- *Le but de cette partie est de développer un système de détection d'entités nommées utilisant le classifieur icsiboost.*
    1. *Classification entités nommées au niveau des mots*
        - *Exercice 1 : Découpez le corpus corpus_en_200k.train.txt en 3 parties, l'une pour l'apprentissage (train) contenant 70% des données, l'une pour le réglage des hyperparamètres (dev) contenant 20% des données et enfin une partition d'évaluation (test) contenant 10% des données. Bien sûr ces 3 partitions (train, dev, test) ne doivent avoir aucun recouvrement.*
        - *Exercice 2 : Prédiction du label entité nommée (O , B-person , I-person , B-geoloc , I-geoloc , B-org , I-org , B-product , I-product) au niveau de chaque mot, en reprenant le même principe que pour la classification en POS. Cette fois vous devez prédire les mots qui sont dans la colonne 4 du fichier corpus_en_200k.train.txt, et vous avez le droit d'utiliser des traits de la colonne 3 (les POS) ainsi que tous les traits que vous jugez nécessaire pour représenter le contexte. Par exemple, 2 mots/pos avant et 2 mots/pos après. Vous pouvez utiliser vos propres programmes, ou bien vous inspirer des codes Python donné dans le dossier à la suite du TP. Afficher la courbe d'apprentissage sur les corpus apprentissage/dev/test. Calculez les taux d'erreurs et F-mesure pour chaque étiquette entité nommée en utilisant icsiboost en mode classification avec l'option "-o". Que constatez vous ?*
        - *Exercice 3 : Pour éviter d'avoir autant d'étiquettes 'O', dans cette expérience vous n'allez garder dans vos corpus que les mots qui ont une étiquette qui n'est pas 'O'. Refaites les mêmes expériences qu'à la question précédente, que constatez vous?*
    2. *Détection de candidats entités nommées*
    3. *Maintenant nous allons essayer de limiter le nombre de mots avec le label 'O' sans pour autant "tricher" comme dans la question précédente. Pour cela nous allons sélectionner dans le corpus les séquences de mots susceptibles de représenter une entité nommée. Cette phase de sélection va se faire en utilisant les patrons de POS extraits lors du TP précédents : seront considérés comme "candidates entités nommés", toutes les séquences de mots matchant un patron de POS représentant une entité nommée dans le corpus d'apprentissage.*
        - *Exercice 4 : En utilisant les programmes de la partie précédente, extraire tous les patrons de POS représentant des entités nommées de votre partition train du corpus. Triez ces patrons par fréquence décroissante.*

Exercice 5 : Ecrivez un programme qui prend en paramètre une liste de patrons de POS, un corpus au même format que corpus_en_200k.train.txt, et qui marque toutes les séquences de mots qui correspondent aux patrons en POS passés en paramètres. Pour cela vous rajouterez une colonne dans le corpus envoyé en paramètre qui indiquera la valeur hyp pour les mots faisant partie d'une séquence matchant un patron de POS ou O si le mot ne fait partie d'aucun patron. Par exemple, avec les 3 patrons nc adj / np np / nc prep np, nous obtiendrons :

```
1 le            det  O        O
2 secrétaire    nc   O        hyp
3 général       adj  O        hyp
4 des           prep O        O
5 Nations_unies np   B-org    hyp
6 Kofi_Annan    np   B-person hyp
7 se            clr  O        O
8 rendra        v    O        O
9 lundi         nc   O        hyp
10 à            prep O        hyp
11 Washington   np   B-geoloc hyp
12 pour         prep O        O
13 s'           clr  O        O
14 y            clo  O        O
15 entretenir   v    O        O
16 avec         prep O        O
```

```
17 le           det  O        O
18 président    nc   O        hyp
19 américain    adj  O        hyp
20 George       np   B-person hyp
21 Bush         np   I-person hyp
```

- *Vous pourrez utiliser votre propre programme, ou bien celui donné en exemple.*

4. *Extraction d'entités nommées : détection + classification*
   - *Comme vous pouvez le voir sur l'exemple précédent, certaines séquences d'hypothèses correspondent effectivement à des entités nommées (George Bush), d'autres sont de fausses détections (secrétaire général) et d'autres peuvent contenir des erreurs de segmentation (Nations_unies Kofi_Annan ou lundi à Washington). Il nous faut un moyen de trier ces séquences candidates afin d'extraire les entités nommées et trouver leur type (lieu, personne, organisation). Pour effectuer ce tri nous allons utiliser une approche à base de classifieur supervisé, comme pour les POS, en réutilisant l'outil à base de boosting icsiboost.*
   - *Sur le modèle du classifieur en POS, le but de votre modèle sera de prédire la 4e colonne du corpus corpus_en_200k.train.txt en choisissant un label parmi tous les labels possibles d'entités nommées : O , B-person , I-person , B-geoloc , I-geoloc , B-org , I-org , B-product , I-product. Cependant, au lieu de traiter tous les mots du corpus comme dans la question 2 de ce TP, vous vous limiterez aux mots faisant partie d'une séquence correspondant à un patron de POS d'entités. Cela permet de limiter les exemples négatifs pour l'apprentissage car seule une petite partie des mots du corpus peuvent être des entités nommées.*
   - *Exercice 6 : Transformer les programmes précédents pour qu'il puisse générer des exemples d'apprentissage pour le classifieur icsiboost pour les séquences de mots détectée grâce aux patrons de POS. Vous ferez un premier modèle baseline qui utilisera uniquement des traits relatifs au contexte d'occurrence de l'entité (2 mots avant ; 2 mots après avec leurs POS), ainsi qu'à la forme de l'entité elle-même. Vous rajouterez dans les données au format icsiboost un champs index-ligne qui donnera le numéro de ligne dans le fichier original du mot devant être traité. Etant donné que seuls les mots appartenant à des séquences de POS correspondant aux patrons d'entités sont traités, cela permettra de garder la correspondance avec le fichier corpus de départ.*
   - *Exercice 7 : utilisez maintenant icsiboost pour apprendre et tester le modèle baseline sur vos données. Etudier l'impact du choix des patrons de POS dans la génération des données d'apprentissage du modèle et des performances du modèle de classification. En particulier vous regarderez la répartition des étiquettes à prédire selon le nombre de patrons de POS retenu. Pour les performances de classification, ne vous contentez pas de regardez le taux de bonne classification ! Les mesures qui comptent sont les mesures de précision, rappel et F-mesure sur les étiquettes d'entités nommées (toutes sauf O). Que pouvez-vous en déduire ? Comment évoluent les performances au moment de l'apprentissage entre le train et le dev ? Le comportement est-il similaire à ce que vous aviez observé sur la tâche d'étiquetage en POS ?*
   - *Exercice 8 : vous allez maintenant proposer des modèles avec des ensembles de traits plus riches que le modèle baseline. Parmi les traits possibles, vous pouvez augmenter la taille du contexte, généraliser en utlisant les POS, rajouter des informations sur la graphie du mot (ex: commence par une majuscule), utiliser des lexiques de noms propres, etc.*