M2SOL034 Corpus, ressources et linguistique outillée

TD10 : Évaluation complète d'un pipeline NLP : de l'annotation à l'analyse des performances

Wiem TAKROUNI

Sorbonne Université

Master « Langue et Informatique »

UFR Sociologie et Informatique pour les Sciences Humaines

Semestre 2, 2024-2025, le 11 avril 2025

Objectifs du TD

1. Appliquer les métriques d'évaluation (classification, annotation, génération).
2. Évaluer manuellement un corpus annoté et calculer l'accord inter-annotateurs.
3. Comparer plusieurs modèles NLP (SVM, BERT…).
4. Utiliser des outils Python pour l'évaluation (sklearn, seqeval, evaluate).
5. Interpréter les résultats et identifier les biais.

Partie 1 – Évaluation d'un corpus annoté

Objectif : évaluer la qualité d'un corpus en utilisant des métriques d'accord inter-annotateurs.

Données : mini corpus annoté à deux ou trois annotateurs (NER, sentiment, etc.)

Étapes :

1. Charger les annotations dans un format compatible (ex : CSV ou JSON).
2. Calculer **Cohen's Kappa** (pour 2 annotateurs) et/ou **Krippendorff's Alpha**.
3. Identifier les divergences (exemples contradictoires).
4. Proposer des ajustements ou règles d'annotation.

Librairies :

```
from sklearn.metrics import cohen_kappa_score
```

Partie 2 – Évaluation de modèles de classification
Objectif : entraîner et évaluer plusieurs modèles de classification de sentiments.
Données : IMDb (ou Twitter, Amazon, etc.)
Étapes :

1. Prétraitement : nettoyage, lemmatisation, stopwords.
2. Modèle 1 : SVM
3. Modèle 2 : BERT fine-tuné avec transformers
4. Calculer : **accuracy**, **precision**, **recall**, **F1-score**
5. Générer la matrice de confusion, analyser les erreurs.

Bonus : visualiser avec seaborn.

Partie 3 – Évaluation d'un modèle de NER
Objectif : comparer deux modèles pour la reconnaissance d'entités.
Données : CoNLL-2003 ou corpus annoté simplifié (BIO format).
Étapes :

1. Appliquer un modèle spaCy ou Flair pré-entraîné.
2. Annoter quelques phrases à la main.
3. Évaluer avec seqeval (F1, precision, recall par entité).
4. Comparer les sorties, détecter les entités mal reconnues.

Partie 4 – Évaluation de la génération automatique
Objectif : tester un modèle de résumé automatique et évaluer les résultats.
Données : quelques extraits d'articles + résumés de référence.
Modèles possibles : BART, T5, Pegasus via transformers.
Étapes :

1. Générer des résumés.
2. Évaluer avec **ROUGE** et **METEOR** via evaluate.

```
from evaluate import load
rouge = load("rouge")
meteor = load("meteor")
```

3. Discussion : différences de formulation, paraphrases, omissions.

Partie 5 – Analyse critique et biais
Objectif : développer un esprit critique sur les résultats obtenus.
Exercice :

- Identifier les biais dans les corpus utilisés.
- Critiquer les métriques choisies (ex : BLEU pour résumé créatif).
- Proposer des alternatives.