Big BERT : Inférence en langue naturelle par enrichissement syntaxique

Raphael Croteau, François Vidal, Clément Maurel

Département Génie Informatique, Polytechnique de Montréal, Montréal, Canada

I. Introduction

Le problème de l'inférence en langue naturelle consiste à déterminer la relation liant deux phrases données. Cette relation peut être une relation d'implication : la deuxième phrase est une conséquence logique de la première, une contradiction : la deuxième phrase doit être fausse si la première est vraie, ou neutre : la deuxième phrase est indépendante de la première.

Exemples :

Le grand oiseau s'envole	**Contradiction**	Le volatile s'est endormi
La voiture démarre	**Implication**	Le moteur est allumé
Le grand oiseau s'envole	**Neutre**	La voiture démarre

L'inférence est un sujet central de l'intelligence artificielle, qui permet de déterminer si une hypothèse (phrase) *b* peut, à juste titre, être déduite d'une prémisse *a*. L'accent est mis sur le raisonnement informel, la connaissance sémantique lexicale et la variabilité de l'expression linguistique. Nous avons travaillé avec le corpus Stanford Natural Language Inference (SNLI) qui contient environ 550k paires de phrases annotées manuellement (en anglais) pour une classification équilibrée entre les étiquettes implication, contradiction et neutre. Le corpus sert à la fois à entraîner et à tester notre modèle.

Le modèle BERT, pour Bidirectional Encoder Representations from Transformers (Devlin et al., 2015), est un modèle de représentation du langage pouvant être appliqué à l'inférence. Les choix d'architecture du modèle ont une influence sur la précision qui peut dépasser 90% pour la base SNLI (Liu et al., 2019). La principale innovation technique de BERT consiste à appliquer la transformation bidirectionnelle de Transformer, un modèle d'attention, à la modélisation du langage. Après avoir essayé plusieurs modèles simples, nous avons développé le nôtre à partir du modèle DistilBERT, plus léger et plus rapide que le modèle BERT (Sanh et al., 2019). Les résultats sont présentés plus bas.

II. Méthodologie

Définition de la tâche/objectif

Notre objectif est de prédire le plus fidèlement possible le lien (implication, contradiction ou neutre) entre deux phrases de notre corpus. Pour entraîner et tester notre modèle, on a découpé le corpus SNLI contenant 510,705 paires de phrases comme suit : 85% des données pour l'entraînement, 10% pour l'ajustement et 5% pour la validation. On prend en entrée des paires de phrases en langue anglaise naturelle et on retourne la prédiction du lien entre les paires de phrases de nos données de validation.

Sommairement, nous avons comparé plusieurs modèles simples sur un échantillon de 10,000 paires de phrases afin de choisir le modèle le plus performant sur lequel bâtir le nôtre. Résultats obtenus pour 10,000 échantillons et 4 epoch :

Tableau 1. Performance des quelques modèles retenues sur 10'000 données

Embeddings BERTBASE + FeedForward	0.343
Embeddings $BERT_{BASE}$ + BiLSTM	0.663
$DistilBERT_{BASE}$	0.751
$DistilBERT_{BASE}$ + Etiquettes + FeedForward	0.784

Sur de petits échantillons, c'est DistilBERT avec étiquettes syntaxiques et feedforward qui semble le plus

 © xxxx IOP Publishing Ltd

performant. Nous nous concentrerons sur ce modèle pour la suite.

Définition de l'algorithme / méthode / technique

Afin de rendre nos données compatibles avec notre modèle, nous transformons les paires de phrases en un format compatible avec Bert, c'est-à-dire une liste d'identifiants, de la manière suivante :

1. On concatène les deux phrases en une seule, en les séparant par un jeton de classification [SEP] et en précédant la phrase d'un jeton [CLS].
2. On tokenize avec les tokenizer de la librairie (Tokens et WordPiece)
3. On transforme la liste de tokens en une liste d'identifiants (Bert Ids).

Les étapes sont résumées dans la Figure 1.

Nous souhaitons ajouter un étiquetage morpho-syntaxique (associer aux mots de la phrase les informations grammaticales correspondantes) à notre modèle. Exemple : « Nous sommes allées en Bretagne » pourrait être étiqueté « Nous/ PRO:PER sommes/ VER:pres allées/ VER:pper en/ PRP/ en Bretagne/ NAM ». Nous utilisons pour cela la fonction pos_tag de la bibliothèque NLTK. Comme Bert peut produire plusieurs tokens pour chaque mot et qu'il ne peut y avoir qu'une étiquette morphosyntaxique par mot, il nous faut faire la correspondance entre cette étiquette et les multiples tokens (Bert tags). Enfin, nous convertissons les étiquettes textuelles en un format numérique (« NN » → « 8 »), puis appliquerons une normalisation sur la matrice résultante (afin de réduire les chances que le modèle forme des relations du type "plus c'est grand, mieux c'est") .
Comme exemple, prenons les deux phrases "Wet brown dog swims towards camera." et "The dog is sleeping in his bed." qui sont des contradictions. Le prétraitement est donné, dans l'ordre, dans la Figure 2.
Cette dernière ligne est ensuite donnée à BERT. Nous prendrons la dernière couche cachée de celui-ci, appliquerons un "Average Pooling" afin de réduire la couche en un vecteur unidimensionnel puis concaténerons ce dernier avec le vecteur de tags qui sera donné à la couche de MLP.

Spécificité du modèle

Afin de combiner la sortie de Bert aux étiquettes syntaxiques, on a mis en place une fonction de fusion, qui est au centre de notre méthodologie. L'intuition guidant notre travail est que, sachant que BERT ne possède pas de mécanique explicite lui permettant de prendre en compte la syntaxe pour la classification, l'ajout de cette information pourrait améliorer les performances du modèle. Il suffit alors d'ajouter cette information à la sortie de Bert et d'entraîner un simple feed-forward dessus :

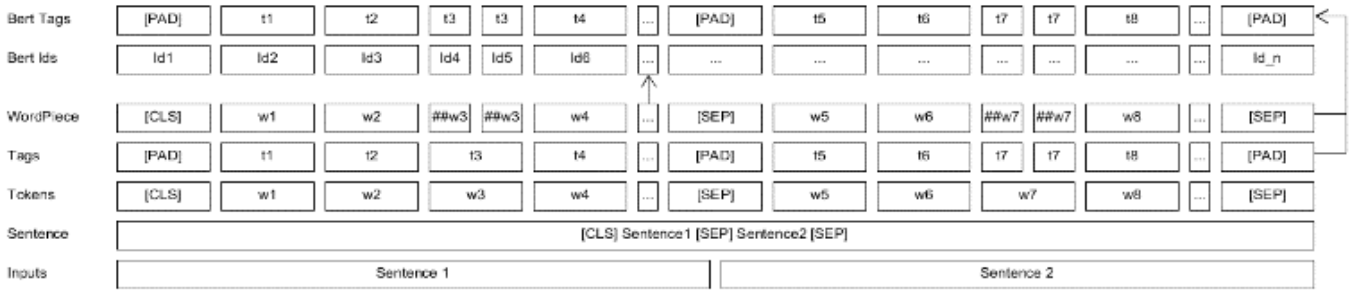
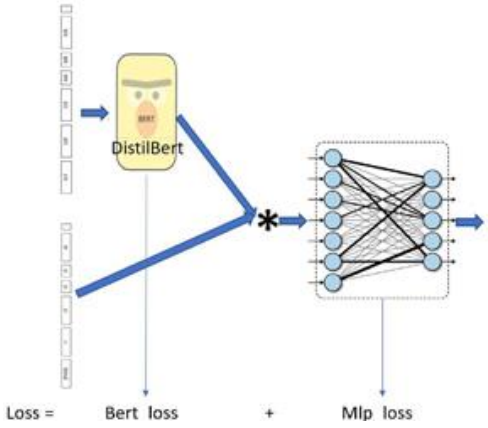


Figure 1 : pipeline de pré-traitement

Tokenize	'[CLS]', 'wet', 'brown', 'dog', 'swim', '##s', 'towards', 'camera', '.', '[SEP]', 'the', 'dog', 'is', 'sleeping', 'in', 'his', 'bed', '.', '[SEP]'
Tags	'[PAD]', 'WRB', 'JJ', 'NN', 'NN', 'NNP', 'NNS', 'NN', '[PAD]', 'VB', 'DT', 'NN', 'VBZ', 'VBG', 'IN', 'PRP$', 'NN', '[PAD]'
Wordpiece Tags	'[PAD]', 'WRB', 'JJ', 'NN', 'NN', 'NNP', 'NNS', 'NN', '[PAD]', 'VB', 'DT', 'NN', 'VBZ', 'VBG', 'IN', 'PRP$', 'NN', '[PAD]'
Normalized Bert Tags	'1.00, 0.91, 0.18, 0.27, 0.27, 0.27, 0.36, 0.45, 0.27, 1.00, 0.64, 0.00, 0.27, 0.82, 0.73, 0.09, 0.55, 0.27, 1.00, 0.00...'

Figure 2 : pipeline des étiquettes morphosyntaxiques



La fonction de fusion fait donc le lien entre la sortie de Bert et le vecteur des identifiants des tags. Nous avons testé plusieurs fonctions simples pour trouver la meilleure fonction de fusion :

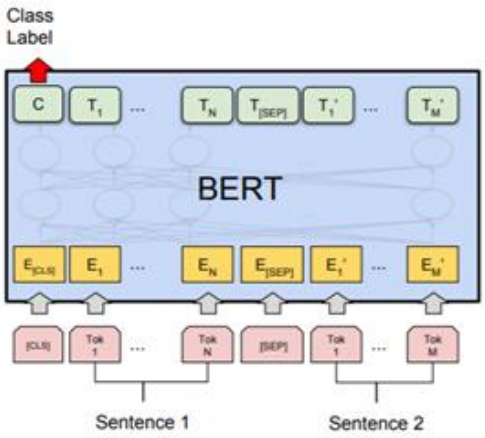
Tableau 2. Performance des fonctions retenues sur 10,000 données

elem à elem *	elem à elem +	elem à elem /	concat
0.774	0.353	0.347	0.784

Les résultats sont obtenus pour 10,000 échantillons et 4 epochs. On constate que la concaténation des deux vecteurs est la plus efficace, ce sera donc la fonction de fusion que nous adopterons pour la suite.

Mise en place du modèle

Nous avons choisi le modèle DistilBert Base pour sa grande vitesse d'exécution permettant des itérations rapides, avec une perte de puissance de seulement 3% et une augmentation de la vitesse d'entraînement de près de 60%.



Les nœuds de sortie correspondent aux nœuds intrants associés et la fonction de perte (loss) est calculée à partir de la sortie du jeton [CLS].

Le modèle BERT utilisé est un modèle pré-entraîné de la librairie Transformers de HuggingFace. Cette librairie est basée sur PyTorch, qui sera donc la base de nos modèles. Le modèle sera donc un module PyTorch liant ensemble une instance Bert et un MLP.

III. Evaluation expérimentale

Notre méthodologie était basée sur quelques hypothèses qui ont été abordées brièvement plus tôt. La première hypothèse est que, comme BERT ne possède pas de mécanique explicite pour prendre en compte la syntaxe d'une phrase (bien que cette dernière soit prise en compte implicitement), il serait intéressant d'ajouter cette information à la sortie de BERT et d'entraîner un simple multi-level perceptron sur les vecteurs résultants. La deuxième hypothèse que nous avions est que cette information pourrait ensuite être jumelée avec des étiquettes sémantiques pour offrir un plus grand pouvoir discriminant.

Afin de prouver ou réfuter ces hypothèses, nous avons décidé de mettre en place un modèle PyTorch en trois étapes, soit une étape BERT, une étape de fusion BERT+PoS et une étape MLP. Le modèle serait ensuite entraîné en faisant de la rétropropagation sur la somme du calcul de perte du MLP ainsi que de BERT. Ce dernier point à été décidé en considérant que BERT est déjà excellent dans la tâche de classification et que sa sortie à l'index du jeton [CLS] est du même fait très discriminante.

Afin d'évaluer les différents modèles ou les différents hyperparamètres, la métrique du pourcentage de prédiction réussie s'est imposée (soit la mesure de précision).

Les données utilisées pour l'entraînement étaient simplement le corpus du SNLI. Comme le corpus est balancé (170271 données pour neutral, 170316 données pour entailment et 170118 données pour contradiction), il n'est pas nécessaire de faire une sélection par rapport à cette classe. Les données du SNLI sont très intéressantes de par leur nombre et de par leur qualité. Leur nombre, car il est très rare de trouver des bases de connaissances textuelles pour l'apprentissage supervisé de cinq cent mille données. Leur qualité, car chaque entré a été analysée par cinq personnes différentes pour déterminer sa classe.

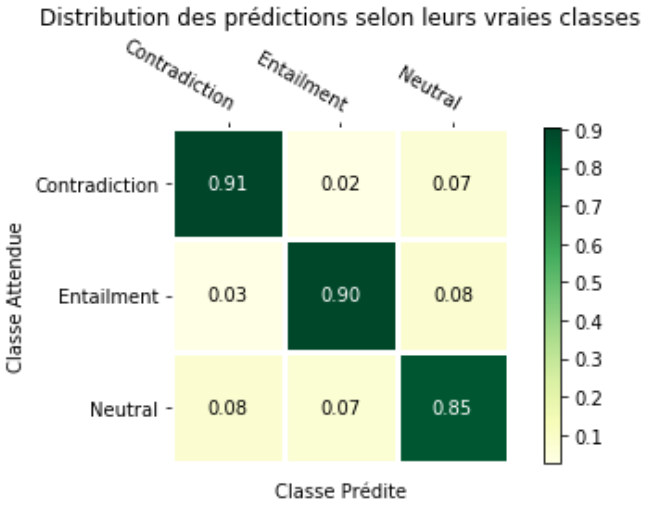
Notre méthodologie est inspirée de SemBert (Zhousheng Zhang et al. '19b), soit un modèle basé sur l'enrichissement de BERT par le SRL (Semantic Role Labeling). Alors que ce modèle utilise des étiquettes SRl, le nôtre se base sur les étiquettes morphosyntaxiques. Alors que SemBert utilise un CNN comme dernière étape dans leur modèle, le nôtre utilise un simple MLP. Comme la sémantique possède beaucoup plus d'information en ce qui a trait aux liaisons entre les phrases, il est attendu que ce modèle ait de meilleures performances. Cependant, le SRL est un processus extrêmement lourd, pouvant prendre jusqu'à 1.5 seconde pour analyser une entrée, alors que l'étiquetage morphosyntaxique est un processus très rapide à environ 0.00088 seconde par entrée. D'autres études ont réussi à extraire de bonnes performances à l'aide d'entraînement semi-supervisé (Xiaodong Liu et al. '19). Cette méthodologie, comme BERT, se base sur le transfer learning, soit le préapprentissage non supervisé d'un modèle basé sur le transformeur ainsi que sur des données non étiquetées. Ce modèle est ensuite fine-tuned sur quatre grands problèmes du TAL simultanément. Notre méthodologie suit un processus similaire, étant basée sur une architecture transformeurs, mais nous nous intéressons uniquement à la tâche de l'inférence.

Résultats

Tableau 3. État de l'art vs nos résultats

Méthodes	Test Acc	# Parameters
300D DMAN Ensemble (Boyuan Pan et al. '18)	89.6	79m
SemBert (Zhousheng Zhang et al. '19b)	91.8	340m
Embedding BertBase w/ BiLSTM	78.0	
DistilBertBase (baseline)	86.6	67m
DistilBertBase + PoS + FeedFoward	88.7	67m

Notre modèle est donc significativement meilleur que le baseline, mais est très loin de l'état de l'art en ce qui est de l'inférence. En utilisant le test A/B, la différence est statistiquement significative.



Discussion

Notre hypothèse est vérifiée, l'ajout d'information syntaxique à la sortie de BERT permet une amélioration du pouvoir prédictif du modèle sur la tâche de l'inférence. Cependant, la syntaxe est moins discriminante que la sémantique, ce qui créer un premier creux entre SemBert et notre méthode. De plus, DistilBert est moins performant que Bert Large, ce qui créer un deuxième creux. Pour ce qui est des forces, selon nos expériences, notre modèle est cependant significativement plus rapide autant au niveau du prétraitement des données que de la prédiction comparément aux deux modèles connexes que nous avons analysés, en plus d'être significativement plus léger en mémoire (voir tableau 3).

IV. Travaux connexes

Le problème de l'inférence en langue naturelle fait partie des problèmes les plus étudiés en TAL. Et le corpus SNLI est l'un des principaux corpus d'entraînement pour ce problème. Depuis la publication du corpus en 2015, de nombreux travaux se sont intéressés à la résolution de ce problème et à l'optimisation de résultats précédents. L'arrivée récente de BERT a permis une amélioration nette des résultats pour de nombreux problèmes de TAL, parmi lesquels le NLI. Notre travail s'inscrit donc, en partie, dans la lignée des approches récentes s'appuyant sur un modèle BERT, et plus largement des approches s'appuyant sur des plongements lexicaux issus de modèles profonds.

Les auteurs de (Qian Chen et al. '16) présentent une approche s'appuyant sur deux modèles : un biLSTM et un LSTM ayant une structure d'arbre qui permet de considérer l'arbre syntaxique de la phrase. Cette approche montre l'impact positif de la prise en compte

d'informations sur la syntaxe sur les résultats en NLI. Cependant, l'approche présentée est moins efficace que des approches récentes basées sur des modèles plus complexes et les possibilités d'amélioration en se limitant à des modèles LSTM semblent limitées.
L'approche introduite plus haut sous le nom de SemBERT, et qui possède l'un des meilleurs scores présentés sur le site du SNLI, a montré que l'utilisation de données sémantiques (SRL) avait une influence positive sur les résultats de NLI. Cependant, le problème de SRL étant loin d'être trivial, le modèle SemBERT s'avère être beaucoup plus coûteux en temps et en ressources.

V. Travaux futurs et conclusion

Comme dit précédemment, les étiquettes syntaxiques ont intrinsèquement un moins grand pouvoir discriminateur que, par exemple, les étiquettes sémantiques. Il serait cependant intéressant de jumeler ces deux catégories d'étiquettes afin de voir si elles ont une relation synergique permettant de meilleures prédictions.

Pour aller dans cette direction, nous avons ajouté à notre modèle des labels sémantiques obtenus avec la librairie AllenNLP. Cependant, à cause des délais importants pour l'obtention de ces tags, nous n'avons pas pu les utiliser sur le corpus entier. Nous avons cependant entrainé un modèle Bert associé aux tags sémantiques sur un nombre réduit de samples et les résultats semblent indiquer que cette méthode produit bien une amélioration des performances du modèle. Nous rapportons ci-dessous les résultats obtenus : on constate une amélioration de plus de 1% avec le SRL par rapport au modèle de base.

Tableau 4. Résultats sur 25.000 samples

Method	Train Acc	Test Acc	# Parameters
Bert Base	97.2	79.3	67m
Bert Base + SRL (1 Layer)	98.5	80.5	67m
Bert Base + SRL (2 Layer)	95.9	77.4	67m

Une autre limite est que notre modèle prédit beaucoup mieux les classes "entailment" et "contradiction" que "neutral". Cette dernière étant aussi responsable du plus grand taux de faux négatifs dans les autres classes (e.x. le modèle prédit neutral au lieu de entailment). Il serait intéressant d'augmenter la proportion de données étiquetées comme étant neutral, soit en en créant des nouvelles ou en diminuant le nombre de données des autres classes. Il pourrait aussi être intéressant d'analyser le corpus des faux négatifs X-Neutral pour déterminer si certaines structures syntaxiques ou certains mots provoquent ceux-ci.

En bref, notre méthode permet une classification tout de même acceptable sur la tâche de l'inférence, mais avec une grandeur moindre en mémoire que les modèles concurrentiels aussi basés sur des transformeurs. De plus, notre hypothèse sur l'utilisation des étiquettes syntaxiques a été validée. Bien que cette intuition soit plutôt simple, le fait que son utilisation soit rapide pourrait permettre de former des modèles plus puissants l'utilisant avec d'autres mécaniques.

Références

Xiaodong Liu et al. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. arXiv preprint arXiv:1901.11504v2

Zhousheng Zhang et al. 2019. Semantics-aware BERT for Language Understanding. arXiv preprint arXiv:1909.02209v2

Boyuan Pan et al. 2018. Discourse Marker Augmented Network with Reinforcement Learning for Natural Language Inference. ACL, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), p 989–999.

Qian Chen et al. 2016. Enhanced LSTM for Natural Language Inference. arXiv preprint arXiv:1609.06038v3

D. Jurafsky, J.H. Martin 2019. Speech and Language Processing. Pearson.

Devlin et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805v2

Liu et al., 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692

Sanh et al. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108v2