TPSA: Conception et implémentation d'un algorithme d'analyse de sentiment basé sur les aspects

Le terme "analyse des sentiments" est utilisé pour désigner la tâche consistant à déterminer automatiquement la polarité d'un texte, qu'il soit positif, négatif ou neutre. L'analyse de sentiment est de plus en plus considérée comme une tâche essentielle, tant d'un point de vue académique que commercial. La majorité des approches actuelles, cependant, tentent de détecter la polarité globale d'une phrase, d'un paragraphe ou d'un ensemble de mots, indépendamment des entités mentionnées (par exemple, les ordinateurs portables, les restaurants) et de leurs aspects (par exemple, la batterie, l'écran ; la nourriture, le service). Le sentiment peut être déterminé à différents niveaux : le sentiment associé aux mots ; le sentiment associé aux phrases, aux SMS, aux messages, dans les chats et aux tweets ; aux sentiments dans des revues de produits, des articles de blog et des documents entiers.

Le but du TPSA est de concevoir et implementer un algorithme de classification de sentiment basé sur les aspects. Plus en detail, ce TP vise à identifier les aspects des entités cibles données et le sentiment exprimé à l'égard de chaque aspect. Des ensembles de données comprenant des commentaires de clients avec des annotations par des annotateurs humains identifiant les aspects, les entités cibles et la polarité du sentiment de chaque aspect sont fournis.

Les données annotées :

1. Les phrases du jeu de données sont fournies en format `XML`.

Le jeu de données `Restaurant_Train_v0.2` est composé de 3041 phrases en anglais tirées des critiques de restaurants [Ganu et al, (2009)]. Le jeu de données pour inclus des annotations pour les termes d'aspects apparaissant dans les phrases, les polarités des termes d'aspects et les polarités spécifiques aux catégories d'aspects. Des annotateurs humains expérimentés ont identifié les termes d'aspect des phrases et leurs polarités.

```
<sentence id="813">
        <text>All the appetizers and salads were fabulous, the steak was mouth watering and
        <aspectTerms>
                <aspectTerm term="appetizers" polarity="positive" from="8" to="18" />
                <aspectTerm term="salads" polarity="positive" from="23" to="29" />
                <aspectTerm term="steak" polarity="positive" from="49" to="54" />
                <aspectTerm term="pasta" polarity="positive" from="82" to="87" />
        </aspectTerms>
        <aspectCategories>
                <aspectCategory category="food" polarity="positive" />
        </aspectCategories>
</sentence>
```

Les valeurs possibles pour la polarité des aspects sont : "positive", "negative", "conflict", "neutral". Les valeurs possibles des categories sont : "food", "service", "price", "ambience", "anecdotes/miscellaneous".

Note : les données sont annotées aussi par rapport à la polarité "conflictuelle", *c'est-à-dire à la fois positive et négative, mais nous ne prendrons pas en compte cette polarité. On se concentrera sur la polarité positive, négative ou neutre.*

Veuillez noter que :

- Toute citation dans un terme d'aspect (par exemple, "sales" team) a été remplacée par `&quot ;` (le texte et les offsets restent les mêmes), par exemple, `<aspectTerm term="&quot;sales&quot; team"./>`.
- Les phrases peuvent contenir des fautes d'orthographe.
- Pour chaque terme d'aspect des données d'entrainement, nous incluons deux attributs ("de" et "à") qui indiquent son décalage de début et de fin dans le texte (par exemple, `<aspectTerm term="staff" polarity="negative" from="8" to="13"/>`).

Première tâche : analyse des sentiments du jeu de données sur les restaurants et les ordinateurs.

On commence par télécharger les fichiers suivants qui se trouvent dans le repertoire TPSA/datasets sur Moodle :

- Restaurants_Train.xml
- Restaurants_Test_Gold.xml
- Restaurants_Test_NoLabels.xml

Objectif 1: calculer la polarité des mots dans les deux jeux de données à l'aide d'un lexicon de sentiment.

Avant d'utiliser le lexicon pour calculer la polarité des mots contenus dans les phrases dans les deux jeux de données, il est nécessaire de faire un **prétraitement des phases** (negation, tokenizer, PoS tagger, et NER).

Vous devez *implémenter* un système d'extraction d'informations simple. Le texte brut de chaque phrase est subdivisée en mots à l'aide d'un **tokenizer**. Ensuite, chaque phrase est étiquetée avec des balises de partie de discours (**PoS tagger**), ce qui s'avérera très utile à l'étape suivante, la détection d'entités nommées (**NER**).

> Pensez à stocker toutes les informations extraites, elles vous seront utiles par la suite !

Vous pouvez choisir d'utiliser le tokenizer, PoS tagger, et NER de :

- NLTK
  - https://www.nltk.org/book/ch03.html#chap-words
  - https://www.nltk.org/book/ch05.html#chap-tag
  - https://www.nltk.org/book/ch07.html
- SPACy
  - https://spacy.io/api

Une fois que le pré-traitement des phrases est terminé, vous pouvez télécharger le lexicon SentiWordNet (https://github.com/aesuli/SentiWordNet) ou le lexicon RC Word-Emotion Association Lexicon (EmoLex) (https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm).

Voici un extrait du code dont vous aurez besoin pour extraire les valeurs de sentiment des mots individuels. Dans ce code, il est fait référence aux TAG PoS NLTK de chaque mot de la phrase.

```
def  penn_to_wn(tag):
        """Convertion des tags en simple WORDNET TAGS"""
        if  tag.startswith('J'):
                return wn.ADJ
        elif  tag.startswith('N'):
                return wn.NOUN
        elif  tag.startswith('R'):
                return wn.ADV
        elif  tag.startswith('V'):
                return wn.VERB
        return  None
lemmatizer  =  WordNetLemmatizer()

def  get_sentiment(word,tag):
        '''
        Return une liste de score positif negatif ou neutre et return une liste vide si le mo
        '''
        wn_tag  =  penn_to_wn(tag)
        if  wn_tag  not  in  (wn.NOUN,  wn.ADJ,  wn.ADV):
                return  []

        lemma  =  lemmatizer.lemmatize(word,  pos=wn_tag)
        if  not  lemma:
                return  []

        synsets  =  wn.synsets(word,  pos=wn_tag)
        if  not  synsets:
                return  []
        # Prend le premier sens du mot c'est à dire le sens le plus commun
        synset  =  synsets[0]
        swn_synset  =  swn.senti_synset(synset.name())
        return  [swn_synset.pos_score(),swn_synset.neg_score(),swn_synset.obj_score()]
```

Apres le téléchargement, vous devez identifier la polarité associé a chaque mot dans les phrases contenues dans les jeux de données (fichiers Train et Test, 4 fichiers à traiter) en utilisant le lexicon SentiWordNet ou EmoLex:

- pour chaque mot (que vous avez identifié avec le tokenizer, stop words exclues) vous cherchez si le mot est present dans le lexicon.
  - S'il est present, alors vous assignez à ce mot la polarité positive/negative associée au mot dans le lexicon ansi que le degré associé. A vous de choisir le format (balises) pour stocker ces informations, qui vous seront utiles après.
  - S'il n'est pas present, vous pouvez passer au mot suivant.

> Enfin, vous pouvez générer une visualisation des données à travers des graphiques pour montrer combien de mots ont une polarité positive / negative dans chaque fichier.

Objectif 2 : Determiner la polarité des termes des aspects

En partant des jeux de données annotées avec les termes des aspects et leur polarité, il faut concevoir un algorithme *capable de déterminer automatiquement* si la polarité de chaque terme d'aspect est positive, négative ou neutre.

Exemple :

- "I loved their fajitas" → {fajitas: positive}
- "I hated their fajitas, but their salads were great" → {fajitas: negative, salads: positive}
- "The fajitas are their first plate" → {fajitas: neutral}

Le fichier `Restaurants_Train.xml` sont a utiliser pour la phase d'entrainement. Vous avez que les aspects sont déjà identifiés dans les fichiers d'entrainement.
Par exemple :

```
<text>All the appetizers and salads were fabulous, the steak was mouth watering and the pasta
<aspectTerms>
        <aspectTerm term="appetizers" polarity="" from="8" to="18" />
        <aspectTerm term="salads" polarity="" from="23" to="29" />
        <aspectTerm term="steak" polarity="" from="49" to="54" />
        <aspectTerm term="pasta" polarity="" from="82" to="87" />
</aspectTerms>
```

Le fichier `Restaurants_Test_NoLabels.xml` sont à utiliser pour tester votre algorithme.

Les annotations produites par votre algorithme sont a comparer avec celles contenues dans le fichier `Restaurants_Test_Gold.xml`.

Par exemple :

```
<text>All the appetizers and salads were fabulous, the steak was mouth watering and the pasta
<aspectTerms>
        <aspectTerm term="appetizers" polarity="positive" from="8" to="18" />
        <aspectTerm term="salads" polarity="positive" from="23" to="29" />
        <aspectTerm term="steak" polarity="positive" from="49" to="54" />
        <aspectTerm term="pasta" polarity="positive" from="82" to="87" />
</aspectTerms>
```

Pour implementer votre algorithme d'analyse de sentiment basée sur les aspects vous pouvez prendre en compte les elements suivants :

2. Le but de votre algorithme n'est pas de classifier le sentiment de la phrase (par exemple "All the appetizers and salads were fabulous, the steak was mouth watering and the pasta was delicious!!!"), mais de déterminer le sentiment relatif aux aspects identifiés dans la phrase (par exemple, "appetizers", "salads", "steak"). Une stratégie possible consiste à chercher dans la phrase l'aspect donné en entrée et prendre en compte les mots qui entourent cet aspect (dans une fenêtre $t = n$ et $t+n$ ou t est la position du mot relatif à l'aspect sur lequel on se focalise). Il y a différents façons de choisir n, par exemple d'une façon empirique (en se basant sur les données du gold standard) ou en se basant sur le résultat d'une parsification de la phrase.

3. Determiner la polarité des termes en utilisant le lexique, comme demandé dans l'objectif 1.

4. Deux solutions sont possibles pour *calculer* le sentiment des aspects en entrée :

   - **Approche a règles** : un stratège de base (baseline) consiste à sommer les polarités des mots dans la phrases en utilisant les ressources lexicales dédiées a cette tache (voir point 3) et normaliser cette somme. Un approche a règles vise a améliorer cette stratégie de base avec des règles qui sont déterminées explicitement pour les revues contenues dans les deux jeux de données.
   - **Apprentissage automatique** : vous pouvez utiliser ces resources lexicales avec les elements relevant de la structure syntaxique de la phrase (negation, tokenisation) comme features pour classifier les termes avec leur polarité : positive, negative, neutral. Vous pouvez utiliser la suite scikit-learn (https://scikit-learn.org/stable/index.html) pour effecter cette tache de classification.

5. Evaluation de vos résultats (comme pour le TD de QA) :

$$\textbf{précision}_i = \frac{\textbf{nb de documents correctement attribués à la classe } i}{\textbf{nb de documents attribués à la classe } i}$$

$$\textbf{rappel}_i = \frac{\textbf{nb de documents correctement attribués à la classe } i}{\textbf{nb de documents appartenant à la classe } i}$$

$$F = 2 \cdot \frac{(\textbf{précision} \cdot \textbf{rappel})}{(\textbf{précision} + \textbf{rappel})}$$

Pour avoir une idée des résultats entendus, voici les résultats des systèmes courants pour cette tache :

```
Nom du système  Accuracy
NRC-Can.                82.92
XRCE                    78.14
UNITOR                  76.29
```

ou

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Nous n'attendons pas que vous arrivez a faire mieux, mais ces informations sont utiles pour vous comparer et savoir combien les résultats obtenus sont loin de l'état de l'art.