# Programme et modalités d'évaluation

Gustave Cortal

# Présentation

Gustave Cortal, doctorant en traitement automatique des langues pour l'analyse des émotions à l'ENS Paris-Saclay

Mail : gustave.cortal@ens-paris-saclay.fr

Web : `https://lmf.cnrs.fr/Perso/GustaveCortal`

# Traitement Automatique des Langues

Le Traitement Automatique des Langues (TAL), en anglais *Natural Language Processing* (NLP), est un domaine multidisciplinaire impliquant la linguistique, l'informatique et l'intelligence artificielle, qui vise à créer des outils de traitement du langage naturel pour diverses applications[1]

Quelques mots clés : algèbre linéaire, statistiques, régression logistique, classification, régression, modèle de langue, modèle n-gram, réseaux de neurones *feedfordward* et récurrents, plongements ou *embeddings*, transformer, validation croisée, f1-score, analyse du sentiment

[1] `https://www.wikiwand.com/fr/Traitement_automatique_des_langues`

# Horaires

| ECUE | Type | Date début | HEURES | Groupes | Salles |
|---|---|---|---|---|---|
| NLP1 | Magistral | 17/02/2025 14:00 | 1 | SCIA | KB003 (amphi 3) |
| NLP1 | TD | 17/02/2025 15:00 | 2 | SCIA | KB003 (amphi 3) |
| NLP1 | Magistral | 24/02/2025 14:00 | 1 | SCIA | KB003 (amphi 3) |
| NLP1 | TD | 24/02/2025 15:00 | 2 | SCIA | KB003 (amphi 3) |
| NLP1 | Magistral | 03/03/2025 14:00 | 1 | SCIA | KB003 (amphi 3) |
| NLP1 | TD | 03/03/2025 15:00 | 2 | SCIA | KB003 (amphi 3) |
| NLP1 | Magistral | 10/03/2025 14:00 | 1 | SCIA | KB000 (amphi 0) |
| NLP1 | TD | 10/03/2025 15:00 | 2 | SCIA | KB000 (amphi 0) |
| NLP1 | Magistral | 17/03/2025 14:00 | 1 | SCIA | KB003 (amphi 3) |
| NLP1 | TD | 17/03/2025 15:00 | 2 | SCIA | KB003 (amphi 3) |
| NLP2 | Magistral | 24/03/2025 14:00 | 1 | SCIA | KB003 (amphi 3) |
| NLP2 | TD | 24/03/2025 15:00 | 2 | SCIA | KB003 (amphi 3) |
| NLP2 | Magistral | 31/03/2025 14:00 | 1 | SCIA | KB003 (amphi 3) |
| NLP2 | TD | 31/03/2025 15:00 | 2 | SCIA | KB003 (amphi 3) |
| NLP2 | Magistral | 07/04/2025 14:00 | 1 | SCIA | KB003 (amphi 3) |
| NLP2 | TD | 07/04/2025 15:00 | 2 | SCIA | KB003 (amphi 3) |
| NLP2 | Magistral | 28/04/2025 14:00 | 1 | SCIA | KB003 (amphi 3) |
| NLP2 | TD | 28/04/2025 15:00 | 2 | SCIA | KB003 (amphi 3) |
| NLP2 | Magistral | 19/05/2025 14:00 | 1 | SCIA | KB003 (amphi 3) |
| NLP2 | TD | 19/05/2025 15:00 | 2 | SCIA | KB003 (amphi 3) |

# Structure du cours

- Cours (1h)
  - retours sur le TP précédent (?m)
  - rappel cours précédent (5m)
  - cours (40m)
  - **posez des questions pendant le cours** (10m)
- Pause (20m)
- TP (1h40)

# Sommaire

- 17/02 - Tokenization and datasets
- 24/02 - N-gram language models
- 03/03 - Naive Bayes, text classification, evaluation
- 10/03 - Logistic regression, gradient descent
- 17/03 - Vector semantics and embeddings
- 24/03 - Neural networks and neural language models
- 31/03 - Recurrent neural networks
- 07/04 - Transformers, large language models
- 28/04 - *Written exam*
- 19/05 - *Final project presentation*

# Brief summary

Tokenization is splitting text into individual tokens

A language model is a probabilistic model that can compute the probability of a sequence of words and compute the probability of an upcoming word

N-grams are simple probabilistic language models based on Markov assumption

Naive bayes classifiers are generative models based on class-specific unigram

Embedding represents word meaning as a vector

Logistic regressions are discriminative models based on the sigmoid function

Feedforward neural networks handle longer inputs and generalize better compared to N-grams thanks to embeddings, have fixed context windows

Recurrent neural networks handle temporal data inherently in the architecture, have infinite context windows, hidden states have local information

Information flow is better in gated recurrent networks due to better context management

Attention mechanisms solve the bottleneck problem to produce dynamically derived context vectors

Transformers use self-attention layers combined with feedforward layers to handle more complex distant relationships between tokens, enable parallelization due to independent computation between tokens, have fixed context windows

# Modalités d'évaluation

Un *examen écrit* qui teste vos connaissances

Un *projet* en équipe (4 à 5 personnes) portant sur un jeu de données que vous aurez choisi et qui aura été validé par moi-même. Voici les différentes étapes à suivre :

- Présentation du jeu de données (cf. datasheet cours 1)
- Pré-traitement du jeu de données :
  - appliquer la tokénisation à base d'expressions régulières et la tokénisation byte-pair encoding, cf. cours 1 et 2.
  - appliquer des méthodes de normalisation du texte comme la suppression des stop words, la lemmatisation et le fait de tout mettre en minuscule, cf. cours 1 et 2.
- Statistiques descriptives sur vos données : nombre de documents, phrases, tokens, classes à prédire, les tokens les plus fréquents, etc.

- Entrainement de plusieurs modèles prédictifs sur votre jeu de données avec vos propres implémentations ou en utilisant des bibliothèques comme NLTK et scikit-learn :
  - Entrainement de : n-gram (cours 2), bayésien naïf (cours 3), régression logistique (cours 4), tf-idf et word2vec (cours 5), réseaux de neurones feedforwards (cours 6), réseaux de neurones récurrents (cours 7), transformer (cours 8).
  - Évaluer les performances de vos modèles entrainés et comparer les avec des métriques comme la perplexité, le recall, la precision, le f1-score, etc. (cours 3).
  - Varier plusieurs configurations d'entrainement pour évaluer l'impact de certains choix sur les performances. Par exemple, varier la façon de pré-traiter les données, varier les hyperparamètres de vos modèles, etc.
- Limitations de vos approches, difficultés rencontrées et pistes d'améliorations

Points sur les manières créatives d'aborder les étapes : interprétation des modèles, optimisation des hyperparamètres, évaluation rigoureuse des perfromances, transfert de connaissance en croisant plusieurs jeux de données, etc.

Les groupes devront rendre un rapport écrit sur LateX qui rend compte des différentes étapes. Le rapport écrit ne doit **pas dépasser 10 pages** et doit suivre la structure décrite ci-dessus. Deadline pour la rendu du projet écrit le **12/05**

Les groupes devront aussi faire une présentation orale de 10 min durant le dernier cours, le **19/05**

Les projets des étudiants précédents ont été variés : génération de paroles de chansons et de poèmes, modération automatique des conversations sur Twitch, détection de spoilers dans les commentaires de films, identification de textes générés, etc.

# Ressources

**Croisez les ressources !**

*Apprendre les langues aux machines*, Collège de France

*Speech and Language Processing*, Dan Jurafsky and James H. Martin

Chaine youtube « 3Blue1Brown », *Essence of linear algebra*, *Neural Networks*, *Language Models* playlists

Kaggle, HuggingFace, Google Colab, etc.

N'hésitez pas à m'envoyer des ressources intéressantes ...

Comment communiquer ?