

# **Learning noun countability using minimally supervised (deep) learning**

**Cindy Aloui**

LIF

**Encadrant(s)**

Carlos Ramisch et Alexis Nasr (co-encadrant)

**Résumé**

Résumé en français obligatoire.

**Mots-clés :** mot-clés obligatoires.

## 1 Introduction

La comptabilité des noms est importante dans plusieurs tâches de traitement automatique du langage, notamment pour la traduction automatique. En effet, certains langages tels que le Japonais n'ont pas de système d'article ni de marque du pluriel, dans ces cas-là, il est utile de connaître la comptabilité des mots pour déterminer les articles à utiliser par exemple en japonais : "gohan o tabemasu" se traduira "je mange du riz" alors que "pasuta o tabemasu" se traduira "je mange des pâtes".

La comptabilité peut aussi être utilisée pour lever l'ambiguïté de certains mots tel que gourmandise qui peut être comptable ou massif, il peut donc être utile de pouvoir déterminer la comptabilité d'un contexte, cela permet toujours en traduction automatique de connaître la traduction la plus pertinente, par exemple savoir si gourmandise sera plutôt traduit greed ou delicacy en anglais. Nous souhaitons donc pour associé un score de comptabilité et de massivité à un nom mais aussi à un contexte. Nous souhaitons aussi que notre système soit généralisable à d'autre classe sémantique telle que concret/abstrait ou animé/inanimé car cela permettrait de désambiguïser plus de mots, pour cela il faudrait que notre système n'ai besoin que d'un minimum de données supervisées pour que l'on puisse l'entraîner pour d'autres classes facilement.

## 2 État de l'art

A faire

## 3 Description des données

Comme dis précédemment, nous souhaitons résoudre ce problème avec un minimum de supervision. Nous avons donc décidé d'utiliser pour l'entraînement seulement une liste de 200 mots typiquement massif et une liste de 200 mots typiquement comptable et du corpus frWaC qui a été annoté par le TreeTagger qui nous donne pour chaque mot sa forme lemmatisée et sa partie de discours.

Dans un premier temps nous avons séparé ces listes en un ensemble de mots pour l'entraînement et un autre pour le test (100 mots comptable et 100 mots massifs pour l'entraînement et la même chose pour le test).

Dans un second temps, nous avons choisi aléatoirement 100 noms qui apparaissent dans le frWaC et ensuite choisi à nouveau aléatoirement pour chacun de ces mots 100 phrases où il apparaît. Nous avons donné cet ensemble de phrases à une linguiste qui pour chaque mot a annoté 50 des phrases extraites et a indiqué si ce nom dans ce contexte précis est plutôt massif, comptable ou ni l'un ni l'autre.

### Exemple :

*C Le [[ balai ]] est tombé dans un trou de plusieurs mètres de profondeur .*

Pour cette phrase le balai dans ce contexte a été annoté comptable.

Ces 5000 phrases vont nous servir lors de l'évaluation de notre classifieur.

## 4 Nom à trouver

Pour réaliser la tâche de classification massif/comptable nous avons penser à trois approches différentes :

- classifier le nom hors contexte en fonction des contextes dans lequel il apparaît (par exemple comme le nom table apparaît dans des contextes comptable, il serait classé en temps que nom comptable)
- classifier le contexte seul sans regarder le nom à classer
- classifier le contexte en utilisant aussi des informations lexicales du mot (éventuellement utilisé la classe du nom hors contexte)

Je vais par la suite entrer plus en détail sur les motivations de chacune de ces approches et expliquer les différentes implémentations.

### 4.1 Classifieur lexical

Nous avons tout d'abord pensé à classifié les mots hors contexte et à appliquer la classe prédite à tout les contexte où ce mot apparaît. Nous pensions que ça permettrait d'avoir une première baseline assez rapidement, car la grande partie des mots sont soit très majoritairement massifs soit très majoritairement comptable, de plus c'est une information que nous pensons utile pour la suite en particulier pour la dernière approche.

#### Word Embedding

Nous avons tout d'abord pensé à utilisé des word embedding qui sont une représentation vectorielle des mots calculer à l'aide des contextes où les mots apparaissent. Cette représentation à la particularité que les mots apparaissant dans des contextes similaires possèdent des vecteurs proches, ce type de représentation semble donc utile pour notre tache de classification. Nous avons donc calculé les word embedding des mots de tout le frWaC, à l'aide de la bibliothèque Python Gensim avec comme paramètres :

- une taille de 200, ce qui est une taille standard
- un fenêtre de 2, c'est à dire que pour l'on créer les word embeddings en prenant comme contexte deux mots avant et deux mots après
- skip-gram comme modèle pour apprendre les embeddings

Nous avons ensuite entraîner un classifieur KNN (Méthode des k plus proches voisins) implémenté dans la librairie Python Scikit-Learn sur les word embeddings des 100 mots massifs et les 100 mots comptables de nos données d'apprentissage. Le nombre k de voisins est déterminé à l'aide d'une validation croisé sur l'ensemble d'entraînement.

Ce système a été évalué de deux manières :

- Nous avons classé les 100 mots comptables et les 100 mots massifs de test, cela donne une précision de 90,3%, cependant il s'agit de mot que l'on considère "facile" à classer.
- Nous avons classé les 100 noms que nous avons choisis aléatoirement et appliqué leur classe prédite à toutes les phrases annoté et calculé la précision en fonction de l'annotation, ce qui donne 68,6% de réussite.

Nous avons testé une autre approche qui consiste pour chaque mot que l'on souhaite classer à faire la moyenne des cosinus pour tout les mots comptable et massifs de l'entraînement et de la classer dans la classe ayant la plus grande moyenne. L'évaluation est faite de la même manière que précédemment est les résultats sont :

- 85% sur les listes de mots massifs et comptables hors contexte.
- 66,9% sur les phrases annotées.

Cette approche ne s'avère pas concluante, cela est dû au peu de données d'apprentissage, en effet les points les plus proches peuvent être en réalité très éloignés, de plus les word embeddings ne sont pas calculés pour cette tâche en particulier, ils ont donc des informations non utiles et deux mots peuvent être proches car leur sens est proche (ils apparaissent donc dans des contextes semblables) mais leurs classes peuvent être différentes. C'est pour cela que nous avons décidé d'abandonner les word embeddings

## **Nom à trouver**

Nous avons ensuite décidé d'utiliser le classifieur de contexte (nous reviendrons sur ce classifieur dans la section 4.2) pour donner un score de massivité aux lemmes. Ce classifieur nous donne un score de massivité associé à un certain contexte, pour un lemme donné nous faisons la moyenne des scores de massivité pour tout les contextes où il apparaît. Ce score de massivité est un score entre 0 et 1, on considère que si ce score est supérieur à 0.5, le mot est massif, s'il est inférieur à 0.5 le mot est comptable. Cette approche donne les résultats suivants :

#TODO

## **4.2 Classifieur de contexte**

Réseau de neurones

## **4.3 Classifieur final**

# **5 Autres applications**

# **6 Résultats**

# **7 Pistes pour la suite**

# **Bibliographie**

# **8 Consignes**

Insérer ici le texte du rapport dans le format et la structure que vous voulez. N'oubliez pas de mettre à la fin du rapport la bibliographie.

Une fois que votre rapport est prêt, produisez-en une version pdf. Relisez-la avec attention. Vérifiez qu'il n'y a pas de problèmes d'accentuation.

Merci d'avoir bien voulu suivre ces instructions.