Master 1 Informatique

Rapport de Projet : Traitement Automatique du Texte en IA

Sujet :

Identification des opinions exprimées dans les avis et commentaires d'un ordinateur.

Yann MARTIN D'ESCRIENNE
Yohann TOGNETTI

« Année universitaire 2020 - 2021 »

Table des matières

1 Introduction

1.1 Présentation générale du projet

Dans le cadre de notre cours de Traitement automatique du texte en IA, il nous a été demandé d'effectuer un projet de notre choix impliquant une intelligence artificielle travaillant sur des données textuelles. La restriction sur le sujet se limitant à l'existence d'un jeu de données conséquent.

1.2 Choix du sujet

Notre choix de sujet fut **l'identification des opinions exprimées dans les avis et commentaires d'un ordinateur.** Plus précisément, comment retrouver parmi un ensemble de commentaires les parties de l'ordinateur visées, sur quels aspects et quel en est l'avis général qui en ressort.

2 Description des tâches

Ce sujet est fortement inspiré du « SemEval-2015 Task 12 » sur la partie « Aspect Based Sentiment Analysis (ABSA) : Laptop Reviews » et partage donc le même objectifs avec des simplifications. Les différentes entités et catégories qui vont suivre provienne également des annotations du sujet de SemEval. Chacun des commentaires est fragmentés phrase par phrase afin de faciliter le travail de l'IA.

2.1 Entités

Tout d'abord il s'agit de retrouver dans la phrase le(s) entité(s) ciblée. Elles peuvent être l'ordinateur comme un tout, ses parties physique (clavier, écran..) , des logiciels ou OS (Windows, navigateur, jeux...) ou bien même une compagnie et ses services. (DELL, Apple, le support technique, la livraison..).

Remarque : Rien n'interdit d'avoir plusieurs entités dans la même phrase.

Voici la liste des entités possible :

- DISPLAY (=moniteur, écran),
- CPU (=processeur),
- MOTHERBOARD (=carte mère),
- HARDDISC (=disque dur),
- MEMORY (=mémoire, RAM),
- BATTERY (=batterie),
- POWER_SUPPLY (=chargeur, unité de chargement, cordon d'alimentation, (power) adapteur),
- KEYBOARD (=touche, clavier, pavé numérique),
- MOUSE (=sourie, pavé tactile)
- FANS_COOLING (=ventilateur, système de refroidissement),
- OPTICAL_DRIVES (=lecteur CD, DVD ou Blue-ray),
- PORTS (=USB, HDMI, VGA, lecteur de carte),
- GRAPHICS (=carte graphique, carte video),
- MULTIMEDIA_DEVICES (=son, audio, microphone, webcam, haut-parleur, casque, écouteurs).

2.2 Catégories

Une fois obtenue il faut également trouver sur quelle(s) catégorie(s) de l'entité le commentaire porte. Cela peut être un aspect général , sa prise en main, ses performances, son design et ses fonctionnalités, etc...

Remarque : Là encore, rien n'interdit d'aborder plusieurs catégorie pour la même entité au sein d'une même phrase.

Voici la liste des catégories possible :

- GENERAL,
- PRICE (=prix),
- QUALITY (=qualité),
- DESIGN_FEATURES (=design et fonctionnalités),
- OPERATION_PERFORMANCE,
- USABILITY (=ergonomie, prise en main),
- PORTABILITY (=portabilité),
- CONNECTIVITY (=connectivité),

— MISCELLANEOUS (=divers).

L'entité E et la catégorie C qui s'y rapporte forme ainsi un couple E#C.

Remarque : Il est à noter la possibilité qu'aucun couple E#C ne se rapporte à une phrase d'un commentaire.

2.3 Polarité

Enfin, chaque phrase (et non chaque couple E#C comme dans le sujet de SemEval) se verra attribuer une polarité. Les valeurs de cette polarité sont : **negative** pour les phrases soulignant des défauts ou de mauvais avis, **positive** pour celles qui au contraire mettent en valeur des points ou des avis positif, **neutral** lorsque la phrase ne met pas d'opinion en avant et donne par exemple un conseil et finalement **mixed** pour les textes critiquant un aspect de l'ordinateur mais appréciant un autre.

3 Jeux de données

Les jeux de données se divisent en deux groupes : le jeu de données d'entrée correspondant à ce que l'IA va utiliser pour s'entrainer et celui de test sur lequel les mesures seront effectuées. Tout deux proviennent du site du SemEval.

3.1 Structure des jeux de données

Ce sont tout deux des documents XML possédant la structure suivante (simplifiée) :

```
</sentence>
<sentence id="198:315">
    <text>The keyboard is good and easy to touch type on.</text>
    <Opinions>
        <Opinion category="KEYBOARD#GENERAL" polarity="positive"/>
        <Opinion category="KEYBOARD#USABILITY" polarity="positive"/>
    </Opinions>
</sentence>
<sentence id="198:316">
```

FIGURE 1 – structure du fichier XML

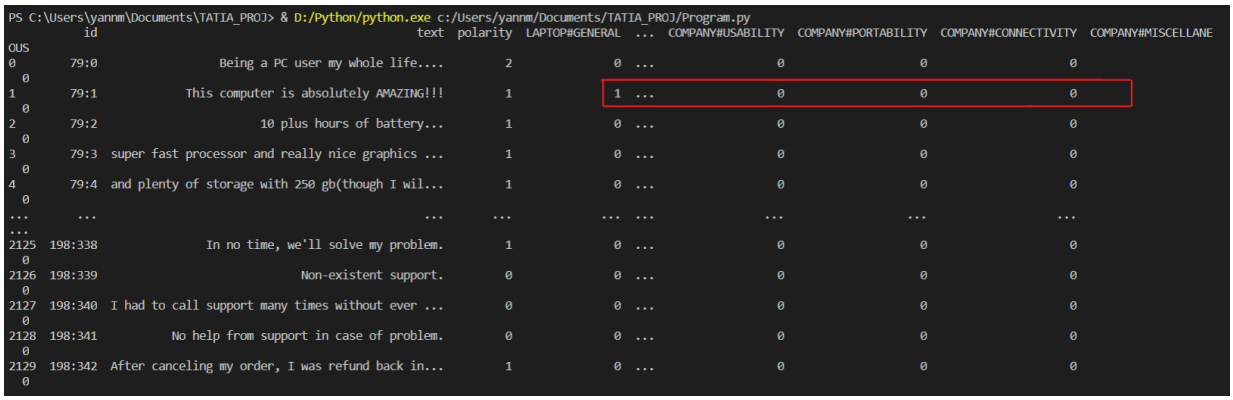
3.2 Modification des jeux de données

Suite à notre implémentation de notre IA qui sera décrite dans le chapitre suivant, de nombreux problèmes nous ont forcés à ajouter nous même des phrases dans le jeu de données d'entrainement.

En effet notre IA répond de la présence ou non de chaque couple E#C dans la phrase actuelle pour chaque combinaison d'entité E et catégorie C. Cela se traduit par une nécessité de nombreuses données appartenant à chacun des couples pouvant apparaitre afin d'obtenir une fonction d'évaluation optimale.

Environ 350 phrases ont été ajoutées, toutes sont la section 'sentences' d'**ID 198**. Certaines proviennent d'avis de consommateur sur les ordinateur portables les plus commentés sur Amazon (en anglais), notamment pour les phrases portant sur les clavier, écran, pavé tactile et batterie. Nous avons rédigés le reste pour les catégories plus complexe et moins abordées en générale. Par exemple la qualité des lecteurs DVD ou bien l'ergonomie des OS.

Remarque : Il est à noté que certain couple n'apparaissent ni dans les données d'entrainement, ni dans les données de test. Par simplification, ces couples ont été ignorés et aucunes phrases n'est étiquetées avec.



```
PS C:\Users\yannm\Documents\TATIA_PROJ> & D:/Python/python.exe c:/Users/yannm/Documents/TATIA_PROJ/Program.py
          id                                          text  polarity  LAPTOP#GENERAL  ...  COMPANY#USABILITY  COMPANY#PORTABILITY  COMPANY#CONNECTIVITY  COMPANY#MISCELLANE
OUS
0       79:0                       Being a PC user my whole life....         2               0  ...                  0                    0                     0
  0
1       79:1                  This computer is absolutely AMAZING!!!         1               1  ...                  0                    0                     0
  0
2       79:2                           10 plus hours of battery...         1               0  ...                  0                    0                     0
  0
3       79:3  super fast processor and really nice graphics ...         1               0  ...                  0                    0                     0
  0
4       79:4  and plenty of storage with 250 gb(though I wil...         1               0  ...                  0                    0                     0
  0
...      ...                                               ...       ...             ...  ...                ...                  ...                   ...
...
2125  198:338                  In no time, we'll solve my problem.         1               0  ...                  0                    0                     0
  0
2126  198:339                              Non-existent support.         0               0  ...                  0                    0                     0
  0
2127  198:340  I had to call support many times without ever ...         0               0  ...                  0                    0                     0
  0
2128  198:341             No help from support in case of problem.         0               0  ...                  0                    0                     0
  0
2129  198:342  After canceling my order, I was refund back in...         1               0  ...                  0                    0                     0
  0
```

FIGURE 2 – dataframe du fichier XML

4 Implémentation de l'algorithme

Notre algorithme est développé en python. Il se divise en 4 fichiers de code ayant chacun un rôle précis dans l'implémentation de l'algorithme d'apprentissage.

4.1 DataInitializer.py

Ce fichier permet de lire un documents XML du jeu de données et regroupe les informations essentielles dans un dataframe de la librairie *panda*. Chacune des phrases correspondent à une ligne dans le dataframe.

Il vient tout d'abord récupérer l'ID et le texte de chaque phrase.

Ensuite il récupère chaque couple E#C et transforme le tout en un vecteur binaire ayant des 1 uniquement aux colonnes correspondant au couple comme sur l'image ci-dessus. Étant donné le nombre de couples possible, il s'agit d'une vecteur à 197 dimensions comportant majoritairement des 0.

Finalement, le programme récupérera la polarité de chaque couple E#C et en conclue une polarité générale de la phrase selon les règles suivantes :

1. **neutral + X = X, X une polarité.**
2. **positive + negative = mixed.**
3. **mixed + X = mixed, X une polarité.**

Cette polarité général est ensuite convertie en entier :
0=negative, 1=positive, 2=neutral, 3=mixed.

4.2 PreProcessing.py

Ce fichier permet d'effectuer un pré traitement sur les textes du premier data frame afin de facilité le travail d'apprentissage de l'IA.

Tout d'abord le texte va être mit en minuscule afin de ne pas différencier des mots n'ayant pas la même casse. Ensuite chacun de ses mots (y compris la ponctuation) va être transformé en « **token** ». De ces tokens sera retiré la ponctuation ainsi que ce que l'on appelle les **STOPWORDS**. Ce sont les mots n'ayant pas de grande valeurs syntaxique comme les déterminants, adjectifs possessifs, conjonctions de coordination, verbes communs (être, avoir)...

L'étape suivante consiste à effectuer une **lemmatisation** sur chacun des tokens. La lemmatisation est un traitement lexical qui consiste à appliquer aux verbes, substantifs, adjectifs... un codage renvoyant à leur entrée lexicale commune, leur « forme canonique ».

Exemple :
Les / la
étoiles /étoile
claires / clair
luisent / luire
noire / noir

```
[2130 rows x 201 columns]
          id                                                text
US
0        79:0                             pc user whole life ....
 0
1        79:1                          computer absolutely amazing
 0
2        79:2                            10 plus hour battery ...
 0
3        79:3  super fast processor really nice graphic card ..
 0
4        79:4       plenty storage 250 gb though upgrade ram ..
 0
...       ...                                                ...
..
2125  198:338                             time 'll solve problem
 0
2126  198:339                               non-existent support
 0
2127  198:340  call support many time without ever getting help
 0
2128  198:341                           help support case problem
 0
2129  198:342               canceling order refund back le time
 0
```

FIGURE 3 – Phrases pré-traitées

Voici une image des phrases du dataframe précédent où celles-ci ont été pré-traitées :

4.3 Classify.py