

---

LINFO1104  
Rapport de projet - Twitt'Oz

---

Réalisé par  
Corentin Lengele (1475-17-00)

# 1 Introduction

Le but de ce projet était d'écrire un programme en oz capable de parser des fichiers texte contenant un tweet sur chaque ligne en utilisant des threads et d'ensuite etre capable de prédire un mot en fonction d'un texte donné.

## 2 Choix d'implémentation

### 2.1 Parsing

Pour le parsing, j'ai choisi de supprimer toute la ponctuation et de réduire toutes les lettres en minuscules afin de ne pas avoir de problème lors de la sauvegarde. La sauvegarde s'effectue dans un dictionnaire. La clé représente un mot et sa valeur est une liste de mots qui sont précédé par le mot présent dans la clé.

Pour ce qui est de la gestion des threads, j'ai utilisé la méthode "Concurrency for dummies" vue au cours. Cela permet d'éviter les deadlocks à 100 pourcents et de toujours obtenir le même résultat.

### 2.2 Prédiction de mots

Pour la prédiction de mot, j'ai implémenté la méthode en 1-gramme. Le programme va donc donner le mot le plus fréquent en se basant sur le dernier mot écrit.

### 2.3 Modification du Reader

Une classe Reader nous était donnée avec le template du projet. Mais cette classe contenait uniquement une fonction permettant de lire la n-ième ligne d'un fichier. Or j'avais besoin de lire un fichier dans son intégralité. J'ai donc changé la fonction donnée pour qu'elle renvoie la ligne suivante du fichier et non la n-ième.

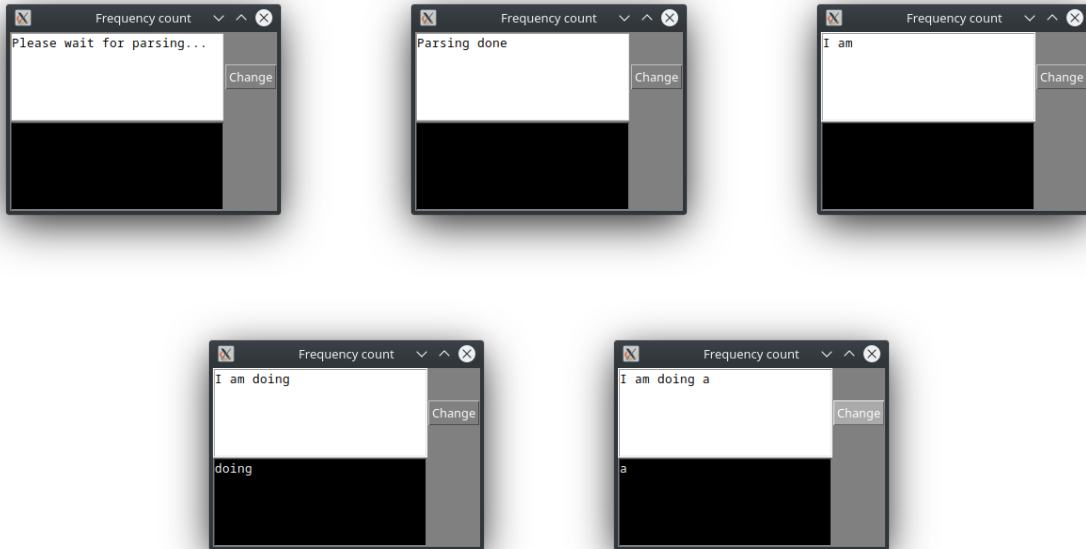
## 3 Difficultés rencontrées

J'ai eu quelques difficultés en retirer toute la ponctuation car, même en filtrant uniquement les alphas-numériques et les espaces, il y avait quand même de la ponctuation dans mes strings.

J'ai également eu beaucoup de difficultés à trouver de la documentation sur ce langage. Il existe une page web qui regroupe certaines choses mais qui contient des doublons ou alors certaines pages ne sont pas accessibles directement. Par exemple, pour la page de document pour les dictionnaires, il en existe deux versions et je n'ai trouvé l'accès à la bonne version que via Google.

## 4 Utilisation de l'interface

En lançant le programme, l'interface va s'afficher en écrivant que le parsing est en cours. Une fois le parsing terminé il sera possible de demander une prédiction de mot en écrivant le texte dans la case du haut et d'ensuite appuyer sur le bouton. En appuyant sur le bouton, le mot prédit va s'ajouter à la phrase écrite dans la première case et s'afficher également dans la deuxième pour ne pas avoir de mal à voir quel mot a été rajouté.



## 5 Conclusion

Le programme fonctionne correctement mais il peut évidemment encore être amélioré. Par exemple en implémentant un algorithme de prédiction plus poussé ou en proposant plusieurs choix de prédiction.