TP2: Análisis de Sentimiento

Métodos Numéricos

12 de octubre de 2018

Dataset de reseñas

El conjunto de datos a utilizar está basado en el *Large Movie Review Dataset* el cual fue creado para el trabajo de Maas et al.[1]. Consta de 50.000 reseñas de películas de obtenidas de IMDB, segmentadas en positivas (aquellas que tuvieron puntaje mayor a 7 estrellas) y negativas (puntaje menor a 4).

El dataset está partido exactamente en dos: 25.000 instancias de entrenamiento, y otras 25.000 instancias de testeo. A su vez, las instancias de entrenamiento y testing tienen la mitad de reseñas positivas, y la otra mitad negativas.

Tokenizado

La *tokenización* [1] consiste en cortar el texto en unidades llamadas *tokens*, a la vez que tiramos ciertos caracteres (por ejemplo, algunos signos de puntuación). Esta tarea, que a priori puede parecer trivial, tiene sus bemoles: por ejemplo, no siempre tenemos que separar por signos de puntuación o espacios: a veces tenemos siglas ("A.F.A.") o contracciones ("Mr.", "Dr.").

Si bien uno pueden identificar a priori un *token* con una palabra, esto no es siempre así: por ejemplo, en nuestro dataset tenemos tokens como "sci-fi" o "low-budget" que tienen sentido para nuestro trabajo.

Para facilitar el trabajo, les brindamos el texto ya tokenizado, como explicamos a continuación.

Listado de archivos provistos Todos los archivos de entrada/salida deberán estar en el formato `.csv`, el cual es un formato de archivo de texto muy sencillo para representar matrices, donde cada línea representa una instancia y cada dato está separado por una coma.

- `imdb_dataset.csv`: Este archivo tiene como columnas:
  `index,type,review,label`
  - *index* es el índice al texto de la reseña
  - *type* si es de train o test
  - *review* el texto en sí
  - *label* su polaridad: "neg" o "pos"

[1] `https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html`

- `vocab.csv`: Tiene las siguientes columnas:
  `word,count,num_documents,document_frequency,code`
  - *word* es la palabra
  - *count* cantidad total de veces que aparece en el dataset
  - *num_documents* cantidad de documentos en los cuales aparece la palabra
  - *document_frequency* es el valor anterior dividido por N, donde N es la cantidad de instancias = 50.000
  - *code* un número único con el cual se identifica a la palabra en la tokenización
- `imdb_tokenized.csv`: Tiene para cada línea:
  `index,type,label,token_list`
  - *index* es el índice al texto de la reseña
  - *type* si es de train o test
  - *label* su polaridad: "neg" o "pos"
  - *token_list* es una lista de enteros representando el texto de la reseña mediante tokens
    Importante: notar que la lista de tokens para cada documento será de longitud variable y su fin estará dado por una nueva línea `\n`.

Por ejemplo, la siguiente reseña es la número 0 en el archivo `imdb_dataset.csv`, es de tipo test y con polaridad negativa:

```
0, test, "Once again Mr. Costner has dragged out a movie for far longer
    than necessary. Aside from the terrific sea rescue sequences, of which
     there are very few I just did not care about any of the characters.
    Most of us have ghosts in the closet, and Costner's character are
    realized early on, and then forgotten until much later, by which time
    I did not care. The character we should really care about is a very
    cocky, overconfident Ashton Kutcher. The problem is he comes off as
    kid who thinks he's better than anyone else around him and shows no
    signs of a cluttered closet. His only obstacle appears to be winning
    over Costner. Finally when we are well past the half way point of this
     stinker, Costner tells us all about Kutcher's ghosts. We are told why
     Kutcher is driven to be the best with no prior inkling or
    foreshadowing. No magic here, it was all I could do to keep from
    turning it off an hour in.",neg,0_2.txt
```

El documento 0 se tokeniza como (ver archivo `imdb_tokenized.txt`):

```
0,test,neg,277,174,578,7118,48,3325,49,3,17,16,225,1113,77,1686,1122, ...
```

Donde 277 es la palabra "once" en el vocabulario (ver `vocab.csv`), 174 es la palabra "again", etc.

Referencias

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.