Corpus, ressources et linguistique outillée

Corpus et IA

Dr Wiem Takrouni
Sorbonne Université
Master Langue et Informatique
UFR Sociologie et Informatique pour les Sciences Humaines

March 20, 2025

Introduction

Qu'est-ce qu'un corpus ?

- **Un corpus** est un ensemble de textes ou de données linguistiques collectées pour l'analyse, l'enseignement ou le **traitement automatique des langues (TAL)**.
- Il peut être constitué de textes écrits (presse, littérature) ou de dialogues oraux (enregistrements, podcasts).

Introduction

Importance des corpus dans le Traitement Automatique des Langues (TAL)

- Les corpus sont essentiels pour **entraîner, tester et évaluer** des modèles de TAL, comme la **traduction automatique**, l'**analyse de sentiment** et la **reconnaissance d'entités nommées**.

Rôle clé des corpus dans l'apprentissage automatique

- Ils servent de base pour l'**entraînement des modèles d'IA**, en fournissant les données nécessaires pour identifier les **patterns**, les **structures linguistiques** et le **vocabulaire**.

Types de Corpus

Corpus Textuels :

- **Exemples :** articles de presse, livres, réseaux sociaux (tweets, blogs).
- **Utilisation :** analyse de la syntaxe, du sentiment, des relations sémantiques.

Types de Corpus

Corpus oraux (enregistrements, dialogues, sous-titres)

- **Exemples :** enregistrements de conversations, podcasts, dialogues dans des films ou sous-titres.
- **Utilisation :** reconnaissance vocale, analyse de la prosodie, transcription automatique.

Types de Corpus

Corpus multilingues (alignés vs non alignés)

- **Alignés :** Traduction parallèle (ex. : Europarl).
- **Non alignés :** Textes dans différentes langues sans correspondance directe.

Types de Corpus

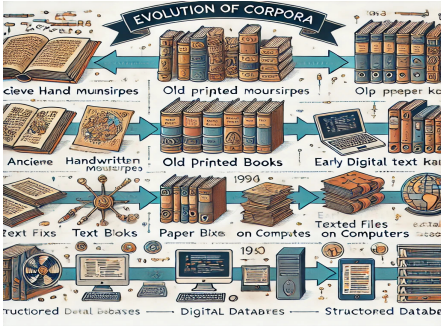
Corpus annotés vs non annotés (exemples : Universal Dependencies, OntoNotes)

- **Annotés :** Contiennent des annotations supplémentaires (étiquettes grammaticales, entités nommées).
- **Exemples :** Universal Dependencies, OntoNotes.

Évolution des Corpus et des Techniques d'Analyse

Passage des corpus manuels aux corpus numériques :

- Les premiers corpus étaient créés manuellement, avec des transcriptions et des annotations faites à la main.
- Aujourd'hui, des outils automatisés permettent de créer et annoter des corpus à grande échelle.



Évolution des Corpus et des Techniques d'Analyse

Amélioration des techniques d'analyse :

- Analyse de texte classique : Approches statistiques, bag-of-words, modèles n-grammes.
- Nouveaux modèles : Modèles de langage basés sur des réseaux de neurones profonds (ex. : transformers, LLMs).

Évolution des Corpus et des Techniques d'Analyse

Sources de données : Common Crawl, Wikipedia, BooksCorpus

- L'évolution des corpus a favorisé la montée en puissance des modèles d'IA, avec des ensembles de données de plus en plus vastes, diversifiés et complexes (ex. : Common Crawl, Wikipedia, BooksCorpus).

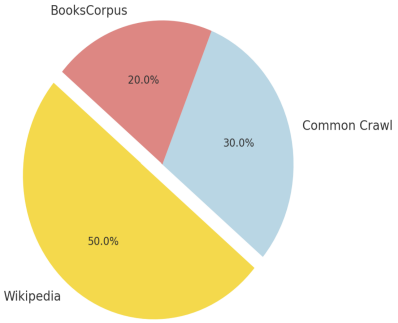
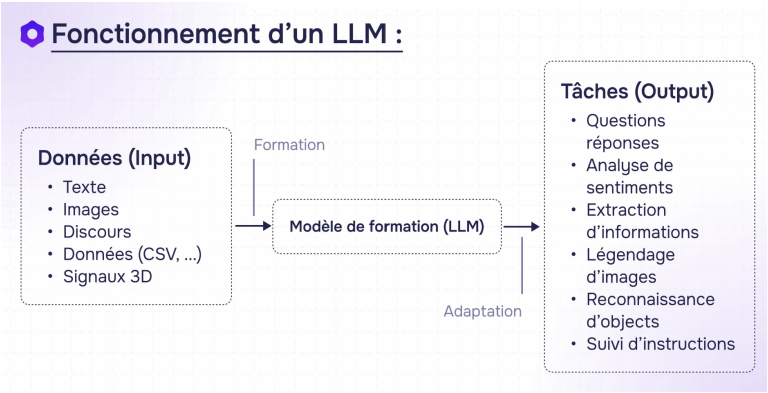


Figure: Répartition des sources de données dans les corpus de TAL

Pipeline de Création d'un Modèle de Langage (LLM)



Applications des Corpus en TAL

Traduction automatique
(Exemple : Google Translate vs DeepL)

Principe

Utilisation de corpus parallèles (paires de textes multilingues) pour entraîner des modèles de traduction automatique.

Google Translate

- Modèle basé sur des réseaux neuronaux.
- Utilisation de vastes corpus

DeepL

- Corpus plus qualitatif.
- Traductions plus naturelles.

Analyse de sentiment (exemple : classification des avis sur Amazon)

- Principe : Analyser le ton ou l'opinion exprimée dans un texte, souvent dans les critiques ou les avis. Utilisation de corpus étiquetés avec des sentiments (positifs, négatifs, neutres).
- Exemple :
  - Avis sur Amazon : Le système évalue les avis des utilisateurs pour déterminer si le commentaire est positif ou négatif.
- Applications :
  - Surveillance des sentiments sur les réseaux sociaux (politique, produits, etc.)
  - Analyse du service client et feedback.
- Challenges : Ambiguïté des sentiments (sarcasme, contextes multiples).

Résumé automatique (exemple : BART, Pegasus)

- Principe : Extraction des points clés d'un texte pour créer un résumé automatique. Utilisation de corpus pour entraîner des modèles à générer des résumés.
- Exemples :
  - BART : Modèle génératif basé sur un corpus de résumés et de textes.
  - Pegasus : Entraîné sur des corpus d'articles de presse, capable de résumer efficacement des documents longs.
- Challenges : Résumer de manière cohérente, maintenir la précision tout en réduisant la taille du texte.

Génération de texte (exemple : GPT pour la rédaction automatique)

- Principe : Créer du texte automatiquement en fonction d'un input donné. Les modèles sont formés à partir de grands corpus de textes (livres, articles, conversations, etc.).
- Exemple :
* GPT : Utilise un large corpus de textes pour générer des textes, rédiger des articles, ou même simuler des conversations.
- Applications : Rédaction automatique, chatbots, génération de contenu marketing.
- Challenges : Créer des textes qui semblent naturels et cohérents sur le long terme.

Applications des Corpus en TAL

Recherche d'information (exemple : Whoosh, Elasticsearch)

- Principe : Trouver des documents ou informations pertinentes dans un grand corpus de données. Utilisation de techniques d'indexation et de recherche pour extraire des résultats pertinents.
- Exemples :
  - Whoosh : Utilisé pour l'indexation de documents en Python, adapté pour des corpus textuels simples.
  - Elasticsearch : Outil de recherche distribué, utilisé pour indexer de vastes quantités de données structurées et non structurées.
- Applications : Recherche sur le web, bibliothèques numériques, systèmes de recommandation.
- Challenges : Gestion des synonymes, recherche contextuelle.

Détection d'entités nommées (exemple : Spacy, Flair)

- Principe : Identifier et classer les entités dans un texte (personnes, lieux, organisations, etc.), pour en extraire des informations structurées.
- Exemples :
  - Spacy : Utilise un modèle d'entités nommées pour identifier et classifier les entités dans le texte.
  - Flair : Offre une détection d'entités et un tagging linguistique précis basé sur des modèles pré-entraînés sur des corpus variés.
- Applications : Extraction d'informations dans des articles de presse, analyse de documents juridiques, surveillance de marques.
- Challenges : Gérer les entités ambiguës et les contextes spécifiques.

Biais et Limitations des Corpus

Biais dans les corpus

- Biais dans les corpus : Données déséquilibrées, erreurs d'annotation, manque de diversité linguistique.

Biais et Limitations des Corpus

Exemple : Étude sur les biais de genre et culturels dans GPT

- Exemples d'erreurs dans GPT et d'autres LLMs (stéréotypes de genre, représentations culturelles biaisées).
- Études sur les réponses différenciées selon les profils de requêtes.

Biais et Limitations des Corpus

Problèmes de qualité et nettoyage des corpus

- Bruit dans les données : fautes, doublons, informations erronées.
- Nécessité de l'étiquetage humain et des corrections.
- Fiabilité des sources : Wikipedia, Common Crawl vs. Corpus spécialisés.

Biais et Limitations des Corpus

Stratégies pour réduire ces biais

- Augmenter la diversité des sources et des annotations.
- Techniques de rééquilibrage des corpus (augmentation de données, techniques de débiaisage).
- Évaluation continue et correction dynamique des modèles IA.

Collecte et Préparation des Corpus

Où et comment récupérer des corpus ?

- Bases de données publiques : Common Crawl, OpenSubtitles, Wikipedia Dumps.
- Corpus spécialisés : Europarl, OPUS, COCA (Corpus of Contemporary American English).
- Web scraping et extraction manuelle de données.

Collecte et Préparation des Corpus

Formats courants :

- TXT : Format brut, lisible mais non structuré.
- XML : Structuré, adapté aux annotations complexes.
- JSON : Utilisé pour les données issues d'API.
- TEI : Format standard en linguistique.

Collecte et Préparation des Corpus

Nettoyage et prétraitement (tokenisation, lemmatisation, stopwords)

- Tokenisation : Découpage des phrases en mots/tokens.
- Lemmatisation et racinisation : Réduction des mots à leur forme canonique.
- Suppression des stopwords : Élimination des mots vides (ex : "le", "de", "et").
- Détection et suppression du bruit (émoticônes, caractères spéciaux, etc.)

Outils de gestion de corpus (NLTK, SpaCy, Gensim)

- NLTK : Outils de prétraitement linguistique.
- SpaCy : Bibliothèque NLP performante et optimisée.
- Gensim : Utilisé pour l'analyse sémantique et les modèles de topic modeling.

Collecte et Préparation des Corpus

Exemple : Extraction et nettoyage d'un corpus en Python

- Récupération d'un texte depuis une source (ex: un article Wikipédia via l'API MediaWiki).
- Application de prétraitements avec NLTK ou SpaCy.
- Stockage du corpus nettoyé dans un format exploitable (JSON, CSV, TXT).

Collecte et Préparation des Données

Objectif	Tâches	Outils/Techniques
Corpus propre et structuré	Récupération : Common Crawl, Wikipedia, OpenSubtitles... Nettoyage : suppression du bruit, tokenisation, lemmatisation Annotations (NER, POS tagging)	NLTK, SpaCy (tokenisation, lemmatisation) Pandas, regex (nettoyage) Doccano, Prodigy (annotation)

Table: Collecte et préparation des données

Étude de Cas et Exemples Pratiques

Choisir un Modèle d'IA : Pré-entraîné ou à Entraîner

- Objectif : Sélectionner une approche adaptée selon les ressources et objectifs.

* Option 1 : Modèle pré-entraîné
  Exemples : BERT, GPT, T5 (pour classification, résumé, génération...)
  Outils : Hugging Face Transformers
* Option 2 : Entraînement d'un modèle
  Exemples : Réseaux de neurones récurrents (LSTM), Transformers entraînés sur mesure
  Outils : TensorFlow, PyTorch

Étude de Cas et Exemples Pratiques

Adapter un Modèle (Fine-Tuning ou Feature Extraction)

- Objectif : Ajuster un modèle pré-entraîné sur un corpus spécifique.
- Méthodes :
  - Fine-tuning sur un corpus spécifique
  - Feature extraction : utiliser des embeddings et entraîner un classificateur
- Modèles & Outils :
  - BERT Fine-Tuning avec Hugging Face
  - Word2Vec, FastText pour extraction de caractéristiques

Évaluation du Modèle

- Objectif : Mesurer les performances avant utilisation.
- Tâches :
  - Séparer un corpus en train/test
  - Calculer précision, rappel, F1-score
  - Détecter et analyser les erreurs
- Outils :
  - Scikit-learn pour les métriques
  - TensorBoard pour le suivi des performances

Conclusion

Corpus

Un corpus propre et bien préparé est essentiel pour garantir la qualité des résultats.

Modèle

Un modèle adapté, qu'il soit pré-entraîné ou entraîné sur mesure, est crucial pour une performance optimale.

Évaluation

Une évaluation rigoureuse est indispensable pour garantir la fiabilité et l'efficacité du modèle.