TP 2:
Modèles de langage

Dans ce second TP, nous allons manipuler des transducteurs pondérés. La manipulation consiste à créer un transducteur qui va restaurer des lettres majuscules dans un texte ne comprenant que des lettres minuscules. En effet, dans les applications de traitement automatique de la langue écrite, on préfère généralement mettre les chaînes de caractères en lettres minuscules, ce qui permet d'éviter des doublons.

Exercice 1.

En premier lieu, il s'agit de créer un modèle de langage. Pour cela nous allons utiliser un document écrit, ici un texte issu d'un livre en anglais (*The War of Worlds*) afin de construire un modèle de langage 5-grams. La modélisation d'un phénomène du réel a pour objectif soit de comprendre ce phénomène, soit de prendre des décisions liées à ce phénomène. Ici nous allons modéliser la langue anglaise grâce à des séquences de (au plus) 5 mots consécutifs.

A partir du texte, on peut déterminer la fréquence de ces séquences de 5 mots consécutifs, c'est-à-dire le nombre d'occurrences des 5-grams. Un modèle de langage peut être construit avec des transitions tous les 1-gram à 5-grams vus dans le texte brut, son poids est associé directement à sa fréquence d'apparition.

Sur UMTICE, vous avez à votre disposition plusieurs fichiers:

- `wotw.txt`: le texte directement issu du livre
- `wotw.syms.txt`: l'alphabet correspondant aux différents mots présents dans le livre
- `wotw.lm`: le modèle de langage sous format texte ($\mathcal{L}$)
- `lexicon_opt.fst`: modèle de lexique permettant de tokéniser des chaînes de caractères (traduction de chaînes de caractères en mots) ($\mathcal{T}$).
- `ascii.syms`: alphabet correspondant à l'ensemble des caractères ASCII
- `full_downcase.txt`: un modèle permettant de transformer les majuscules en minuscules ($\mathcal{M}$).

1. Observer les pondérations du fichier `wotw.lm`. Sur quel semi-anneau est défini le transducteur ?
2. Repérer dans le texte brut issu du livre la première phrase qui commence par *No one would have believed in the last years of the nineteenth.*
3. Vérifier que les mots de cette phrase apparaissent bien dans l'alphabet.
4. A partir du fichier `wotw.lm`, donner le calcul (pour rappel un calcul est une séquence d'états) correspondant au 5-gram *No one would have believed.* Quelle est sa probabilité ?
5. Compiler le transducteur correspondant au modèle de langage 5-grams que l'on appellera `wotw.lm.fst`
6. Une manière de visualiser un chemin de manière aléatoire dans un transducteur est d'utiliser la commande `fstrandgen`. Observer un chemin typique `wotw.lm.fst`

```
fstrandgen --select=log_prob wotw.lm.fst |
     fstprint --isymbols=wotw.syms.txt --osymbols=wotw.syms.txt
```

Exercice 2.

Nous allons maintenant chercher à obtenir la restauration de majuscules la plus probable.

1. Définir clairement les alphabets d'entrée et de sortie des modèles de langage (déjà fait), de tokénisation et de transformation des majuscules en minuscules. Notamment préciser si les alphabets consistent en des caractères, des mots, avec ou sans majuscules.
2. Expliquer ce que va traduire le transducteur suivant: $\mathcal{M}^{-1} \circ \mathcal{T} \circ \mathcal{L}$
3. Réaliser la première composition $\mathcal{L}' = \mathcal{T} \circ \mathcal{L}$. C'est long, non ? Proposer des solutions pour réduire le temps de compilation.
4. Réaliser la seconde composition $\mathcal{M}^{-1} \circ \mathcal{L}'$
5. Tester votre restaurateur avec la séquence *Mars man*.