un nouveau mur se dresse à l'horizon, le « mur des modèles pré-entraînés ». En effet, l'emploi des gros modèles de langue pré-entraînés comme BERT de Google, Transformers de AI2 , ELMo de OpenAI (dont le controversé modèle génératif GPT-2), et ULMFiT de Fast.ai pourrait devenir le nouvel état de l'art.

Ce mur est plus difficilement surmontable car il combine à la fois la masse des données et la masse des calculs. La mise au point de gros modèles pré-entraînés est hors de portée de quiconque ne dispose pas d'énormes corpus (milliards de mots) et d'importantes infrastructures de calcul.

L'importance de disposer de gros modèles génériques pré-entraînés pour un maximum de langues et distribués selon des licences libres est un enjeu stratégique pour toutes les communautés linguistiques et la démocratisation de l'IA .

Franchir le « mur des données massives »

Conclusion

Ce travail empirique, mené avec des ressources informatiques limitées, a montré différentes techniques d'amplification des données textuelles simples, pratiques et faciles à mettre en œuvre.

Beaucoup de travail reste à faire pour explorer chaque technique d'amplification de manière plus détaillée, faire varier les paramètres d'amplification et les combiner. La poursuite de ces travaux exigera l'accès à une infrastructure de calcul équipée de processeurs graphiques.

Enfin, un travail de diffusion s'impose pour faire connaître ces techniques aux praticiens, ingénieurs et chercheurs à la recherche de solutions concrètes et pratiques pour franchir le *mur des données massives* dans l'application de l'apprentissage profond pour le traitement de la langue naturelle.

Article sur arXiv: https://arxiv.org/abs/1812.04718

Remerciements

Je profite de l'occasion pour remercier la TÉLUQ et l'UQAM où j'ai bénéficié (et continue de bénéficier) d'une grande liberté et de beaucoup de flexibilité dans le cadre du programme de Doctorat en informatique cognitive.

Un merci particulier à mon directeur de recherche M. Gilbert Paquette (IA et éducation), et mes deux codirecteurs Mme Neila Mezghani (science des données) et M. Michel Gagnon (TALN) de la Polytechnique Montréal.

Un clin d'oeil à Antoine Saucier Polytechnique Montréal et à Marie Bourdon, linguiste informaticienne, chez Coginov inc Montréal.

Merci également à MITACS et Coginov pour mon stage en entreprise.

Claude COULOMBE
scientifique des données
candidat au doctorat en informatique cognitive