Utilisation des modèles Transformer avec Hugging Face

Traitement Automatique du Langage Naturel

Département d'Informatique

10 juin 2025

Table des matières

1 Introduction

L'écosystème Hugging Face a démocratisé l'accès aux modèles Transformer pré-entraînés, permettant aux praticiens de bénéficier d'architectures sophistiquées sans avoir à les implémenter from scratch. Ce TP vous permettra de maîtriser l'utilisation pratique de ces modèles pour résoudre deux tâches fondamentales du NLP : la classification de texte et la génération/réponse automatique.

Vous apprendrez à naviguer dans le Hub, sélectionner des modèles appropriés, les fine-tuner sur vos données, et évaluer leurs performances de manière rigoureuse.

2 Objectifs pédagogiques

— Maîtriser l'écosystème Hugging Face (Hub, Transformers, Datasets)
— Apprendre à sélectionner des modèles appropriés pour différentes tâches
— Développer des pipelines d'évaluation robustes
— Comparer les performances de différentes approches
— Déployer et utiliser des modèles entraînés

3 Consignes générales

3.1 Modalités pratiques

— **Durée recommandée :** 4h
— **Travail :** Binôme
— **Livrable :** notebook Jupyter avec code et analyses
— **Bonus :** déploiement d'une application interactive avec Gradio
— **Librairies :** transformers, datasets, torch, evaluate

4 Partie 1 : Classification de sentiment

4.1 Questions

1. **Exploration du Hub et sélection de modèles :**
   — Explorez le Hugging Face Hub pour identifier 3 modèles différents adaptés à la classification de texte
   — Analysez leurs caractéristiques : taille, langue, domaine d'entraînement
   — Justifiez le choix de 2 modèles pour votre expérience
   — Comparez les architectures sous-jacentes (encoder-only vs encoder-decoder)
2. **Implémentation baseline avec pipeline :**
   — Utilisez l'API `pipeline` pour créer un classificateur de sentiment
   — Testez sur des exemples simples en français et anglais
   — Évaluez les performances sur un petit dataset de test
   — Identifiez les limites de cette approche zero-shot

5 Partie 2 : Génération de texte et question-réponse

5.1 Questions

(a) **Sélection et comparaison de modèles génératifs :**
- Identifiez 2 modèles adaptés à la génération de texte sur le Hub
- Comparez leurs approches : autoregressive vs seq2seq
- Testez leurs capacités de génération avec différents prompts
- Analysez la qualité et la cohérence des textes générés

(b) **Implémentation d'un système de question-réponse :**
- Utilisez un modèle pré-entraîné pour le question-answering extractif
- Créez une base de connaissances à partir de documents textuels
- Implémentez une interface de recherche et extraction de réponses
- Évaluez la pertinence des réponses sur des questions test

(c) **Techniques de décodage avancées :**
- Implémentez différentes stratégies : greedy, beam search, sampling
- Expérimentez avec les paramètres temperature et top-k/top-p
- Analysez l'impact sur la diversité et la qualité des générations

6 Partie 3 : Analyse comparative et optimisation

6.1 Questions

(a) **Évaluation multi-modèles :**
- Comparez les performances des différents modèles testés
- Analysez le trade-off performance/vitesse/taille
- Créez des tableaux de résultats et visualisations
- Identifiez les forces et faiblesses de chaque approche

(b) **Analyse d'erreurs qualitative :**
- Sélectionnez des exemples mal classifiés/générés
- Analysez les patterns d'erreurs communes
- Proposez des hypothèses sur les causes des échecs
- Suggérez des améliorations possibles

(c) **Optimisation et déploiement :**
- Implémentez des techniques d'optimisation (quantization, distillation)
- Mesurez l'impact sur les performances et la latence
- Testez la robustesse avec des inputs adversariaux

6.2 Guide méthodologique

6.2.1 Conseils d'implémentation

- Utilisez `AutoTokenizer` et `AutoModel` pour la flexibilité

— Implémentez un logging détaillé pour tracer vos expériences
— Utilisez `wandb` ou `tensorboard` pour le monitoring
— Sauvegardez régulièrement vos checkpoints et configurations
— Documentez vos choix d'hyperparamètres et leurs justifications

6.2.2 Bonnes pratiques

— Validez vos implémentations sur des datasets de référence
— Utilisez des seeds fixes pour la reproductibilité
— Implémentez des tests unitaires pour vos fonctions critiques
— Créez des environnements virtuels isolés pour chaque expérience
— Respectez les licences des modèles utilisés

7 Datasets recommandés

7.1 Classification

— Sentiment140 (Twitter sentiment)
— Amazon Product Reviews
— 20 Newsgroups
— AG News Classification
— Custom dataset de votre choix

7.2 Génération/QA

— SQuAD (Question Answering)
— MS MARCO
— WikiText pour language modeling
— Common Crawl News
— Corpus spécialisé de votre domaine

8 Évaluation

Votre TP sera évalué sur :

— **Maîtrise des outils** Hugging Face et implémentation correcte (30%)
— **Qualité de l'analyse** comparative et interprétation des résultats (25%)
— **Rigueur expérimentale** : méthodologie, validation, reproductibilité (20%)
— **Documentation et présentation** : clarté du code et des explications (15%)
— **Créativité et extensions** : approches originales et approfondissements (10%)

9 Ressources recommandées

— Documentation officielle Hugging Face Transformers
— Cours "Hugging Face Course" (huggingface.co/course)
— Papers With Code pour les benchmarks
— Documentation des datasets : huggingface.co/datasets
— Guides de fine-tuning et optimisation
— Forums et communauté Hugging Face

10 Modalités de rendu

10.1 Informations pratiques

— **Format :** Notebook Jupyter avec code exécutable et analyses
— **Envoi :** dépôt sur la plateforme du cours
— **Fichiers attendus :**
  — `main_notebook.ipynb` : notebook principal avec toutes les expériences
  — `requirements.txt` : dépendances et versions
  — `config/` : fichiers de configuration des modèles
  — `results/` : résultats, métriques et visualisations
  — `README.md` : synthèse des résultats et instructions

10.2 Livrables attendus

— Rapport d'analyse comparative des modèles testés
— Code commenté et modulaire
— Métriques de performance détaillées
— Visualisations des résultats
— Réflexion critique sur les limites et améliorations possibles

Bon travail !