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

Dans de nombreuses langues dont le français un nom peut être comptable ou massif. Les noms comptables peuvent être "comptés" comme leur nom l'indique. Ils peuvent être au singulier ou au pluriel, par exemple : un stylo, deux stylos. Les noms massifs eux ne peuvent être comptés, ils ont donc seulement une forme singulier comme *eau* ou *sable*.

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" alors que le seul mot changeant dans la phrase est *gohan*(riz) qui devient *pasuta*(pates).

La comptabilité peut aussi être utilisé pour lever l'ambiguïté de certain 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.

On peut aussi utiliser la comptabilité des mots pour détecter des erreurs dans l'utilisation d'article indéfini et de forme pluriel notamment chez les personnes n'ayant pas le français comme langue maternelle. Par exemple si un mot typiquement massif tel que sable se trouve dans un contexte comptable, il s'agit sûrement d'une erreur (Nagata et al., 2006).

Nous pensons que connaître seulement la classe d'un mot hors contexte que l'on appelera la **classe lexical** du mot n'est pas suffisant, en effet, il y a tout d'abord le problème des mots ambigus qui peuvent être massifs ou comptables en fonction du contexte comme gourmandise dont on a déjà parlé précédemment. On peut aussi rencontrer le cas de mots typiquement massif ou comptable qui sont employé dans des contextes ne correspondant pas à leurs classes, par exemple : "Il s'agit d'une **eau** contenant beaucoup de minéraux".

On peut aussi essayé de classifié le "contexte" c'est à dire les mots entourant le nom à classifier, par exemple si un nom est précédé de "un peu de" le contexte est plutôt massif, par contre si un mot est précédé de "plusieurs" le contexte sera comptable. Cependant cette **classe du contexte** ne suffit pas non plus, car un grand nombre de contextes ne sont pas discriminants, par exemple pour les phrase "Le tableau est beau" et "Le sable est beau" le seul mot changeant est le nom à classifié, dans le premier cas le nom est comptable et dans le deuxième cas le nom est massif.

On pense donc avoir besoin pour répondre à un plus grand nombre de problèmes de pouvoir classifier un mot hors contexte, un contexte seul et un mot dans un contexte.

Nous souhaitons aussi que notre système soit généralisable à d'autres grandes classes sémantique telle que concret/abstrait ou animé/inanimé, car cela permettrait de désambiguïser un plus grand nombre de mots, pour cela, il faudrait que notre système soit général et non pas spécialement adapté à la tâche de classification comptable/massif. De plus les annotations en contexte sont très onéreuse on souhaiterais donc limité ces annotations et donc que notre système n'ai besoin que d'un minimum de données supervisé pour que l'on puisse l'en-

traîner pour d'autres classes facilement.

Dans ce mémoire, nous allons utiliser des réseaux de neurones et des plongements de mots pour résoudre ce problème. Ces méthodes d'apprentissages profonds permettent d'avoir des modèle très généralisables et fonctionne très bien pour résoudre un grand nombre de problèmes de traitement de la langue, nous espéreront donc que ces modèles nous aide à résoudre ce problème en particulier. Dans la Section 2, nous allons présenter l'état de l'art dans ce domaine, dans la Section 3, nous présenterons les données, dans la Section 4, nous allons présenter les différents modèles utilisés et présenter les résultats et finalement nous allons conclure et présenter les pistes pour la suite dans la section 5.

2 État de l'art

Comme il existe un certain nombre d'applications, plusieurs chercheurs ont travaillé sur le problème de la comptabilité (pour l'anglais).

Baldwin and Bond [1][2] ont proposé une méthode pour apprendre la comptabilité des mots hors contexte à partir de données d'un corpus annoté. Ils ont distingué 4 classes dont massif et comptable qui sont fortement majoritaire, les noms peuvent avoir différentes classes, gourmandise par exemple serait classé comptable et massif. Ils représentent les noms par un vecteur de caractéristiques (tel que la proportion d'apparition au pluriel de ces noms), ces caractéristiques sont extraites à partir de données annotées. Ils utilisent 4 classifieurs (un pour chaque classe) entraînés grâce à l'algorithme KNN. Ils ont avec ce modèle 94,6% de réussite.

Lapata et Keller [3] ont proposé des modèles web pour certains problèmes du traitement de la langue dont le problème de la comptabilité des noms hors contexte. Ils ont tester un modèle qui prédit la comptabilité des nom en fonction de leur fréquence d'apparition au pluriel et leur fréquence d'apparition précédé de certain determinant. Ils obtiennent 88,62% de réussite sur les mots comptables et 91,53% de réussite sur les mots massifs avec cette méthode.

Pend et Araki [4] ont proposé des modèles Web pour apprendre la comptabilité des mots composés hors contexte. Leur méthode est proche de celle de Lapatta et Keller vu qu'ils utilisent pour classer un nom composé, sa fréquence d'apparition au pluriel ainsi que sa fréquence d'apparition précédée de certains déterminants. Ils ont 89,2% de réussite.

Parler de WSD en français + bootstrapping + article comm

À notre connaissance il n'y a pas de travaux sur la comptabilité des mots français ni de modèle utilisant peu de données annotées et cherchant un modèle généralisable. Les autres travaux portant sur la comptabilité utilisent des informations spécialisées pour définir la comptabilité d'un certain nom comme par exemple la proportion d'apparition de ce nom au pluriel ainsi que sa proportion d'apparition précédée par un article indéfini qui sont des informations très adaptés à cette tache en particulier et ne permettent donc pas de généraliser cette approche à d'autres classes sémantique. De plus, ces travaux tentent uniquement de définir seulement la classe lexicale des mots. Les plus grandes différences entre notre travail et les travaux préexistant sont que nous allons

utiliser des réseaux de neurones et des représentations vectoriels pour cette tâche, ce qui n'avait pas était fait à l'époque de plus nous allons travailler sur très peu de données supervisés.

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. Nous allons aussi utiliser le corpus frWaC qui a était annoté par le TreeTagger qui nous prédit pour chaque mot sa forme lemmatisé et sa partie de discours.

Nous avons ensuite choisi aléatoirement 100 nouveaux noms qui apparaissent dans le frWaC et ensuite choisi à nouveau aléatoirement pour chacun de ces mots 50 phrases où il apparaît. Nous avons ensuite annoter ces phrases en indiquant si le nom dans ce contexte précis est plutôt massif ou comptable.

Exemple :

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

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

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

4 Modèles proposés

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

- définir la classe lexical du nom en fonction des contextes dans lequel il apparaît (par exemple comme le nom table apparaît dans des contextes comptable, il serait classé en tant 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 lexical du nom)

Je vais par la suite entrer plus en détail sur les motivations de chacune de ces approches et expliquer rapidement les différentes implémentations testé et ensuite m'attarder un peu plus sur le système que l'on trouve le plus pertinent.

On a décider de commencer par classifier les mots hors contexte puis ensuite appliquer la classe prédite à tout les contextes où ce mot apparaît. Nous pensions que ça permettrait d'avoir une première baseline assez rapidement, car la grande partie des nom ne sont pas ambiguë, de plus c'est une information que nous pensons utile pour la dernière approche.

Word Embedding

Pour ce faire nous avons pensé à utiliser des plongement de mot (ou *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.

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.

Pour évaluer ce système 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, avec cette méthode nous arrivons à 66,9% de réussite sur les phrases annotées.

Cette approche utilisant les word embedding ne s'avère pas concluante, cela est du 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 calculé pour cette tâche en particulier, ils ont donc des informations non utile 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

4.1 Classifieur de contexte

Réseau de neurones

Le classifieur de contexte étant entraîné à l'aide d'un réseau de neurone, il est important de comprendre l ?idée derrière ce système.

Pour commencer un réseau de neurone est appelé de cette façon car l'idée de base est d'imiter le comportement du cerveau et plus précisément des neurones. On souhaiterais avoir de nombreux neurones qui apprennent chacun un certain nombre de chose.

Un réseau de neurone est composé d'un certain nombre de neurones artificiels. Chaque neurones un un ensemble de poids qui sont mis à jour lors de l ?entraînement du réseau. Les neurones sont organisés en couches qui sont relier entre elles.

On appelle ensemble d'entrainement en ensemble de couple entrée, sortie tel que l'entrée soit un exemple du type de donnée que l'on souhaiterais classifier et la sortie correspond à ce que l'on souhaiterais prédire.

Au début de l'entraînement, les poids des neurones sont initialisés aléatoirement. On donne ensuite au réseau les données d ?entraînement, il va ensuite pour chaque entrée essayer de prédire une sortie, ensuite grâce à l'algorithme de rétropropagation du gradient le système va pouvoir corriger les poids des neurones pour minimiser une fonction de perte. La fonction de perte est la fonction qui permet de calculer la différence entre la prédiction du classifieur et la sortie juste.

Pendant l'entraînement le système peut ajuster les poids des neurones après chaque exemple ou après ensemble d'exemple (appelé *batch of samples* en anglais). Dans ce cas là il faut définir la taille de cet ensemble. Plus la taille est grande plus l'entraînement sera rapide, cependant il est possible que la précision du systeme soit plus faible. On peut aussi entraîner le réseaux sur plusieurs itérations de l'ensemble d ?entraînement.

4.2 Classifieur final

5 Autres applications

6 Résultats

7 Pistes pour la suite

Bibliographie

[1] Timothy Baldwin and Francis Bond. Learning the countability of english nouns from corpus data. In *ACL '03 Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 463–470, 2003.

[2] Timothy Baldwin and Francis Bond. A plethora of methods for learning english countability. In *EMNLP '03 Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 73–80, 2003.

[3] Mirella Lapata and Frank Keller. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing (TSLP)*, Volume 2(Article No.1), Fevrier 2005.

[4] Jing Peng and Kenji Araki. Detecting the countability of english compound nouns using web-based models. In *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*, 2005.