Advanced Natural Language Processing - CS - 2024 Competition

Nathan Chalumeau Théotime Fossat Abdelillah Bourchad
Louis Malichier Giorgi Oganezovi

Team NC submission

Abstract

Ce rapport présente les défis et les solutions utilisés pour la classification automatique de textes sous 12 catégories pour le concours NLP-CS-2024. Après une phase initiale de génération des embedding de texte avec BERT, nous avons transitionné vers l'utilisation de "all-MiniLM-L6-v2", un modèle Sentence Transformer, pour une représentation plus efficace des données. Nous avons ensuite affiné ce modèle avec une augmentation des données, permettant une amélioration significative de la précision. L'adaptation ultérieure vers le réseau Mistral/Mistral-TB-Instruct-v0.2 a abouti à des meilleurs scores de précision. Les résultats obtenues valident notre méthode combinant entraînement, fine-tuning et enrichissement des données, fournissant un aperçu de leur efficacité et de leurs limites.

1 BERT

Nous avons opté pour une approche initiale utilisant BERT pour notre tâche de classification. Pour ce faire, nous avons employé les outils BertTokenizer et BertModel de la bibliothèque transformers.

En raison de la quantité limitée de données disponibles pour l'entraînement de notre modèle, nous avons choisi d'utiliser BERT en tant que modèle basé sur les caractéristiques ("feature based model"). Concrètement, nous avons utilisé BertTokenizer et BertModel pour obtenir un embedding de chaque phrase, avec une dimension de 768. Par la suite, nous avons appliqué un algorithme de classification KNN sur les embeddings obtenus. Toutefois, cette approche n'a pas produit des résultats satisfaisants, ce qui nous a conduit à finetuner BERT.

Pour ce faire, nous avons ajouté une couche de sortie linéaire suivie d'une fonction softmax. Étant donné la petite taille de notre ensemble d'entraînement, nous avons adopté une stratégie d'apprentissage semi-supervisé. Cela signifie que nous avons inclus dans notre ensemble d'entraînement les données de test pour lesquelles nous avons pu prédire la classe avec une probabilité supérieure à un seuil fixé.

De plus, pour augmenter la taille de notre ensemble d'entraînement, nous avons introduit une technique de masquage, consistant à masquer aléatoirement un mot de chaque phrase.

Après analyse de notre ensemble de test, nous avons observé que de nombreuses phrases contenaient le nom des catégories visées. Par exemple, la phrase "The link between poor nutrition and bone health issues such as osteoporosis is well established." peut être aisément classée dans la catégorie "health". En conséquence, nous avons ajouté une étape préliminaire à l'utilisation de BERT, consistant à rechercher la présence de ces catégories dans les phrases. Si une correspondance est trouvée, la catégorie correspondante est directement prédite ; sinon, nous faisons appel à BERT. Cette approche a nettement amélioré les performances de notre modèle.

2 Sentence Transformer

Malgré ces étapes, les performances de nos modèles demeurent insuffisantes. Nous avons donc décidé de recourir à un modèle différent dans le but d'améliorer nos résultats.

Nous avons opté pour le modèle "all-MiniLM-L6-v2" de la bibliothèque sentence_transformers. Ce modèle génère un embedding de taille 384 pour chaque phrase. Nous avons ensuite évalué ce modèle selon deux approches : en tant que modèle basé sur les caractéristiques ("feature based") et après finetuning. Nous avons suivi les mêmes étapes que celles appliquées avec BERT, notamment l'apprentissage semi-supervisé et la préclassification.

Les performances de ce modèle se sont avérées supérieures à celles obtenues avec BERT. Nous

avons atteint une précision de 61 % en utilisant l'approche basée sur les caractéristiques, et de 63 % après finetuning du modèle.

3 Mistral

Les résultats obtenus avec BERT n'étaient pas ceux attendus. Nous avons donc envisagé qu'un modèle de langage plus grand et mieux entraîné pourrait réaliser cette tâche avec plus d'efficacité. Notre choix s'est porté sur le réseau développé par Mistral AI.

3.1 Modèle

Nous avons utilisé le réseau "mistralai/Mistral-7B-Instruct-v0.2" distribué par HuggingFace.

Compte tenu du poids conséquent du réseau, nous avons opté pour sa quantisation en 4 bits afin de réduire les ressources nécessaires à son exécution.

3.2 Requête

Afin d'obtenir des informations structurées, nous avons fourni au modèle des commandes et des exemples précis, ainsi que les phrases à catégoriser. Le modèle devait renvoyer un seul mot, contenu entre les caractères "[/INST]" et "</s>".

Les phrases à catégoriser étant en anglais, nous avons choisi de rédiger les instructions dans un anglais concis pour minimiser les ressources nécessaires lors du traitement d'un grand volume de requêtes.

Voici la requête utilisée :

```
CONTINUE THE EXAMPLE BY CHOOSING ONE OF THE FOLLOWING WORDS: "Politics", "Health", "Finance", "Travel", "Food", "Education", "Environment", "Fashion", "Science", "Sports", "Technology", "Entertainment"

EXAMPLE:

"The mayor announced a new initiative to improve public transportation." => "Politics"

"The stock market saw a significant drop following the announcement." => "Finance"

"LA PHRASE À CLASSER" =>.
```

3.3 Entraînement

Le modèle ne parvenait pas à renvoyer la sortie désirée exactement. Nous avons donc procédé à un fine-tuning pour lui apprendre la sortie désirée.

Les paramètres retenus sont les suivants :

Paramètre	Valeur
learning_rate	5×10^{-5}
per_device_train_batch_size	1
num_train_epochs	8
weight_decay	0.01

Table 1: Paramètres de fine-tuning du modèle

Avec cet entraînement, nous avons obtenu un score de 76%.

3.4 Augmentation des données d'apprentissage

Considérant que le volume initial de données était insuffisant, nous avons tenté d'améliorer le score en augmentant le volume de données.

Nous n'avons ajouté que deux phrases par catégorie, soit un total de 24 phrases.

Le score obtenu après cette augmentation est de 84%.

4 Conclusion

En conclusion, ce projet illustre l'efficacité de combiner des modèles pré-entraînés comme BERT avec des techniques d'apprentissage machine learning comme le KNN et des stratégies de fine-tuning pour classifier des textes complexes. À travers l'expérimentation avec différentes architectures de modèles et des méthodes d'augmentation de données, nous avons démontré que même avec un ensemble de données relativement restreint, il est possible d'obtenir des résultats significatifs en affinant les approches de préparation des données et en adaptant les modèles.

Afin d'améliorer les performances, nous avons utilisé le modèle "all-MiniLM-L6-v2" qui permet de générer des embeddings de phrases plus compactes, ainsi que le réseau Mistral/Mistral-TB-Instruct-v0.2 de Hugging Face qui a permis une exécution efficace grâce à une quantisation intelligente des poids du modèle.

Cette méthode nous a permis d'avoir une précision de 76% sur l'ensemble du teste. Dans le but d'améliorer ces résultats, nous avons décider d'augmenter les données. Bien qu'elle est modeste

en quantité, elle a eu un impact significatif sur la précision du modèle qui a dépassé 84%.

Ce projet constitue un exemple pour des prochaines recherches dans le domaine de la classification textuelle, suggérant que des performances encore meilleures pourraient être atteintes en explorant davantage les techniques de fine-tuning et d'augmentation des données.