IFT6285 (TALN) — Devoir 7 Étiquetage morphosyntaxique avec NLTK

Par Medric B. Djeafea Sonwa (20207022)
Projet: https://github.com/medric49/NLTK_POS_tagging

1. Utilisation de la classe CRFTagger

Nous avons entraîné un tagger sur les données du Penn Tree Bank avec la classe CRFTagger de la librairie NLTK. Ces données sont constituées de 3914 phrases taggées. Pour notre étude, nous les avons divisé un lot d'entraînement de 3000 phrases et un lot de test de 914 phrases.
Le score suite à l'entraînement de ce modèle crf est de **94,74%**.

La fonction d'extraction des features par défaut de la classe CRFTagger ne tient en compte que le token en cours d'analyse. Afin d'apporter des éléments contextuels dans cette liste de features, nous proposons une nouvelle fonction qui, pour un token, ajoute à la liste de features, un indicateur de caractères numériques, un indicateur de ponctuation et le mot des tokens précédent et suivant dudit token.
Nous constatons que l'ajout des propriétés des tokens précédent et suivant ont un effet positif sur les performances du modèle. En effet, le modèle final obtient un score de **95,03%**. Nous avons également constaté que les features les plus bénéfiques dans cette amélioration sont celles provenant du token précédent.

2. Modèle transformationnel

Pour la suite de l'étude, nous entraînons un modèle transformationnel en faisant usage de la classe BrillTagger de NLTK.
Nous utilisons la classe RegexpTagger afin de définir les tags initiaux définis à partir d'expressions régulières. Ces expressions régulières sont tirées des exemples présentés par la librairies :

- Nombres cardinaux (formés de chiffres)
- Articles (*The, the, A, a, An, an*)
- Adjectifs (se terminant par *able*)
- Noms formés à partir d'adjectifs (se terminant par *ness*)
- Adverbes (se terminant par *ly*)
- Noms au pluriel (se terminant par *s*)
- Verbes au gérondifs (se terminant par *ing*)
- Verbes au passé (se terminant par *ed*)
- Noms

Nous obtenons avec cette méthode un score de **45.75%**.
Afin d'améliorer les performances de ce modèle, nous l'avons entraîné en utilisant comme tagger initial le modèle précédemment entraîné.
Ce processus a été répété 130 fois, et le modèle final obtient le score de **76,71%**.
La figure 1 présente l'évolution du score lors de l'entraînement de ce modèle.

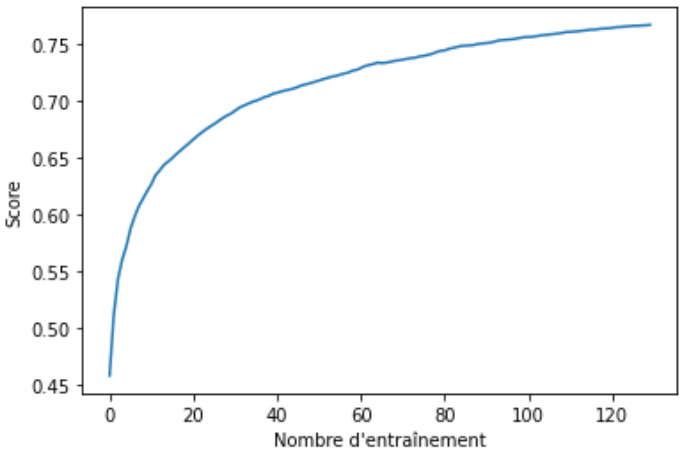

Figure 1. Évolution du score du modèle transformationnel entraîné à partir du tagger basé sur les expressions régulières par défaut définies par NLTK

De la même manière, nous avons mené la même expérience, cette fois-ci en utilisant le modèle crf pré-entraîné sur le jeu de données d'entraînement. Nous avons utilisé ce modèle comme tagger pour un nouveau modèle transformationnel afin d'évaluer ses limites et le comportement du score.
Nous avons effectué cet entraînement 20 fois et le score final obtenu est de 95.19%. La figure 3 présente l'évolution de ses performances.

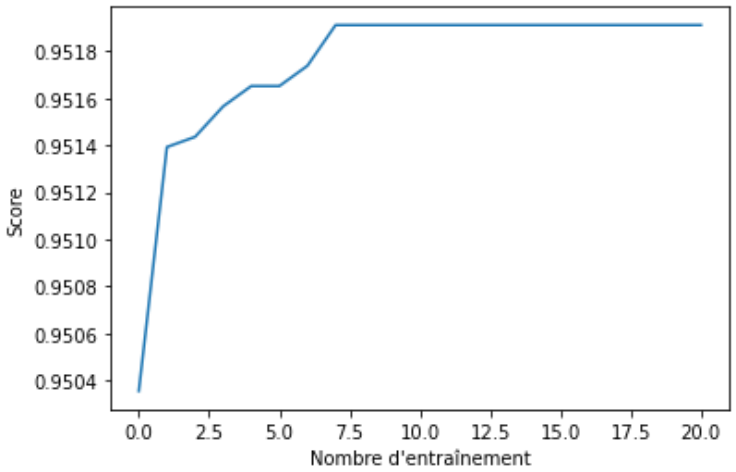

Figure 2. Évolution du score du modèle transformationnel entraîné à partir d'un tagger issu d'un modèle crf pré entraîné

Nous constatons que l'amélioration n'est pas flagrante et cela peut s'expliquer par le fait que que le tagger initial a été entraîné sur le même jeu d'entraînement. Il est donc difficile de faire considérablement mieux que le modèle initial.

3. Spacy Tagger

La dernière étude menée est l'utilisation du tagger de Spacy comme tagger initial d'un modèle transformationnel. Nous avons utilisé ici le modèle *en_core_web__sm* de Spacy pour analyser les phrases et extraire les tags.
Une information importante à mentionner est que ce modèle obtient initialement le score de **88.84%** sur le jeu de test, démontrant qu'il s'agit d'une base assez performante pour le modèle transformationnel que nous souhaitons entraîné.
En fin de compte, après 40 *epochs* d'entraînement, nous obtenons **96%** de score final sur le jeu de test. L'évolution du score de ce modèle pendant l'entraînement est présentée à la figure 3.

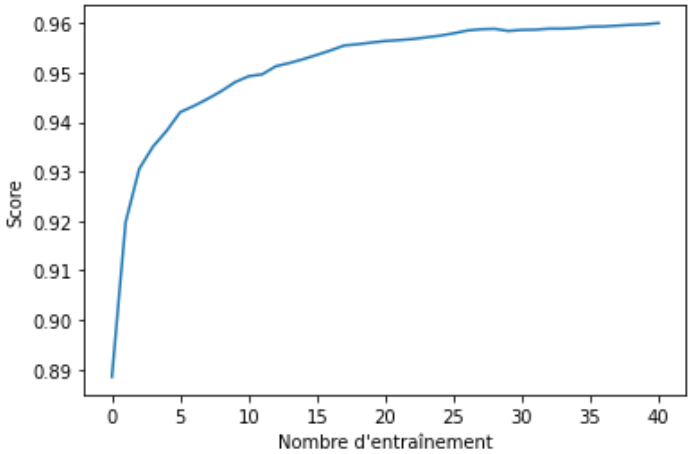

Figure 3. Évolution du score du modèle transformationnel entraîné à partir du tagger du modèle *en_core_web_sm* de Spacy

Ce score nous permet de réaliser à quel point le tagger initial influence les performances finales du modèle transformationnel. En effet, comparé à l'étude de la section 2 utilisant les expressions régulières (un tagger assez limité), nous constatons que le modèle final est nettement plus performant et surpasse le modèle obtenu à partir du tagger crf étudié à la section 2.