Use case - Lydia
Danieau Pierre-Louis

Ce problème consiste à évaluer la correspondance sémantique entre un document initial (un libellé d'une transaction) et un document cible (le nom d'un marchand).

Plusieurs méthodes sont envisageables. Nous commençons par présenter celle qui nous paraît être la plus adaptée au problème étudié avant d'en présenter une plus complexe.

1ère méthode : Afin de calculer la similarité entre un libellé d'une transaction et le nom d'un marchand nous pouvons baser nos prédictions sur le TF-IDF (Term Frequency-inverse document frequency). Cette technique est la base de l'évaluation de la similarité de deux documents en fonction de la pertinence des mots qu'ils ont en commun. Pour cela nous allons calculer le score Tf-IDF pour chaque libellé de transactions ainsi que pour chaque nom de marchand. Cette méthode appliquée à l'ensemble des chaînes de caractères disponibles permet d'obtenir une matrice creuse avec un score associé à chaque token. Le score est calculé en fonction de la fréquence d'apparition et de la pertinence du token associé. Avant de calculer ce score, il est important de passer par une étape de preprocessing comme la suppression des stop words dans les documents (le, la, à, et…), la suppression de la ponctuation, des caractères HTML, de la mise en majuscule et la "lemmatization" afin que chaque mot soit réduit à sa forme la plus simple pour faciliter la comparaison. Une fois que nous avons pour chaque document (libellé) et pour chaque nom de marchand un vecteur TF-IDF nous pouvons calculer la similarité d'un couple (libellé / nom de marchand) avec la mesure cosinus. Ainsi nous serons en mesure d'obtenir un score (entre 0 et 1) pour chaque couple. Il ne restera plus qu'à "ranker" les scores pour chacun des couples pour obtenir les marchands les plus susceptibles de correspondre au libellé de la transaction. Cette méthode fonctionne bien notamment si beaucoup de mots sont en commun entre les libellés et les noms de marchands (ce qui est probablement le cas). L'inconvénient de cette méthode est de passer à côté d'une similarité sémantique entre 2 mots qui ne sont pas exactement les mêmes, bien que dans notre cas le risque semble être faible et qu'il est d'autant plus maîtrisé avec une étape de preprocessing.

2ème méthode : Si lors de nos expérimentations nous nous apercevons que cette méthode est trop simpliste, il est possible de mettre en place un modèle plus complexe basé sur la similarité sémantique des libellés et des noms de marchands. Pour cela, il existe des modèles de "word embedding" comme word2vec ou GloVe qui permettent de représenter vectoriellement le sens d'un mot. Une fois ces matrices créées, il sera possible d'appliquer une mesure de similarité entre les libellés et les noms de marchands. Cette méthode est plus complexe a entraîné et peut s'avérer coûteuse en temps de calcul si le dataset est important. D'autant plus qu'elle ne me semble pas essentielle au vu de la nature du problème étudié.

Dans tous les cas, ce problème demandera de la puissance de calcul, c'est pourquoi une architecture distribuée serait préférable.