CR 30 janvier 2020

Envisager le deep learning => trouver une bonne façon de découper les phrases (faire attention a la simplicité du langage car il y a peu de ponctuation dans ces annonces). Il existe des algo non supervisés qui savent différencier les points dans une phrase. Ici, il faudrait trouver comment faire un algo pour différencier les différents types d'espace (pour aller plus loin par rapport à ce qu'on a déjà fait). Cela permettrait de bien tokeniser nos annonces.

CRF : vaut le coup de refaire un point sur la théorie un peu plus tard car ce n'est pas si trivial
=> Lire un article sur le CRF pour bien comprendre

Problème avec le mot gare => peut être créer une feature pour qu'on puisse faire la différence

Pour muscler ce premier code on peut (cellule 23) faire de la cross-validation et regarder la moyenne des résultats. Permets d'avoir un tableau plus robuste car lissé. Aussi il serait intéressant d'avoir l'écart-type.

Pour dans une semaine / une semaine et demi :
- comprendre la théorie du CRF
- Faire la cross validation chunk 23
- Essayer de comprendre ce qui fait se tromper l'algo et améliorer les feature
- Doccano peut il ingérer des annonces déjà taggées (permettrait d'enrichir l'entrainement)
- Lire tuto B-LSTM

Tokeniser n'est pas fondamental pour l'instant, il serait préférable d'aborder la partie DL (cf repo github). Représenter les mots par des vecteurs afin d'en dériver une représentation du mot dans un vecteur où le mot est un point dans l'espace. Ces donc ce vecteur qu'on injecte dans le CRF. Il serait intéressant de regarder la différence. 2017 LSTM-CRF est dans les modèles les plus puissants.

Comme on a avance tenter un Blstm LNTK => séquence tokeniser, cela nécessite de regarder le split par phrase

RDV fin de semaine prochaine