




Modelo de Lenguaje y Sistema de Recomendación

Se desea crear un sistema para recomendar películas. El archivo `movies.csv` posee una base de datos donde usuarios calificaron (del 1 al 5) diferentes películas (0 significa sin calificar).

(a) *Modelo de Lenguaje*: Se desea diseñar un buscador de títulos de películas, de manera que si el usuario comete algún error u omisión cuando lo escribe, el buscador pueda entender. Para ello, se estudiará la similitud entre los *embeddings*. A continuación se describen los pasos para diseñar el buscador; se recomienda que los mismos sean métodos dentro de la clase del buscador mencionado.

- Descargar las representaciones pre-entrenadas *GloVe* de dimensión 300. : El modelo pre-entrenado puede descargarse en <http://nlp.stanford.edu/data/glove.6B.zip>
- Cargar el modelo de lenguaje. : Posible código (adaptar a sus necesidades)

```
language_model = {}  
with open("glove.6B.300d.txt", encoding="utf-8") as f:  
    for line in f:  
        parts = line.strip().split()  
        word = parts[0]  
        vec = np.array(parts[1:], dtype=float)  
        language_model[word] = vec
```

- Implementar un *word2vec*. Si la palabra está en el vocabulario debe devolver el vector del modelo de lenguaje, caso contrario debe devolver un vector de ceros.
- Implementar una *bolsa de palabras* que transforme cualquier *string* en un vector. Los pasos a seguir son:
 - Convertir las mayúsculas en minúsculas.
 - Eliminar caracteres extraños.
 - Unificar espacios en blanco.
 - Convertir el *string* en una lista de palabras.
 - Convertir cada palabra en un *embedding* usando el *word2vec*.
 - Sumar las representaciones para formar un solo vector.
- Se desea medir que tan parecidos son dos *embeddings*. Para ello, implementar un código que calcule la *similitud coseno*. : La similitud coseno se define como el coseno del ángulo entre dos vectores $\mathbf{SC}(u, v) = \frac{u \cdot v}{\|u\| \|v\|}$.
- Implementar un buscador que, dado un *string* (y su correspondiente *embedding*), devuelva la película con una representación más similar.

(b) *Sistema de Recomendación*: Se desea diseñar el sistema de recomendación y utilizarlo para recomendarnos películas.

- Agregar un usuario a la base de datos con al menos 10 películas calificadas. Utilice el buscador para no tener que escribir los títulos perfectos.
- Utilizando *gradiente descendente* entrenar un filtro colaborativo con un espacio latente de dimensión 10, $\lambda = 10$ y *learning rate* 10^{-3} . Graficar el riesgo regularizado empírico en función del número de iteraciones (al menos 2000).
- Crear un *rating* ponderando en partes iguales la salida del filtro colaborativo y la calificación media de las películas.
- Recomendar las 5 películas **no vistas** con más alto *rating* al usuario creando anteriormente.