Introduction au TAL - Projet

HILARICUS Cyrille-Laure
RICHARD Julie
Année 2016-2017

Responsable de cours :
Mme BAWDEN Rachel

FIGURE 1 – source : http ://www.datanalysis.re

Chapitre 1

Le Projet

1.1 Présentation et contexte

Notre projet consiste à créer un système d'analyse des opinions (E-opinions), qui nous permettra d'analyser des reviews clients, ayant acheté des ordinateurs portables.
Á cet effet, nous avons recueilli une série d'avis en anglais, sur le site marchand Amazon, triés en fonction du nombre d'étoiles attribuées à chaque avis (1 étoile correspondant à un produit plutôt mauvais et 5 étoiles pour une excellent produit) attribué par des acheteurs afin d'en tirer une analyse sur le fait que ces reviews soient positives ou négatives à travers notre système.
De façon plus globale, l'idée est de pouvoir traiter un grand nombre de données texte et d'en tirer à travers des systèmes de comparaisons et des algorithmes de probabilité, des observations concrètes sur la qualité perçue par un client de produits high-tech.
Autrement dit, nous voulons grâce à notre système savoir explicitement selon un avis client, si un produit est un bon produit ou pas et sortir de nos analyses, les critères positifs et négatifs qu'ont pu trouver des acheteurs chez certains produits.
Cela nous permettra d'avoir un avis réel (autre que celui technique et commercial) se rapprochant plus des besoins et envies des utilisateurs généralement.

1.2 Choix d'implémentation

1.2.1 Choix stratégiques

— **Informations générales** :

Afin d'avoir une vision claire de la chronologie et des différents processus mis en place dans l'analyse d'opinions.Nous avons procédé par étapes de traitement :

1. Récupération de l'avis stocké sur un disque dur.

2. Traitement du texte, parsing, tagging.

3. Analyse des adjectifs et verbes et comptage de ceux qui représentent à notre sens du négatif ou du positif.

4. Calcul d'un pourcentage de mots négatifs et positifs.

5. Matching des résultats afin de sortir une tendance concernant l'avis analysé.

Tout d'abord, nous partons de l'architecture suivante :

1. Un dossier contenant l'ensemble des avis recueillis.

2. Des sous-dossiers (créés en fonction du nombre d'étoiles qu'a obtenu un produit), dans notre cas nous avons cinq sous dossiers.

3. Des avis stockés sous un format .txt dans les sous-dossiers en fonction du score (en étoile) qu'il a obtenu.

Afin de récupérer ces données nous avons donc ré-utilisé les méthodes vues en TD.

Nous traitons donc un avis, issu d'un dossier au préalablement analysé, qui une fois récupéré par notre système sous forme de liste de mots, une fois cette liste de mots créée, nous avons grâce à NLTK, pris le parti de classer ces mots en fonction de leur nature (adjectifs, noms, adverbes, etc.). Une fois ces mots classés nous avons donc pu les comparer à notre liste exhaustive d'adjectifs et d'adverbes afin de tirer de l'avis analysé, le pourcentage de mots positifs et de mots négatifs ce qui nous permettra de définir de façon intrinsèque la tendance de l'opinion analysé. A savoir si celui-ci est un avis plutôt positif sur le produit ou si au contraire il s'agit d'un avis négatif ou mitigé.

Comment décidons-nous qu'un avis est plutôt positif et qu'un autre non ?

Nous avons décidé avant tout de mettre en place un système de comptage des mots positifs et des mots négatifs dans chacun des avis analysés. Une fois ces nombres recueillis, nous effectuons une moyenne globale de l'ensemble des ces valeurs afin d'en tirer un pourcentage nous indiquant le pourcentage de mots positifs, celui des mots négatifs globalement sur l'avis analysé et nous essayons d'en déduire si il s'agit d'un "bon" ou "mauvais" avis par rapport à l'avis sélectionné.

Notre code source est réparti dans différents fichiers répondant à chacune des étapes citées au préalable :

— Preprocess.py : fichier réalisant les opérations de pré-traitement des données : tokenisation, parsing,tagging.

— Analyze.py : fichier d'analyse de chaque phrase du fichier afin de déterminer si la phrase a un profil tendant vers le positif ou vers le négatif.
Ce fichier Analyze, contient également nos listes exhaustives d'adjectifs et de verbes positifs et négatifs qui nous permettront d'effectuer à travers les fonctions d'Analyze.py une comparaison et donc notre analyseur.

— TrainingData.py : fichier qui récupère les données à analyser et qui renverra après analyse par Analyze.py le pourcentage de points positifs et/ou négatifs.

— main.py : fichier principal qui appelle les fonctions nécessaires au fonctionnement de notre système d'analyse d'opinion.

— **Utilisation de la bibliothèque NLTK** :

Afin de faciliter le développement de notre système d'analyse nous avons fait le choix d'utiliser la bibliothèque de traitement automatique de la langue : **NLTK** afin d'automatiser la tokenization des opinions analysés, mais aussi le tagging des différents mots les composant.
Autre point que nous avons souhaité via NLTK a prendre en compte les synonymes que peuvent avoir les mots analysés.

— **Stratégie d'implémentation de l'analyse des avis** :

Comme stipulé au préalable, l'ensemble de nos fonctions d'analyse des données sont dans notre fichier Analyze.py. Afin donc d'expliquer le fonctionnement de notre système nous allons maintenant détailler la fonction principale de ce fichier : *analyzeFile*.
Cette fonction fait appel à d'une part à des fonctions NLTK telles que *word_tokenize* et *pos_tag* mais aussi a

notre fonction principale de déduction *getPosAndNeg* : permettant d'ajouter les adjectifs positifs et négatifs à notre compteur de mots.

— Calcul du pourcentage. Nous avons créer une fonction nommée *getBestMatch* qui nous permet de déduire la tendance qui ressort de la review analysée.

Afin d'effectuer cette analyse, notre fonction getBestMatch fait appel à une notre fonction de notre fichier : *posAndNeg*.
Dans *posAndNeg*s, On suppose que la négation qui accompagne un adjectif ou un verbe est toujours placée avant celui-ci.

Fonctionnement de getBestMatch :

Pour chaque catégorie, on calcule la différence entre les données apprises et celles que l'on teste :

— PosEnt - posTest.

— NegEnt - negTest.

Une fois ce calcul effectué, on en fait une moyenne :

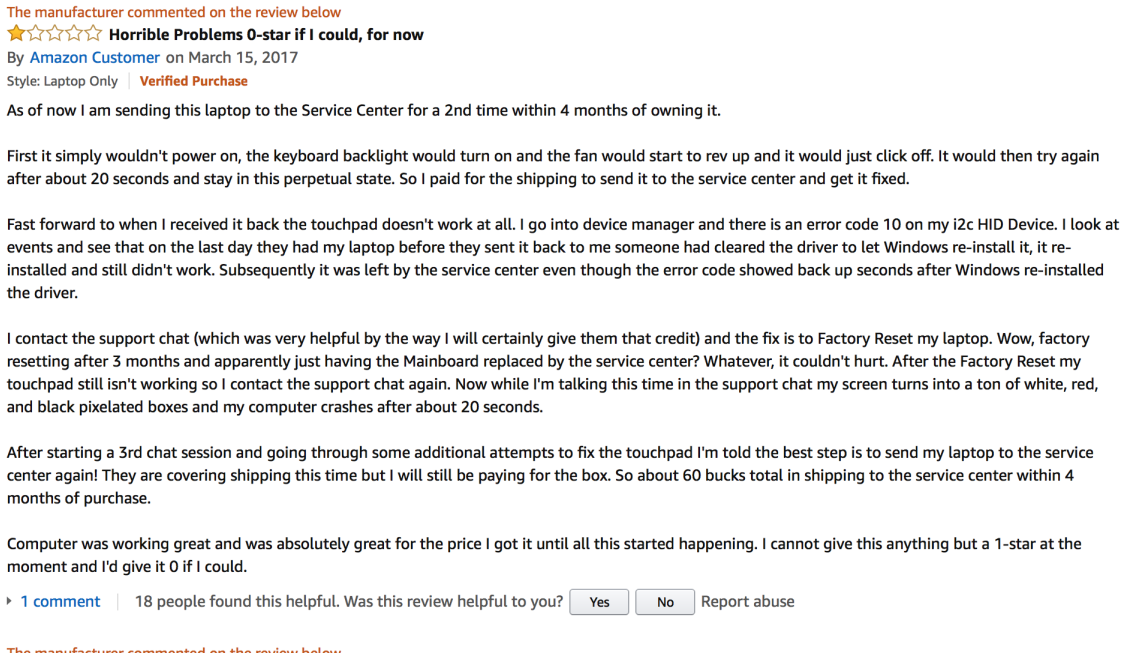
$\frac{(PosEnt - posTest + NegEnt - negTest)}{2}$.

Si le fichier a été évalué comme plus positif que négatif, on regarde la partie supérieur des catégories, sinon, l ?inverse. On cherche la moyenne la plus basse et la catégorie correspondant à cette moyenne est choisie pour le fichier test.

Par exemple : si on a évalué qu ?il y a pos=20 neg = 1, on regarde les moyennes (calculées juste avant) des catégories 3 à 5. On choisit la moyenne la plus petite entre les cat 3, 4, 5.

— **Exemples de fonctionnement** :

Les captures d'écrans ci-dessous représente un des avis analysé récupéré sur le site marchand Amazon, il s'agit d'un article noté par le client 1/5, en analysant cet avis que nous avons récupéré en format .txt, on obtient bien une analyse négative de l'avis.



The manufacturer commented on the review below

Horrible Problems 0-star if I could, for now

By Amazon Customer on March 15, 2017

Style: Laptop Only | Verified Purchase

As of now I am sending this laptop to the Service Center for a 2nd time within 4 months of owning it.

First it simply wouldn't power on, the keyboard backlight would turn on and the fan would start to rev up and it would just click off. It would then try again after about 20 seconds and stay in this perpetual state. So I paid for the shipping to send it to the service center and get it fixed.

Fast forward to when I received it back the touchpad doesn't work at all. I go into device manager and there is an error code 10 on my i2c HID Device. I look at events and see that on the last day they had my laptop before they sent it back to me someone had cleared the driver to let Windows re-install it, it re-installed and still didn't work. Subsequently it was left by the service center even though the error code showed back up seconds after Windows re-installed the driver.

I contact the support chat (which was very helpful by the way I will certainly give them that credit) and the fix is to Factory Reset my laptop. Wow, factory resetting after 3 months and apparently just having the Mainboard replaced by the service center? Whatever, it couldn't hurt. After the Factory Reset my touchpad still isn't working so I contact the support chat again. Now while I'm talking this time in the support chat my screen turns into a ton of white, red, and black pixelated boxes and my computer crashes after about 20 seconds.

After starting a 3rd chat session and going through some additional attempts to fix the touchpad I'm told the best step is to send my laptop to the service center again! They are covering shipping this time but I will still be paying for the box. So about 60 bucks total in shipping to the service center within 4 months of purchase.

Computer was working great and was absolutely great for the price I got it until all this started happening. I cannot give this anything but a 1-star at the moment and I'd give it 0 if I could.

1 comment | 18 people found this helpful. Was this review helpful to you? Yes No Report abuse

FIGURE 1.1 – Avis site marchand

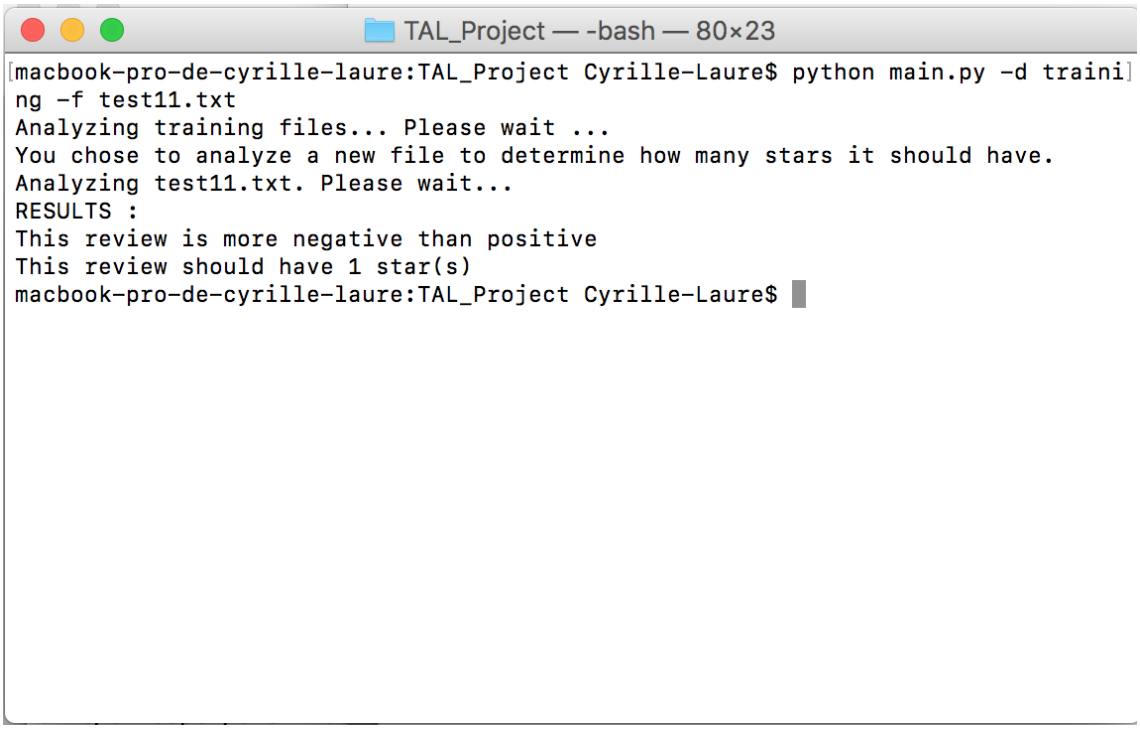


FIGURE 1.2 – Analyse

— **Fichier de statistiques** :

En tenant compte des tests situés dans notre dossier training, nous avons réalisé un tableau de verification de notre programme, celui-ci indique en rouge les analyses échouées (qui ne reflètent donc pas la vraie tendance de l'article), en vert celles correctes et en orange celles se rapprochant d'une bonne analyse. Nous avons également analysé la précision de notre méthode.

Ci-dessous, notre tableau de statistiques :

Fichier	Nb Etoile(s)	Résultats			Différence avec le bon résultat		
		Méthode 1	Méthode 2	Méthode 3	Méthode 1	Méthode 2	Méthode 3
11	1	1	1	1	0	0	0
12		1	1	1	0	0	0
13		1	1	1	0	0	0
14		2	1	1	-1	0	0
21	2	3	4	4	-1	-2	-2
22		5	1	1	-3	1	1
23		3	4	4	-1	-2	-2
24		3	4	4	-1	-2	-2
31	3	1	1	1	2	2	2
32		1	1	1	2	2	2
33		3	4	4	0	-1	-1
34		5	4	4	-2	-1	-1
41	4	1	1	1	3	3	3
42		1	4	4	3	0	0
43		3	4	4	1	0	0
44		1	4	4	3	0	0
51	5	5	1	1	0	4	4
52		2	1	1	3	4	4
53		3	4	4	2	1	1
54		3	4	4	2	1	1

Bien trouvé	5	7	7	Précision	0,6	0,5	0,5
Plus ou moins 1	5	5	5				
Pas trouvé	10	8	8				

FIGURE 1.3 – Fichier de stats

1.2.2 Difficultés rencontrées

— Difficultés générales :

De part l'organisation qui résulte du langage Python, nous nous sommes retrouvées avec des listes imbriquées. Cela nous a contraint à implémenter un certain nombre de boucles for afin d'accéder aux données des sous listes mais aussi de prendre un certain temps en amont avant de coder afin de bien comprendre la façon selon laquelle les données ont étés stockées.
En effet, dans notre structure d'analyse de départ, nous avons des dossiers d'avis rangés par le nombre d'étoiles attribuées à un produit, cela se traduit dans notre système par la récupération d'une liste d'étoile qui elle même contient une liste de fichiers (par exemple pour les articles notés une étoile, on retrouve donc tous les avis ayant reçus un étoile), cette liste de fichier ensuite est décomposée en liste de "sentences" puis de "chunk" puis de "tagword", cela fait un ensemble de listes imbriquées volumineux qu'il est bien nécessaire de comprendre afin de pouvoir accéder au bon niveau lors de l'analyse des données.

Nous avons également rencontré des difficultés à traiter des mots pouvant avoir un caractère positif ou négatif. Par exemple le mot *crazy* est plutôt difficile à classer dans une catégorie bien précise car on peut l'utiliser soit pour exprimer quelque chose de positif soit de négatif.

De plus, à un stade avancé nous nous sommes rendues compte que dans beaucoup des avis récupérés, les utilisateurs utilisaient des abréviations afin de nommer noms propres, des noms d'entreprise qui sont en fait non communes et donc non reconnues par NLTK par exemple "Best Buy" souvent abrégé en "BB" ce qui lors du tagging à entraîner une mauvaise prise en charge de ces informations.
Enfin, l'utilisation de *synset* de la bibliothèque NLTK pour nous permettre d'obtenir les synonymes des mots issus des avis analysés, ne prend en charge les synonymes que de 3 catégories de mots notamment les adjectifs et les verbes et ne prend donc pas en compte des cas de tagging tels que les adverbes ou autre.

— Problème de comparaison entre les mots :

L'un des gros problèmes rencontrés est celui de l'incapacité de notre système a pouvoir comparer tous les mots, une erreur au compilateur nommée *Unicode equal comparison failed to convert both arguments* que nous avons eu à la dernière minute nous a permis de constater que notre analyseur ne peut à cause d'une incompatibilité faire toutes les comparaisons de mots nécessaires afin de mener à bien l'analyse.

Ceci fut essentiellement les difficultés, autre que celle du temps qui ne nous ont pas permis de faire un système plus intuitif que celui que nous proposons actuellement.

1.3 Perspectives

Notre système actuel ne permet que de sortir un pourcentage reflétant les "points positifs" et les "points négatifs" de l'avis analysé, une amélioration que nous pourrions mettre en place est de fournir après analyse une liste des points positifs et une listes des points négatifs sur un produit par avis voire globalement. Afin de mettre en place cette idée, il nous aurait été nécessaire de stocker les adjectifs d'une part positifs et d'un autre côté négatif accompagnés des compléments d'objets et noms qui y sont liés.

1.4 Conclusion

Avant d'utiliser la bibliothèque **NLTK** qui nous permettait d'automatiser un certains nombre de pré-traitements tel que la tokenisation et le tagging, nous avions débuté l'implémentation d'une version manuelle de ces tâches ce qui nous a fait perdre du temps sur notre développement. Cependant celui à été par la suite comblé grâce à la mise en place du processus d'automatisation via NLTK.
Nous avions pris le parti d'effectuer une analyse globale de la tendance de l'article analysé ce qui globalement fonctionne.

Grâce à ce projet, nous avons pu développer des compétences nous permettant de traiter et d'analyser un grand nombre de données écrites, ce qui s'avère très utile dans le monde d'aujourd'hui en informatique où de plus en plus de données sont informatisées (Big data) et nécessitent un traitement automatisé afin de facilité l'interprétation des données. Nous pouvons sans forcément lire un avis savoir si il en ressort une critique positive ou négative concernant un produit réellement, ce type d'outils peut être utile pour des sites de vente en ligne afin de réellement cerné les avis postés par des acheteurs en ligne.

Notre système qui, à la base permet de traiter des avis pourrait s'appliquer dans d'autres domaines par exemple pour l'analyse des sentiments, nous pourrions analyser les propos d'un individu et grâce à notre méthode agrémenter de quelques algorithmes déterminer le sentiment que peut avoir celui-ci.