Rapport Technique – Assistant Intelligent NLP

Projet : Résumé et Extraction d'Informations
Durée : 2 semaines
Dataset : BBC News
Technologies : SpaCy, Word2Vec, LSTM, BERT, DistilBERT, Streamlit

1. Introduction et Corpus

1.1 Contexte

Le traitement automatique du langage naturel (NLP) permet aujourd'hui d'analyser, classer et résumer de grands volumes de textes. Ce projet vise à concevoir un **assistant intelligent** capable de :

- Classer des articles par thématique
- Extraire des entités nommées (personnes, organisations, lieux)
- Générer des résumés extractifs et abstractifs

L'ensemble de la chaîne NLP est implémenté : nettoyage, embeddings, modèles séquentiels, Transformers et évaluation comparative.

1.2 Corpus utilisé

Nous avons utilisé le dataset **BBC News**, composé d'articles répartis en plusieurs thématiques.
Pour ce projet, trois catégories ont été sélectionnées :

- Business
- Entertainment
- Tech

Environ **150 documents par catégorie** ont été extraits, soit un total d'environ **450 articles**.

Chaque document est stocké sous forme de fichier `.txt` et contient le texte brut de l'article.

2. Pipeline de Nettoyage

2.1 Nettoyage Regex

Les textes bruts contiennent souvent du bruit : URLs, balises HTML, caractères spéciaux.
Nous avons appliqué les opérations suivantes :

- Suppression des URLs
- Suppression des balises HTML
- Conservation uniquement des lettres
- Normalisation des espaces

```
texte = re.sub(r'https?://\S+|www\.\S+', '', texte)
texte = re.sub(r'<.*?>', '', texte)
texte = re.sub(r'[^a-zA-Z\s]', ' ', texte)
```

2.2 Traitement SpaCy

SpaCy est utilisé pour :

- Tokenisation

- Lemmatisation
- Suppression des stopwords
- POS-tagging
- Analyse des dépendances

Seuls les mots informatifs sont conservés pour la suite du pipeline.

2.3 Résultat

Chaque document est transformé en une version nettoyée prête pour :

- L'entraînement Word2Vec
- La classification
- Les modèles Transformers

3. Embeddings et Visualisations

3.1 Word2Vec

Un modèle **Word2Vec** est entraîné sur le corpus :

- Dimension : 100
- Fenêtre : 5
- min_count : 2

Chaque mot est représenté par un vecteur dense de 100 dimensions.

3.2 Vecteurs de documents

Les vecteurs de documents sont obtenus par **moyenne des vecteurs de mots** :

```
doc_vector = mean(word_vectors)
```

3.3 Visualisation PCA

Une réduction de dimension avec **PCA (2D)** permet de visualiser les documents.

Résultat :

- Les documents de même thématique se regroupent
- Les catégories sont partiellement séparées

Cela montre que les embeddings capturent une structure sémantique pertinente.

4. Modèles Séquentiels – LSTM

4.1 Préparation des données

- Tokenisation Keras
- Padding à 200 tokens
- Embeddings Word2Vec intégrés

Les labels sont encodés en **One-Hot**.

4.2 Architecture du modèle

- Embedding (Word2Vec, non entraînable)

- LSTM bidirectionnel (64 neurones)
- Dropout (0.5)
- Dense (32)
- Softmax (3 classes)

4.3 Entraînement

- Optimiseur : Adam
- Loss : Categorical Crossentropy
- Epochs : 10

4.4 Résultats

Modèle	Accuracy	F1-score
LSTM	0.85	0.84

Le modèle est rapide (~10 ms d'inférence) mais moins précis que les Transformers.

5. Transformers – BERT & DistilBERT

5.1 Fine-tuning de BERT

Le modèle **bert-base-uncased** est utilisé pour la classification.

Paramètres :

- max_length = 128

- learning_rate = 2e-5
- epochs = 3

5.2 DistilBERT

DistilBERT est une version allégée de BERT :

- Plus rapide
- Moins lourd
- Légère perte de précision

5.3 Résultats

Modèle	Accuracy	F1-score	Temps inférence
LSTM	0.85	0.84	~10 ms
BERT	0.96	0.96	~150 ms
DistilBERT	0.94	0.94	~70 ms

Les Transformers surpassent largement le LSTM en précision.

6. Extraction et Résumé

6.1 Extraction d'entités

SpaCy est utilisé pour détecter :

- PERSON
- ORG
- GPE

Exemple :

```
("Google", ORG), ("London", GPE)
```

6.2 Résumé extractif

Une méthode simple basée sur les phrases :

- Les 2 premières phrases sont sélectionnées
- Permet un résumé rapide mais basique

6.3 Résumé abstractif

Le modèle **BART (facebook/bart-large-cnn)** génère un résumé reformulé.

Avantages :

- Plus naturel
- Plus synthétique
- Plus lisible

7. Évaluation Comparative

7.1 Classification

Modèle	Accuracy	F1	Temps

LSTM	0.85	0.84	10 ms
DistilBERT	0.94	0.94	70 ms
BERT	0.96	0.96	150 ms

7.2 Résumé

Méthode	Qualité	Rapidité
Extractif	Moyenne	Très rapide
Abstractif (BART)	Excellente	Plus lent

7.3 Extraction

SpaCy fournit des résultats fiables avec peu de configuration.

8. Conclusion et Limites (1 page)

8.1 Conclusion

Ce projet a permis de construire un **assistant NLP complet** intégrant :

- Nettoyage
- Embeddings
- LSTM
- Transformers
- Extraction
- Résumé
- Interface Streamlit

Les modèles Transformers offrent les meilleures performances.

8.2 Limites

- Dataset limité (BBC uniquement)
- Pas de fine-tuning du résumé
- Résumé extractif simplifié
- DistilBERT pas toujours stable

8.3 Perspectives

- Ajouter TF-IDF pour mots-clés
- Fine-tuning de BART
- Support PDF
- Multilingue
- API REST