École Polytechnique Montréal
December 9, 2019

INF8460 Projet final

Inférence en langue naturelle

J.N. Demanche, G. Dervaux, M. Menghetti

ABSTRACT

Contexte. Le traitement de la langue naturelle est une des disciplines en plein essor. Récemment, des modèles de plus en plus complexes permettent de comprendre automatiquement des textes écrits par des humains. Différentes tâches sur différents corpus permettent d'évaluer les capacités de nos modèles à comprendre la langue naturelle. Parmi ces tâches, on retrouve notamment l'inférence en langue naturelle, et bien d'autres.
Objectif. Dans le cadre de ce projet, nous allons prédire la relation d'inférence entre deux phrases. La seconde phrase est-elle en contradiction avec la première ? En est-elle une conséquence logique ? Ou bien est-elle neutre ? Il s'agit là d'un problème de classification d'un doublet de phrases en 3 catégories.
Méthodes. Les modèles statistiques ont longtemps été privilégiés pour résoudre ces problèmes mais, très récemment, les réseaux de neurones transformeurs sont les plus efficaces. Nous avons réalisé plusieurs modèles, en commençant par une baseline, le modèle ayant donné les meilleurs résultats lors des labs (MLP avec dropout). Nous avons ensuite utilisé BERT-Base, XLNet-Base et certaines architectures on-top of XLNet qui donnent les meilleurs résultats. Nous avons notamment tenté l'augmentation de données, la stratégie multi-tâches, et des méthodes d'ensemble.
Résultats. Nos résultats s'approchent de l'état de l'art sur le dataset utilisé. Notre meilleure soumission, correspondant à la méthode d'ensemble utilisant BERT, XLNet et notre méthode XL-BiSEPs, a une accuracy de 90,91%, là où l'état de l'art obtient 91,6%.

1. Introduction

Le traitement de la langue naturelle est une discipline informatique en plein essor où les progrès en apprentissage machine permettent de s'attaquer à des problèmes de plus en plus complexes. La détection d'inférence en langue naturelle est une des tâches permettant d'évaluer la compréhension de la langue naturelle par un modèle. Le principal objectif en détection d'inférence est de déterminer le lien unissant deux phrases. Il peut s'agir d'une contradiction, d'une conséquence logique ou d'une relation neutre. Un modèle entraîné sur cette tâche pourrait alors prédire le lien unissant deux phrases et déterminer si deux séquences sont en contradiction ou non. Il serait également capable de générer une nouvelle séquence présentant le lien désiré avec une autre phrase, ce qui pourrait être d'une grande utilité pour les agents intelligents.
Dans le cadre de ce projet, plusieurs modèles ont été mis à l'épreuve dans le but d'obtenir la meilleure accuracy. Parmi les modèles testés, on retrouve un modèle servant de baseline (multilayer perceptron prenant en entrée les occurrences des mots dans chacune des deux phrases), un modèle BERT et un modèle XLNet. Ces deux derniers, figurant parmi les modèles de transformeurs ayant fait d'importants progrès dans les dernières années, sont des outils très prometteurs quant à l'analyse de la langue naturelle. En effet, les travaux récents indiquent que ces modèles peuvent atteindre des scores très impressionnants, entre autres grâce à leur approche bidirectionnelle et leur concept d'attention. C'est d'ailleurs pour cette raison qu'ils ont figuré parmi les outils utilisés afin de s'attaquer au problème de l'inférence en langue naturelle, qui nécessite une compréhension profonde du contexte.

2. Définition du problème

2.1. Tâche et objectif

La tâche à accomplir est la classification de paires de phrases selon le lien les unissant. Le corpus utilisé est le Stanford Natural Language Inference (SNLI) Corpus, présenté dans [3]. Il s'agit d'un groupement de plus de 500 000 paires de phrases accompagnées d'un lien les caractérisant. Un échantillon du dataset est visible (Fig. 1). Le lien peut être un lien de conséquence (entailment), de contradiction ou de neutralité (neutral).

150651 unique values	447545 unique values	entailment 33% neutral 33% Other (1) 33%
A person on a horse jumps over a broken down airplane.	A person is training his horse for a competition.	neutral
A person on a horse jumps over a broken down airplane.	A person is at a diner, ordering an omelette.	contradiction
A person on a horse jumps over a broken down airplane.	A person is outdoors, on a horse.	entailment
Children smiling and waving at camera	They are smiling at their parents	neutral
Children smiling and waving at camera	There are children present	entailment
Children smiling and waving at camera	The kids are frowning	contradiction

Train: **511k** paires de phrases
Test: **10k** paires de phrases

Fig. 1. Résumé du dataset SNLI Stanford (2015)

Ces relations ont été établies par plusieurs personnes afin de limiter le risque d'erreur. En effet, lors de la réalisation du corpus, plusieurs personnes ont dû prédire le lien entre les deux phrases, dont l'auteur du doublet de phrases. Certaines phrases obtiennent leur label à l'unanimité, tandis que d'autres sont plus complexes, même pour des humains. Les auteurs du corpus ont donc introduit le concept de "gold label", qui représente le cas d'un label prédit correctement par la majorité, tout en étant identique au label donné par l'auteur. De plus, on peut remarquer que les phrases 1 se répètent plusieurs fois avec des phrases 2 différentes, amenant à différents liens. De plus, les trois catégories sont équilibrées et contiennent le même nombre d'exemples.
L'objectif de ce projet est de maximiser la "categorisation accu-

racy", c'est-à-dire le pourcentage de phrases classifiées correctement.

2.2. Une première baseline : Multilayer Perceptron (MLP)

Notre premier modèle, utilisé à titre de baseline, est un modèle perceptron multicouches. Il s'agit d'un réseau à trois couches avec dropout entre chacune d'elles. Nous avons récupéré les occurrences des mots dans chacune des phrases d'entrée, cela nous donne une entrée de type Bag-of-Words pour chacune des phrases. L'entrée du modèle est alors une concaténation des deux entrées de type Bag-of-Words, ayant une taille de deux fois la taille du vocabulaire.
L'ensemble de validation a ensuite servi à ajuster les paramètres comme le nombre de couches et le dropout afin de maximiser l'accuracy résultante. La sortie du modèle est le résultat d'une fonction softmax qui attribue une probabilité à chacune des trois classes possibles. La classification résultante est celle dont la probabilité est la plus élevée.

3. Travaux connexes

Les travaux connexes représentent principalement les architectures "transformeurs" basées sur le concept d'attention. Ces architectures sont complexes et possèdent un nombre énorme de paramètres. Les paramètres du modèle après un pré-entraînement sont disponibles, il s'agit de transfer learning. Les modèles pré-entraînés sur un très grand nombre de textes permettent d'avoir une bonne base de départ. Il faut ensuite fine tuner les modèles avec des données en lien avec la tâche à accomplir et la sortie désirée. Nous allons utiliser le modèle BERT base, et le modèle XLNet base.

3.1. Modèle BERT base

Notre baseline (MLP) comprenait plusieurs failles telles que la distinction entre les deux phrases et la connaissance du contexte. Pour cette raison, nous avons ensuite opté pour un modèle pré-entraîné de type BERT tel que présenté dans [1]. Ce dernier consiste en une pile de transformeurs qui permettent une meilleure compréhension du contexte grâce à l'attention portée sur chacun des mots, ainsi que la bidirectionnalité du modèle. De plus, BERT a été pré-entraîné sur l'ensemble de Wikipédia, ce qui lui confère une connaissance du langage beaucoup plus grande que le modèle perceptron créé précédemment.
En guise d'entrée, la paire de phrases a été subdivisée grâce à un token [SEP] placé à la fin de chaque séquence. Ceci a rendu plus claire la distinction entre les deux et permis au modèle BERT de performer beaucoup mieux que le modèle MLP. La classification résultante est attribuée à la sortie correspondant au token [CLS] (placé au tout début de l'entrée). Cette sortie, de même qu'avec le MLP, est une distribution de probabilité d'appartenir à chacune des classes. La probabilité la plus élevée détermine la classe choisie.
Tout comme le modèle MLP, un ensemble de validation a été utilisé afin de jouer sur les paramètres et ainsi maximiser l'accuracy.
Cependant, le modèle BERT possède certaines failles. Notamment, il a été entraîné avec des tokens [MASK] qui ne sont pas présents lors de la tâches qu'on lui donne à faire. De plus, Bert réalise des prédictions de mots indépendantes dans une même phrase, alors que ces prédictions devraient être dépendantes entre elles pour maximiser les chances d'avoir un sens plausible dans la phrase en sortie.

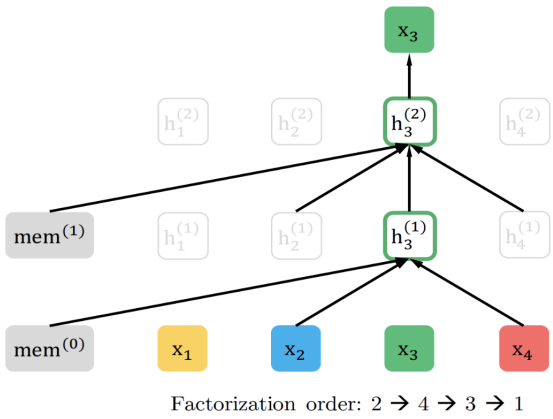


Fig. 2. Modèle de langue à permutation avec XLNet

3.2. Modèle XLNet base

Le modèle XLNet tel que présenté dans [2], s'appuie sur les faiblesses de BERT pour proposer une meilleure architecture et un meilleur pré-entraînement.
Contrairement à BERT qui utilise des [MASK] et des prédictions de mots indépendantes, XLNet s'entraîne de façon dépendante à l'aide d'un modèle de langue à permutation, comme on peut le voir sur la Fig. 2.

Le modèle de langue à permutation est une méthode permettant la bidirectionnalité. XLNet choisit une permutation des termes de la phrase d'entraînement de manière aléatoire. Cette permutation permet de connaître les mots faisant partie du contexte. Le contexte est l'ensemble des mots connus de la phrase qui aident à compléter le mot suivant. Dans l'exemple, la permutation choisie est 2, 4, 3, 1. Au moment de s'entraîner à deviner le mot 3, le modèle aura accès aux mots 2 et 4. Le modèle aura également accès à une mémoire. Celle-ci représente les informations à long-terme de la phrase et provient du modèle Transformer-XL. Cette mémoire permet de gérer les très longues entrées et de mieux comprendre les dépendances à long-terme.

4. Nos modèles

Cette partie regroupe les modèles que nous avons tentés afin d'obtenir de meilleurs résultats que par une simple utilisation de modèles pré-entraînés qui a permis d'obtenir des résultats à des fins de comparaisons (notamment à l'aide de [5]). Il s'agit ici de notre apport lors de ce projet.
Les modèles présentés sont en grande partie différentes approches et architectures "on-top" of XLNet base.
Finalement, différents modèles ont été combinés afin de créer un modèle d'ensemble et garder les meilleures prédictions de chacun des modèles utilisés.

4.1. Modèle XL-BiSEPs

Suite à la présentation de Michel Gagnon en cours sur le modèle "Matching the Blanks" [4], nous avons pensé à une architecture "on-top" of XLNet similaire. L'architecture de base récupère la sortie du token <cls>, réalise un feed-forward fully connected, puis une classification à l'aide de la fonction softmax. Au lieu de cette architecture, nous réalisons une architecture différente

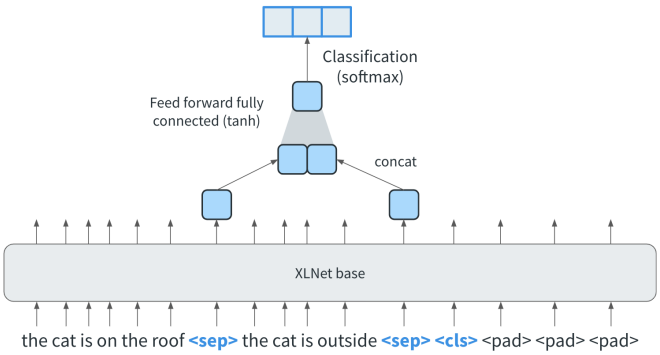
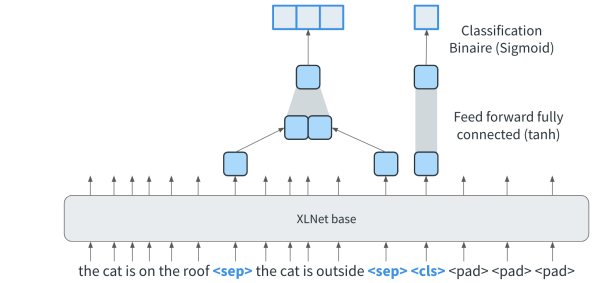


Fig. 3. Architecture du modèle XL-BiSEPs



Pour toute *"sentence1"* on rajoute une *"sentence2"* au hasard avec comme label 'neutre' et 0 pour le deuxième objectif. Les *"sentence2"* originales ont un label 1 pour le deuxième objectif pour signifier qu'elles parlent du même sujet.

Exemple :

I like snow	*I love christmas*	⇒ Neutral, is_linked
The cat is on the roof	*The cat is outside*	⇒ Entailment, is_linked
The cat is on the roof	*I love christmas*	⇒ Neutral, is_not_linked ⇒ **générée**

Fig. 4. Architecture du modèle XL-BiSEPs avec un second objectif (Augmentation de données)

visible à la Fig. 3. Nous récupérons les sorties des deux tokens <sep>. Ces deux sorties vont tenter de représenter chacune des phrases d'entrée. Nous allons ensuite les concaténer, et réaliser le feed-forward habituel vers une dimension réduite de moitié. Finalement la classification est réalisée avec un softmax.
Note : Lors de la présentation du poster, nous avons nommé cette approche "<sep>-concat", et l'avons par la suite renommée "XL-BiSEPs" (pour XLNet bi <sep>).

4.2. Modèle XL-BiSEPs avec 2nd objectif et augmentation de données

Nous avons remarqué que les phrases A et B avaient en général le même sujet. Comme on peut le voir dans les phrases d'exemple de la Fig. 4, le premier doublet de phrases aborde le thème de l'hiver, tandis que le second doublet est à propos d'un chat. En utilisant une phrase de chaque doublet, on obtient deux phrases n'ayant pas le même sujet, et ayant de grandes chances d'être neutres entre elles, comme le montre l'exemple de phrase générée à la Fig. 4. Nous réalisons ici une augmentation de données.

Pour chacune des données, nous rajoutons une information correspondant à "is_linked" ou "is_not_linked". Cela permet de savoir si les phrases parlent du même sujet ou non. Nous considérons les phrases présentes dans le dataset "is_linked", et les phrases générées "is_not_linked". Nous partons ici de l'hypothèse que tous les doublets de phrases présents dans le dataset parlent du même sujet et qu'il y a peu de chance, en prenant des paires de phrases de manière aléatoire, de tomber sur deux phrases traitant du même sujet.
De même qu'avec notre méthode XL-BiSEPs, nous réalisons une prédiction sur la catégorie du doublet de phrases. Nous rajoutons cependant un second objectif parallèle afin de prédire le champ "is_linked". Cette prédiction binaire est réalisée avec la sortie du token <cls>, comme le montre le schéma d'architecture de la Fig. 4.
Dans cette approche, toutes les données utilisent les deux objectifs de manière simultanée. Les deux objectifs sont pondérés, afin de représenter leurs importances respectives dans le problème. Dans le calcul du loss à minimiser, la prédiction du champ "is_linked" est dix fois moins importante que la prédiction de la catégorie d'inférence entre les deux phrases.

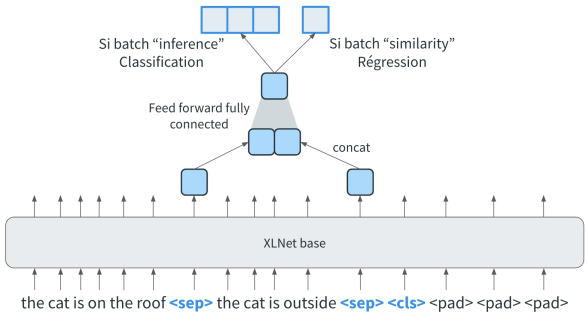


Fig. 5. Architecture du modèle XL-BiSEPs multi-tâches

4.3. Modèle XL-BiSEPs multi-tâches

En lisant le papier correspondant à l'état de l'art sur ce dataset (MT-DNN) [6], nous avons pensé modifier notre modèle précédent avec un second objectif, en utilisant à la place des données plus fiables, provenant d'un dataset vérifié, comme celui de l'autre projet. Le dataset de l'autre projet comprend environ 6 000 doublets de phrases et des valeurs de similarité entre ces phrases sur une échelle de 1 à 5. Les phrases de ce dataset semblent être composées du même type de vocabulaire. Nous pensons qu'entraîner notre modèle sur une autre tâche pourrait lui permettre de mieux généraliser et mieux comprendre les informations trouvées dans les phrases.
Cependant, contrairement à notre précédent modèle, nous n'avons pas de valeurs de similarité sur les phrases provenant du corpus d'inférence, et n'avons pas de catégories d'inférence pour les phrases provenant du corpus de similarité. Notre modèle multi-tâches choisit donc une formule de loss adéquate en fonction des données utilisées lors de chaque batch.
Nous avons également modifié l'architecture, dont un schéma peut être trouvé à la Fig. 5. Nous pensons que l'utilisation des sorties des deux <sep> pour les deux objectifs pourrait aider à mieux faire apprendre au modèle les représentations de chacune des phrases.

Certains batch sont des batches d'inférence, d'autres sont des batches de similarité. En fonction de la tâche, la formule du loss est différente, et les paramètres sont optimisés.
La majorité de nos données restent des données d'inférence compte tenu le peu de données dans le corpus de similarité de l'autre projet. Nous avons choisi d'entraîner notre modèle sur la similarité pour un batch sur 10.

4.4. Combinaison de modèles

Suite à une analyse de nos modèles précédents, un constat est survenu. Il devenait de plus en plus difficile de gagner en performance en créant de nouveaux modèles "on-top" of XLNet. C'est après cette réalisation que nous avons décidé de combiner les résultats de nos modèles précédents dans l'espoir que leurs forces se combinent afin de pallier aux faiblesses des autres.
Plusieurs approches ont été tentées, entre autres en combinant les sorties softmax des différents modèles et en traversant un autre réseau neuronal afin de trouver la classification la plus probable. Cependant, la combinaison qui s'est avérée la plus efficace a été parmi les plus simples. Les classifications des trois modèles les plus performants (BERT base, XLNet base et XL-BiSEPs) ont été comparées et la classification finale est tout simplement la prédiction majoritaire (le mode). Dans les cas où aucune catégorie n'est majoritaire (une prédiction dans chaque catégorie), le résultat choisi est celui du modèle XL-BiSEPs, qui a le mieux performé de façon individuelle. Cette simple étape de combinaison a permis gagner quelques 0,1% d'accuracy.

5. Évaluation

5.1. Ensemble de données, métriques et baselines

Les données utilisées sont un groupement de plus de 500 000 paires de phrases, chaque paire étant associée à une des trois relations possibles. Ce jeu de données présente l'avantage d'un volume très grand, ce qui facilite l'apprentissage d'un modèle grâce à la quantité de données sur lesquelles il sera entraîné. Cependant les modèles utilisés de type transformeurs sont très longs à entraîner (environ 2 à 3 heures par epoch avec GPU) et la taille des batches doit être réduite jusqu'à 32 éléments pour pouvoir tenir en RAM.
Aucune entité nommée n'est présente dans le corpus, ce qui aurait pu ajouter un niveau de difficulté supplémentaire. De plus, les phrases contenues dans le corpus présentent plusieurs situations et contextes différents avec un vocabulaire semblable. Il s'agit ainsi de phrases ordinaires susceptibles de survenir dans un discours quotidien et ne nécessitant aucun contexte externe. Les modèles entraînés pourront donc exclusivement se concentrer sur le contexte provenant de la paire de phrases, ce qui facilitera la tâche de classification.
Les phrases sont majoritairement composées de vocabulaire assez simple et sont "propres". Il ne s'agit pas par exemple de phrases récupérées sur Twitter nécessitant un preprocessing long et fastidieux. Les mots sont correctement orthographiés. Nous avons donc décidé de ne pas réaliser de preprocessing de manière générale. Dans les tâches d'inférence, nous pensons que les stopwords s'avèrent utiles. Notamment, les déterminants peuvent aider à déterminer des quantités, et les mots tels que "from" ou "to" peuvent faire changer le sens de la phrase et la classifier dans une autre catégorie d'inférence.
De plus, nous utilisons directement le tokenizer de Bert et celui d'XLNet afin d'obtenir des tokens en lien avec le vocabulaire de pré-entraînement des modèles.
Nous nous sommes rendus compte que certaines phrases étaient terminées par un "." et d'autres non. Nous avons donc uniformisé les phrases et retiré les points.
Une subdivision du corpus initial a été effectuée en un ensemble d'entraînement (constitué d'environ 500 000 paires de phrases) et en un ensemble de validation (composé de 10 000 paires tirées du même corpus initial). Ce dernier ensemble permettra l'ajustement des hyperparamètres afin d'optimiser le modèle.
Afin d'avoir une métrique de validation proche de celle du test, nous choisissons de prendre les valeurs de validation de manière consécutive dans notre dataset initial. En effet, en prenant des données de manière aléatoire, notre ensemble de validation contiendra principalement des valeurs de phrase A sur lesquelles il s'est déjà entraîné (puisque les phrases A sont répétées plusieurs fois).
La métrique utilisée par la compétition est la "categorisation accuracy". Nous avons également utilisé la matrice de confusion pour déterminer les erreurs d'identification les plus fréquentes commises par les modèles.
Le principal enjeu attaqué par la présente expérimentation est l'identification précise du lien entre les paires de phrases. Notre hypothèse principale est que les modèles pré-entraînés présentent des avantages qui leur permettront de mieux performer dans la tâche d'inférence, identifiant plus souvent la relation appropriée entre deux phrases. Ces modèles ont été entraînés sur de larges corpus, ce qui leur confère une meilleure connaissance du contexte des mots. De plus, nous croyons qu'une combinaison de plusieurs modèles différents pourrait mener vers une meilleure précision que chacun de ces modèles évalués individuellement. Il est également possible que l'augmentation de données et l'entraînement sur d'autres données pour d'autres tâches aident le modèle à obtenir une meilleure performance en lui permettant de mieux comprendre dans l'ensemble, et ainsi de mieux généraliser les informations trouvées dans les phrases.
Afin d'évaluer nos résultats, nous les comparons à une baseline simple (multilayer perceptron), ainsi qu'au modèles transformeurs non modifiés, et à l'état de l'art.
Notre baseline se base sur une matrice document-mot comme entrée de modèle. Les valeurs TF-IDF ont également été utilisées afin d'incorporer la rareté des mots dans les entrées du système, cependant l'utilisation de la pondération TF-IDF ne modifiait pas réellement notre performance avec ce modèle.

5.2. Résultats

Les valeurs d'accuracy de nos modèles perceptron, BERT, XLNet (base, BiSEPs, 2nd objectif) et modèles combinés figurent dans la Table 1.

On constate que notre baseline donne d'assez bons résultats, pour un nombre de paramètres assez faible, et un temps d'entraînement très rapide en comparaison des autres modèles utilisés.
Les modèles Bert et XLNet directement utilisés à l'aide des implémentations PyTorch donnent de très bons résultats. XLNet donne de meilleurs résultats que Bert, comme on avait pu s'y attendre après avoir étudié les faiblesses de Bert.
Concernant nos contributions, XL-BiSEPs est le seul qui obtient une performance supérieure à celle d'XLNet simplement utilisé. Le modèle multi-tâches se base sur des données ayant peut-être plus de sens que l'augmentation de données que nous avons tenté avec XL-BiSEPs double objectif. Il n'est donc pas étonnant que l'approche multi-tâches avec les données de similarité de phrases donne de meilleurs résultats.
Le modèle combiné utilise les deux modèles Bert et XLNet simplement utilisés, ainsi que notre meilleur modèle seul (XL-BiSEPs). Le modèle d'ensemble obtient la meilleure accuracy, légèrement supérieure à celle de XL-BiSEPs.
Comparativement à l'état de l'art, nos résultats sont honorables. En effet, nous ne disposons pas d'autant de ressources pour pouvoir faire tourner de plus gros modèles dans des temps raisonnables.
Nous avons de plus réalisé une étude statistique afin de comparer les résultats de nos modèles. Un modèle est-il significativement

Modèle	Accuracy (Test public Kaggle)
Baseline	
MLP 3 couches + Dropout	0.7026
Utilisation des travaux connexes	
Bert base uncased	0.8956
XLNet base cased	0.9020
Nos modèles	
XL-BiSEPs	0.9081
XL-BiSEPs + 2nd objectif (augmentation de données)	0.8861
XL-BiSEPs + multi-tâches	0.9003
Modèle d'ensemble (Bert + XLNet + XL-BiSEPs)	**0.9091**
État de l'art	
MT-DNN	0.916

Table 1. Résumé de nos résultats

	Bert	XLNet	XL-BiSEPs	Ensemble
MLP	< 0.01	< 0.01	< 0.01	< 0.01
Bert		0.145	< 0.05	< 0.01
XLNet			0.153	0.113
XL-BiSEPs				0.461

Table 2. Étude statistique de nos meilleurs modèles - p-values

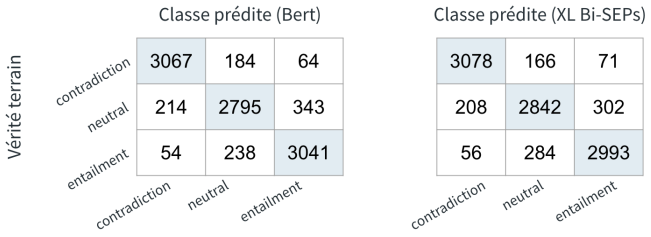


Fig. 6. Matrices de confusion de BERT base uncased et de notre modèle XL-BiSEPs

meilleur qu'un autre? Pour cela nous commençons par émettre l'hypothèse suivante : "Les modèles que nous étudions sont aussi bons les uns que les autres." Compte tenu de la taille du test public (environ 30% de 9 831 données prédites de soumission) et des accuracies de chacun de nos modèles, nous pouvons calculer les p-values permettant de savoir si l'hypothèse énoncée précédemment est valide ou rejetée. Lorsque la p-value calculée est inférieure à 0.05, l'hypothèse est rejetée et un des deux modèles est donc significativement meilleur que l'autre (celui avec la plus grande accuracy). Les valeurs de p-values sont résumées dans la Table 2.

Les matrices de confusions ont également été utilisées afin de mesurer la performance de nos modèles. En particulier, les matrices des modèles BERT et XL-BiSEPs sont visibles à la Fig. 6.

Le principal constat effectué est que la classification neutre est celle qui cause le plus de problèmes. En effet, les deux modèles semblent distinguer assez bien les contradictions des conséquences logiques, mais une zone grise semble apparaître autour de la classe neutre. C'est d'ailleurs ce qui nous a poussé à utiliser un second objectif afin d'augmenter nos données et générer plus de paires de phrases neutres dans l'espoir de mieux les distinguer, même si cela impliquait d'introduire un biais dans nos données par l'ajout de données représentant la classe neutre.

6. Travaux futurs et conclusion

Nos résultats, bien que déjà bons, pourraient certainement être encore améliorés. Pour continuer ce projet, plusieurs pistes sont encore à explorer.

Tout d'abord, dans les dernières avancées majeures en traitement de la langue, on a pu remarquer qu'en général, plus un modèle a de paramètres, plus il est performant. Avec une puissance de calcul suffisante, entraîner nos mêmes algorithmes "on-top" of XLNet large pourrait amener à de meilleures performances.

On pourrait également tenter d'autres modèles de type transformeurs sortis récemment, tels que GPT/GPT-2, XLM, RoBERTa, etc... et inclure leurs prédictions dans une méthode d'ensemble.

Compte tenu le temps d'entraînement des modèles avec autant de données, nous n'avons pas pu tester tous les hyperparamètres. Dans des travaux futures, on pourrait donc tenter de modifier la pondération de chacun des objectifs, la valeur de dropout, les fonctions d'activation (ReLU, tanh...), et d'autres.

Il est évident également de penser que pour la version multi-tâches, on pourrait utiliser diverses données encore plus variées et plus nombreuses, qui aideraient certainement à améliorer notre score.

Ensuite, les architectures "on-top" que nous avons proposées peuvent très certainement être améliorées également. Pour notre modèle XL-BiSEPs, nous ne prenons en compte que les sorties associées aux tokens séparateurs. Il est possibles que de prendre en compte les sorties correspondant à toute la phrase A d'un côté, et toute la phrase B de l'autre, puissent aider le modèle à apprendre une meilleure représentation de chacune des phrases.

Enfin, la méthode de combinaison des différents modèles pourrait être revue afin d'ajouter une notion de poids associée aux sorties softmax de chaque modèle. La classification majoritaire, bien qu'elle ait mieux fonctionné lors de nos tentatives, demeure une méthode assez drastique ne permettant pas de considérer le degré de certitude des prédictions de chaque modèle. Une pondération plus nuancée et l'utilisation d'apprentissage machine dans la combinaison des différents modèles pourrait améliorer nos performances.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[2] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237.

[3] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[4] Soares, L. B., FitzGerald, N., Ling, J., Kwiatkowski, T. (2019). Matching the Blanks: Distributional Similarity for Relation Learning. arXiv preprint arXiv:1906.03158

[5] Chris McCormick. (2019) XLNet Fine-Tuning Tutorial with PyTorch. [En ligne]. https://mccormickml.com/2019/09/19/XLNet-fine-tuning/

[6] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. arXiv preprint 1901.11504.