Recuperación de Información y Recomendaciones en la Web

Clasificador de películas por emoción

Grupo 5

Juan Pablo Borja	5.017.194-4
Zuo Heng Dai	5.283.369-5
Federico Dallo	5.086.465-2
Francisco González	4.944.851-4

Abordaje del problema

El problema que se quiso abordar durante la realización de este proyecto fue el de la clasificación y búsqueda de películas en base a sentimientos o emociones relacionadas a la misma. Esta clasificación se conseguiría a través de un software que tomaría como entrada el texto de descripción de la película (su sinopsis), y le asociará a la misma un conjunto de emociones las cuales serán guardadas dentro de una base de datos, para luego poder realizar una búsqueda a través de ellas, que sería utilizado al momento de realizar el filtrado de las películas.

Los datos de las películas fueron obtenidos a través de The Movie DB API [7] que retorna en estructura json o xml los datos de las películas que utilizaremos para luego realizar el análisis de texto. Estas se presentan de la siguiente manera:


```
{
            "adult": false,
            "backdrop_path": "/v6sqKSZQcyN5R7xzVmNVi3t2Kyf.jpg",
            "genre_ids": [
                99
            ],
            "id": 315946,
            "original_language": "fr",
            "original_title": "Passage de Venus",
            "overview": "Photo sequence of the rare transit of Venus over the face of
the Sun, one of the first chronophotographic sequences.  In 1873, P.J.C. Janssen, or
Pierre Jules César Janssen, invented the Photographic Revolver, which captured a series
of images in a row. The device, automatic, produced images in a row without human
intervention, being used to serve as photographic evidence of the passage of Venus
before the Sun, in 1874.",
            "popularity": 4.896,
            "poster_path": "/XWPDZzK7N2WQcejI8W96IxZEeP.jpg",
            "release_date": "1874-12-09",
            "title": "Passage of Venus",
            "video": false,
            "vote_average": 6.2,
            "vote_count": 71
}
```


Para la obtención de los datos decidimos acotar a solo las películas estrenadas entre 2010 hasta la actualidad, y los sentimientos que se intentaría asociar a las películas serían en principio happiness, sadness, excitement, fear, love y other. Esto puede verse ejemplificado en el archivo “API_Peliculas.py”.

En primer lugar se pensaba desarrollar una aplicación completa en Ruby on Rails, mientras que el analizador de texto para clasificar las películas sería desarrollado en Python. La idea sería primero analizar el corpus de películas y clasificarlo con la aplicación de Python para luego cargar dichos datos a una base de datos que estaría conectada a la aplicación principal.

Clasificador de películas (Python)

En primer lugar se creó un script para obtener los datos desde la API y poder guardar dichos datos en un csv. Este script se utilizó durante todo el desarrollo del proyecto para obtener distintos conjuntos de datos filtrados por distintos años, géneros, etc.
La API de la cual obtuvimos los datos no cuenta con muchas opciones para filtrar los datos a obtener, por ejemplo, los datos vienen paginados siempre en páginas de tamaño 20 fijo y no existe la posibilidad de solicitar un tamaño de página diferente. Debido a esto, en el script lo que hicimos fue iterar según la cantidad de películas que quisiéramos obtener y realizamos la consulta a la API en cada iteración incrementando el número de página. En principio se tomaron 1000 películas.
Dado que la API nos brinda los datos en formato .json, lo que hicimos fue primero guardarlo de esa manera y luego cargar estos datos a un .csv utilizando la librería “pandas”.

Estos datos luego los separamos en 2 corpus, uno de test y otro de entrenamiento, para poder utilizarlos luego para entrenar al clasificador. Se tomaron 800 películas para el corpus de test y 200 para el de training. Además de esto, se obtuvo un set aparte de 100 películas para utilizar como test de verificación. Los corpus de training y verificación fueron clasificados manualmente siguiendo las emociones estipuladas anteriormente. Para esta primera clasificación se tomaron en cuenta por un lado los géneros de cada película y por otro lado su descripción.

Luego de tener los datos prontos se hizo uso de la librería “sklearn” de Python para hacer uso de distintos tipos de clasificadores. En particular utilizamos Naive Bayes, Perceptron, Decision tree y Random forest. Estos clasificadores fueron utilizados representando los datos con Count Vectorizer y TfidfVectorizer (Term frequency – Inverse document frequency) de sklearn. Count Vectorizer hace uso de Bag of Words para representar los datos mediante una matriz en donde las filas representan los documentos a clasificar y las columnas las palabras utilizadas en el vocabulario, en donde se pone un 1 en esa fila y columna si el documento correspondiente a esa fila tiene la palabra correspondiente a esa columna y un 0 en otro caso, mientras que TfidfVectorizer de sklearn es equivalente a CountVectorizer pero utilizando TfidfTransformer para convertir una colección de documentos en una matriz de valores TF-IDF. El valor TF-IDF es una medida numérica utilizada para representar que tan relevante es una palabra para un documento (en este caso tweet) de una colección. Este valor aumenta según la cantidad de veces que una palabra aparece en un documento, pero a su vez es compensada con la frecuencia de aparición de la palabra en toda la colección. Esto permite identificar las palabras que aparecen más comúnmente en los documentos y cuáles no tanto. En cuanto a lo técnico, el valor de TF-IDF se calcula multiplicando la frecuencia de un término y la frecuencia inversa de documento (medida que indica si el término es común o no en la colección de documentos).

Los clasificadores mencionados anteriormente se utilizaron de 2 formas, una clasificando según el texto de la descripción de las películas y otra clasificando según la lista de géneros de la misma. Para todas las clasificaciones utilizamos también un corpus de Stopwords del idioma inglés (ya que los datos de las películas estaban en inglés) para que dichas palabras sean ignoradas a la hora de clasificar debido a que no son importantes para los resultados que se quiere obtener y permite evitar que el clasificador asocie documentos debido a que contienen muchas coincidencias de stopwords. También se probaron los clasificadores con

distintas configuraciones de max_df y min_df, siendo max_df un parámetro que determina el máximo número de veces que puede aparecer una palabra para que no sea ignorada, mientras que min_df determina lo opuesto, el mínimo número de veces que debe aparecer una palabra para que no sea ignorada. En particular se probaron valores entre 1 y 3 para ambos parámetros.

Presentación de Resultados

A continuación se presenta una tabla con los resultados obtenidos para los distintos clasificadores, con sus diferentes configuraciones y separados por los distintos vectorizers utilizados.

Bag of Words	max_df	min_df	F-Score	Bag of Words	max_df	min_df	F-Score
Naive Bayes por texto	1	/	0.19	Naive Bayes por texto	/	1	0.3
Naive Bayes por género	1	/	0.39	Naive Bayes por género	/	1	0.39
Perceptron por texto	1	/	0.16	Perceptron por texto	/	1	0.2
Perceptron por género	1	/	0.3	Perceptron por género	/	1	0.3
Decision Tree por texto	1	/	0.14	Decision Tree por texto	/	1	0.19
Decision Tree por género	1	/	0.45	Decision Tree por género	/	1	0.45
Random Forest por texto	1	/	0.08	Random Forest por texto	/	1	0.1
Random Forest por género	1	/	0.44	Random Forest por género	/	1	0.44
Naive Bayes por texto	2	/	0.21	Naive Bayes por texto	/	2	0.26
Naive Bayes por género	2	/	0.39	Naive Bayes por género	/	2	0.39
Perceptron por texto	2	/	0.15	Perceptron por texto	/	2	0.2
Perceptron por género	2	/	0.3	Perceptron por género	/	2	0.3
Decision Tree por texto	2	/	0.15	Decision Tree por texto	/	2	0.17
Decision Tree por género	2	/	0.44	Decision Tree por género	/	2	0.44
Random Forest por texto	2	/	0.08	Random Forest por texto	/	2	0.14
Random Forest por género	2	/	0.45	Random Forest por género	/	2	0.44
Naive Bayes por texto	3	/	0.22	Naive Bayes por texto	/	3	0.24
Naive Bayes por género	3	/	0.39	Naive Bayes por género	/	3	0.39
Perceptron por texto	3	/	0.19	Perceptron por texto	/	3	0.24
Perceptron por género	3	/	0.3	Perceptron por género	/	3	0.3
Decision Tree por texto	3	/	0.13	Decision Tree por texto	/	3	0.19
Decision Tree por género	3	/	0.45	Decision Tree por género	/	3	0.44

Random Forest por texto	3	/	0.1	Random Forest por texto	/	3	0.2
Random Forest por género	3	/	0.44	Random Forest por género	/	3	0.44

TF-IDF	max_df	min_df	F-Score	TF-IDF	max_df	min_df	F-Score
Naive Bayes por texto	1	/	0.19	Naive Bayes por texto	/	1	0.3
Naive Bayes por género	1	/	0.39	Naive Bayes por género	/	1	0.39
Perceptron por texto	1	/	0.16	Perceptron por texto	/	1	0.2
Perceptron por género	1	/	0.3	Perceptron por género	/	1	0.3
Decision Tree por texto	1	/	0.11	Decision Tree por texto	/	1	0.2
Decision Tree por género	1	/	0.44	Decision Tree por género	/	1	0.45
Random Forest por texto	1	/	0.08	Random Forest por texto	/	1	0.11
Random Forest por género	1	/	0.46	Random Forest por género	/	1	0.46
Naive Bayes por texto	2	/	0.21	Naive Bayes por texto	/	2	0.26
Naive Bayes por género	2	/	0.39	Naive Bayes por género	/	2	0.39
Perceptron por texto	2	/	0.15	Perceptron por texto	/	2	0.2
Perceptron por género	2	/	0.3	Perceptron por género	/	2	0.3
Decision Tree por texto	2	/	0.13	Decision Tree por texto	/	2	0.17
Decision Tree por género	2	/	0.45	Decision Tree por género	/	2	0.44
Random Forest por texto	2	/	0.08	Random Forest por texto	/	2	0.15
Random Forest por género	2	/	0.46	Random Forest por género	/	2	0.42
Naive Bayes por texto	3	/	0.22	Naive Bayes por texto	/	3	0.24
Naive Bayes por género	3	/	0.39	Naive Bayes por género	/	3	0.39
Perceptron por texto	3	/	0.19	Perceptron por texto	/	3	0.24
Perceptron por género	3	/	0.3	Perceptron por género	/	3	0.3
Decision Tree por texto	3	/	0.16	Decision Tree por texto	/	3	0.23
Decision Tree por género	3	/	0.45	Decision Tree por género	/	3	0.45
Random Forest por texto	3	/	0.08	Random Forest por texto	/	3	0.18
Random Forest por género	3	/	0.44	Random Forest por género	/	3	0.43

Podemos observar que para el caso de los clasificadores de texto, los resultados no son muy buenos, rondando valores entre 0,15 y 0,20 de F-score utilizando Bag of Words, y

obteniendo como mejor resultado un F-score de 0,30 utilizando un max_df sin definir y min_df= 1 utilizando el clasificador Naive Bayes. Por otro lado utilizando el vectorizador TF-iDF los resultados no se vieron impactados significativamente, rondando los mismos valores y obteniendo el mejor resultado para este vectorizador con las mismas configuraciones que para el mejor resultado utilizando Bag of Words, y con el mismo valor de F-score = 0,30 obtenido con el clasificador Naive Bayes.

Consideramos que estos resultados no son buenos, ya que estos resultados significan que incluso utilizando el clasificador que dio los mejores resultados, Naive Bayes, las películas presentadas a los usuarios serían en su mayoría incorrectamente asociadas a la emoción por la cual filtraron, y sería mucho más fácil que los usuarios filtraran puramente por género de películas ya que estos, según los resultados presentados por los clasificadores por género, son un mayor indicador de la emoción que le causará al espectador dicha película.

Problemas encontrados

En la primera iteración no conseguimos los resultados deseados, nos encontramos con un clasificador que clasificaba la gran mayoría de las películas con la emoción excitement. Tras un análisis de la situación pudimos sacar varias conclusiones.
Primero que nada, notamos que la amplia mayoría de las películas del corpus eran de acción, por lo que ya para comenzar, aunque hubiese estado bien implementado el clasificador, igual nos encontraríamos con una aplastante mayoría de películas clasificadas como excitement, lo cual se vuelve un problema para los clasificadores que hacen uso de los vectorizadores, y forman relaciones por ocurrencias de palabras que coinciden entre documentos.
Aún ignorando el último punto, notamos también que las descripciones de las películas eran bastantes parecidas entre sí, usaban terminología similar y apelaban a los mismos sentimientos (generalmente la emoción, cosa que también ayudaba a que el resultado terminase siendo mayoritariamente excitement). Esto se puede argumentar si se quiere a que debido a que las descripciones o tramas de las películas las crean los mismos autores de las películas, y suponiendo que los mismos intentan vender lo más posible su película con, entre otras cosas, la trama de la misma, estos van a intentar apelar a la emoción del lector e intentar atraerlos hacia el cine, para que consuman la película. Con esto en mente se puede traer a colación una posible convergencia tanto emocional como en terminología de las tramas, dificultando el trabajo no solo del clasificador, sino la de un ser humano que intente descifrar la hipotética emoción principal de la película. Esto último fue también algo que nos pasó a la hora de hacer una clasificación manual para la validación y testing del clasificador: muchas películas tienen tramas que exageran la emoción de una película con el motivo de, creemos, generar un atractivo a la película que quizás ni siquiera tenga, pero así funciona el marketing también.
Finalmente, pensamos en que quizás pudimos haber hecho una mejor elección en cuanto a las emociones con las cuales clasificar las películas. Algo que notamos a la hora de hacer la clasificación manual fue que había muchas películas dramáticas que se encontraban en el límite dentro de happiness y sadness, porque aunque sean emociones en teoría opuestas, notamos que había una gran cantidad de películas que tocaban ambas emociones repetidas veces. Para solucionar este problema, decidimos realizar un re etiquetado de las películas como se menciona a continuación.

Re-etiquetado

Debido a todos los problemas encontrados previamente, decidimos obtener nuevos corpus de datos y utilizar una distintas listas de sentimientos para la clasificación, ya que nos dimos cuenta que la lista elegida al principio no se adaptó correctamente debido a que, como se mencionó anteriormente, muchas de las películas cayeron en la categoría excitement.

Se optó por probar con 2 nuevas combinaciones de sentimientos distintos para intentar adaptar la calcificación a la realidad con la expectativa de que estas nuevas clasificaciones resultaran en mejores resultados de los clasificadores. Las clasificaciones propuestas fueron las siguientes:

Clasificación V2 5E:

- Happiness
- Emotional (Abarca desde Dramas, hasta Sadness)
- Love
- Fear
- Excitement

Clasificación V2 Automated:

- Love
- Emotional (Abarca Happiness y Sadness)
- Fear
- Excitement

Para entrenar los clasificadores en estos casos se utilizó un enfoque diferente. Intentamos obtener un corpus de training que tuviera más o menos la misma cantidad de películas para cada emoción, logrando así que el clasificador no se viera más influenciado por una emoción en particular como sucedía anteriormente con “Excitement”. Para conseguir esto se obtuvieron nuevos datos de la API filtrados por géneros más específicos que a nuestro parecer se corresponden mejor con las emociones elegidas. En particular se tomaron películas de terror, acción, romance, drama y familiares. Luego se tomaron dos enfoques distintos para cada uno de estos clasificadores.

Generación del corpus de entrenamiento Clasificador V2 5E

Para el Clasificador V2 5E, que cuenta con 5 emociones, para cada uno de dichos géneros se clasificaron las películas manualmente, tomando solo en cuenta el texto de la descripción de la película para ver si tenía palabras que se correspondieran con las emociones planteadas. Según esto se etiquetó la película con la emoción indicada o se eliminó la película del corpus en caso de que no se correspondiera con dicha emoción. (Por ejemplo si en el corpus de terror una película no corresponde con la emoción “miedo”, esta película era eliminada). Luego se fusionaron todos los corpus de cada emoción por separado, generando un corpus de 500 películas etiquetadas cada una con dicha emoción.

Generación del corpus de verificación Clasificador V2 5E

Para la generación de este corpus, se tomó el corpus de verificación para el clasificador V1 (el primero en utilizarse) en donde se re etiquetaron las películas con emociones sadness, a emotional y las películas con clasificación other fueron etiquetadas con la emoción más cercana dentro de las 5 posibles.

Generación del corpus de entrenamiento Clasificador V2 Automated

Para el Clasificador V2 Automated, que cuenta con 4 emociones, para cada uno de dichos géneros se clasificaron las películas automáticamente, tomando solo en cuenta el género de la película para obtener un corpus de mayor tamaño. Esto se hizo realizando una asociación de género-emoción tal que para las películas de Romance las asociamos a la emoción love, las películas de género Terror a la emoción fear, las películas de género Acción a la emoción excitement, y las películas de género Dramático a la emoción emotional. Luego se fusionaron todos los corpus de cada emoción por separado, generando un corpus de 1600 películas etiquetadas cada una con dicha emoción.

Generación del corpus de verificación Clasificador V2 Automated

Para la generación de este corpus, se tomó el mismo corpus de verificación para el clasificador V1 (el primero en utilizarse) en donde se re etiquetaron las películas con emociones sadness y happiness, a emotional y las películas con clasificación other fueron etiquetadas con la emoción más cercana dentro de las 4 posibles.

Presentación de Resultados luego del Re-etiquetado

Resultados Clasificación V2 E5

A continuación se presenta una tabla con los resultados obtenidos para los distintos clasificadores, con sus diferentes configuraciones y separados por los distintos vectorizers utilizados, para la clasificación V2 E5.

Bag of Words	max_df	min_df	F-Score	Bag of Words	max_df	min_df	F-Score
Naive Bayes por texto	1	/	0.23	Naive Bayes por texto	/	1	0.46
Naive Bayes por género	1	/	0.56	Naive Bayes por género	/	1	0.56
Perceptron por texto	1	/	0.26	Perceptron por texto	/	1	0.37
Perceptron por género	1	/	0.5	Perceptron por género	/	1	0.5
Decision Tree por texto	1	/	0.21	Decision Tree por texto	/	1	0.34
Decision Tree por género	1	/	0.58	Decision Tree por género	/	1	0.58

Random Forest por texto	1	/	0.25	Random Forest por texto	/	1	0.38
Random Forest por género	1	/	0.56	Random Forest por género	/	1	0.57
Naive Bayes por texto	2	/	0.28	Naive Bayes por texto	/	2	0.49
Naive Bayes por género	2	/	0.56	Naive Bayes por género	/	2	0.56
Perceptron por texto	2	/	0.26	Perceptron por texto	/	2	0.35
Perceptron por género	2	/	0.5	Perceptron por género	/	2	0.5
Decision Tree por texto	2	/	0.24	Decision Tree por texto	/	2	0.35
Decision Tree por género	2	/	0.58	Decision Tree por género	/	2	0.58
Random Forest por texto	2	/	0.18	Random Forest por texto	/	2	0.38
Random Forest por género	2	/	0.56	Random Forest por género	/	2	0.57
Naive Bayes por texto	3	/	0.29	Naive Bayes por texto	/	3	0.43
Naive Bayes por género	3	/	0.56	Naive Bayes por género	/	3	0.56
Perceptron por texto	3	/	0.29	Perceptron por texto	/	3	0.4
Perceptron por género	3	/	0.5	Perceptron por género	/	3	0.5
Decision Tree por texto	3	/	0.23	Decision Tree por texto	/	3	0.36
Decision Tree por género	3	/	0.58	Decision Tree por género	/	3	0.58
Random Forest por texto	3	/	0.18	Random Forest por texto	/	3	0.39
Random Forest por género	3	/	0.57	Random Forest por género	/	3	0.57

TF-IDF	max_df	min_df	F-Score	TF-IDF	max_df	min_df	F-Score
Naive Bayes por texto	1	/	0.23	Naive Bayes por texto	/	1	0.46
Naive Bayes por género	1	/	0.56	Naive Bayes por género	/	1	0.56
Perceptron por texto	1	/	0.26	Perceptron por texto	/	1	0.37
Perceptron por género	1	/	0.5	Perceptron por género	/	1	0.5
Decision Tree por texto	1	/	0.22	Decision Tree por texto	/	1	0.35
Decision Tree por género	1	/	0.58	Decision Tree por género	/	1	0.58
Random Forest por texto	1	/	0.3	Random Forest por texto	/	1	0.35
Random Forest por género	1	/	0.57	Random Forest por género	/	1	0.57
Naive Bayes por texto	2	/	0.28	Naive Bayes por texto	/	2	0.49

Naive Bayes por género	2	/	0.56	Naive Bayes por género	/	2	0.56
Perceptron por texto	2	/	0.26	Perceptron por texto	/	2	0.35
Perceptron por género	2	/	0.5	Perceptron por género	/	2	0.5
Decision Tree por texto	2	/	0.22	Decision Tree por texto	/	2	0.33
Decision Tree por género	2	/	0.58	Decision Tree por género	/	2	0.58
Random Forest por texto	2	/	0.19	Random Forest por texto	/	2	0.39
Random Forest por género	2	/	0.57	Random Forest por género	/	2	0.57
Naive Bayes por texto	3	/	0.29	Naive Bayes por texto	/	3	0.43
Naive Bayes por género	3	/	0.56	Naive Bayes por género	/	3	0.56
Perceptron por texto	3	/	0.29	Perceptron por texto	/	3	0.4
Perceptron por género	3	/	0.5	Perceptron por género	/	3	0.5
Decision Tree por texto	3	/	0.22	Decision Tree por texto	/	3	0.38
Decision Tree por género	3	/	0.58	Decision Tree por género	/	3	0.58
Random Forest por texto	3	/	0.21	Random Forest por texto	/	3	0.38
Random Forest por género	3	/	0.57	Random Forest por género	/	3	0.57

Podemos observar que para el caso de los clasificadores de texto, los resultados mejoraron considerablemente, rondando el valor 0,23 de F-score utilizando Bag of Words para las distintas configuraciones de max_df y con min_df sin definir. Los resultados para las distinta configuraciones de min_df son en general mejores, rondando el valor 0,40 de F-score utilizando Bag of Words y obteniendo como mejor resultado un F-score de 0,49 utilizando un max_df sin definir y min_df= 2 obtenido por el clasificador Naive Bayes .

Por otro lado utilizando de TF-iDF los resultados no se vieron impactados significativamente de igual manera que en los resultados del clasificador V1, rondando los mismos valores y obteniendo el mejor resultado para este vectorizador con las mismas configuraciones que para el mejor resultado utilizando Bag of Words, y con el mismo valor de F-score = 0,49 obtenido con el clasificador Naive Bayes.

Consideramos que estos resultados son significativamente mejores que los anteriores y más específicamente el clasificador Naive Bayes con cualquiera de los vectorizadores en los cuales sus configuraciones sean max_df sin definir y min_df=2 con stopwords, que resultó en el mejor resultado obtenido, acercándose a los valores obtenidos por los clasificadores de género, los cuáles siempre se mantuvieron por encima de los clasificadores de texto, dándonos a entender que es posible no solo alcanzar pero posiblemente sobrepasar estos valores, brindando importancia a esta investigación.

Resultados Clasificación V2 Automated

A continuación se presenta una tabla con los resultados obtenidos para los distintos clasificadores, con sus diferentes configuraciones y separados por los distintos vectorizers utilizados, para la clasificación V2 Automated.

Bag of Words	max_df	min_df	F-Score	Bag of Words	max_df	min_df	F-Score
Naive Bayes por texto	1	/	0.38	Naive Bayes por texto	/	1	0.37
Naive Bayes por género	1	/	0.42	Naive Bayes por género	/	1	0.42
Perceptron por texto	1	/	0.36	Perceptron por texto	/	1	0.35
Perceptron por género	1	/	0.39	Perceptron por género	/	1	0.39
Decision Tree por texto	1	/	0.37	Decision Tree por texto	/	1	0.34
Decision Tree por género	1	/	0.54	Decision Tree por género	/	1	0.54
Random Forest por texto	1	/	0.38	Random Forest por texto	/	1	0.32
Random Forest por género	1	/	0.29	Random Forest por género	/	1	0.34
Naive Bayes por texto	2	/	0.37	Naive Bayes por texto	/	2	0.37
Naive Bayes por género	2	/	0.42	Naive Bayes por género	/	2	0.42
Perceptron por texto	2	/	0.37	Perceptron por texto	/	2	0.38
Perceptron por género	2	/	0.39	Perceptron por género	/	2	0.39
Decision Tree por texto	2	/	0.38	Decision Tree por texto	/	2	0.34
Decision Tree por género	2	/	0.54	Decision Tree por género	/	2	0.54
Random Forest por texto	2	/	0.4	Random Forest por texto	/	2	0.35
Random Forest por género	2	/	0.34	Random Forest por género	/	2	0.35
Naive Bayes por texto	3	/	0.32	Naive Bayes por texto	/	3	0.42
Naive Bayes por género	3	/	0.42	Naive Bayes por género	/	3	0.42
Perceptron por texto	3	/	0.36	Perceptron por texto	/	3	0.34
Perceptron por género	3	/	0.39	Perceptron por género	/	3	0.39
Decision Tree por texto	3	/	0.34	Decision Tree por texto	/	3	0.37
Decision Tree por género	3	/	0.54	Decision Tree por género	/	3	0.54

Random Forest por texto	3	/	0.28	Random Forest por texto	/	3	0.37
Random Forest por género	3	/	0.37	Random Forest por género	/	3	0.28

TF-IDF	max_df	min_df	F-Score	TF-IDF	max_df	min_df	F-Score
Naive Bayes por texto	1	/	0.38	Naive Bayes por texto	/	1	0.37
Naive Bayes por género	1	/	0.42	Naive Bayes por género	/	1	0.42
Perceptron por texto	1	/	0.36	Perceptron por texto	/	1	0.35
Perceptron por género	1	/	0.39	Perceptron por género	/	1	0.39
Decision Tree por texto	1	/	0.36	Decision Tree por texto	/	1	0.34
Decision Tree por género	1	/	0.54	Decision Tree por género	/	1	0.54
Random Forest por texto	1	/	0.35	Random Forest por texto	/	1	0.42
Random Forest por género	1	/	0.31	Random Forest por género	/	1	0.28
Naive Bayes por texto	2	/	0.37	Naive Bayes por texto	/	2	0.37
Naive Bayes por género	2	/	0.42	Naive Bayes por género	/	2	0.42
Perceptron por texto	2	/	0.37	Perceptron por texto	/	2	0.38
Perceptron por género	2	/	0.39	Perceptron por género	/	2	0.39
Decision Tree por texto	2	/	0.31	Decision Tree por texto	/	2	0.34
Decision Tree por género	2	/	0.54	Decision Tree por género	/	2	0.54
Random Forest por texto	2	/	0.38	Random Forest por texto	/	2	0.39
Random Forest por género	2	/	0.3	Random Forest por género	/	2	0.36
Naive Bayes por texto	3	/	0.32	Naive Bayes por texto	/	3	0.42
Naive Bayes por género	3	/	0.42	Naive Bayes por género	/	3	0.42
Perceptron por texto	3	/	0.36	Perceptron por texto	/	3	0.34
Perceptron por género	3	/	0.39	Perceptron por género	/	3	0.39
Decision Tree por texto	3	/	0.39	Decision Tree por texto	/	3	0.35
Decision Tree por género	3	/	0.54	Decision Tree por género	/	3	0.54
Random Forest por texto	3	/	0.38	Random Forest por texto	/	3	0.34
Random Forest por género	3	/	0.34	Random Forest por género	/	3	0.31

Podemos observar que para el caso de los clasificadores de texto, los resultados mejoraron considerablemente, rondando el valor 0,34 de F-score utilizando Bag of Words para las distintas configuraciones de max_df y con min_df sin definir. Los resultados para las distinta configuraciones de min_df son en general mejores, rondando el valor 0,36 de F-score utilizando Bag of Words y obteniendo como mejor resultado un F-score de 0,42 utilizando un max_df sin definir y min_df=3 obtenido por el clasificador Naive Bayes . Estos resultados fueron en promedio mejores que los del clasificador V2 E5 pero peores que los mejores clasificadores con sus mejores configuraciones.

Por otro lado utilizando de TF-iDF los resultados no se vieron impactados significativamente de igual manera que en los resultados del clasificador V1 y V2 E5, rondando los mismos valores y obteniendo el mejor resultado para este vectorizador con las mismas configuraciones que para el mejor resultado utilizando Bag of Words, y con el mismo valor de F-score = 0,42 obtenido con el clasificador Naive Bayes.

La mayor diferencia entre ambos clasificadores V2 es que el V2 E5 requirió de muchísimo mayor esfuerzo manual a la hora de clasificar mientras el V2 Automated tomó como input las asociaciones de emociones por género lo que permitió que su conjunto de entrenamiento fuera mucho mayor, aunque debido a esto el filtrado de películas que no pertenecen a dicha emoción no es completamente correcto. Aún así en promedio los resultados de V2 Automated son mejores, pero a la hora de tomar un clasificador para utilizar para una aplicación se seleccionaría el mejor clasificador obtenido por la clasificación V2 E5.

Por último observamos que las clasificaciones por género fueron mejores para los clasificadores V2 E5 y se cree que esto se debe a que al filtrar manualmente las películas a la hora de etiquetar el corpus de entrenamiento se filtraron aquellas que pertenecían fuertemente a múltiples categorías, y estas inevitablemente pertenecían a múltiples géneros, mientras que las que sí fueron clasificadas se centraban mayormente en menos géneros, obteniendo una asociación genero-emoción más fuerte.

Problemas encontrados luego del Re-etiquetado

En primer lugar, creemos que el entrenamiento podría haber sido mejorado si los corpus de entrenamiento y verificación hubieran sido más grandes. Esto no se hizo debido a que llevaba demasiado tiempo clasificar las películas manualmente.
Por otro lado, un problema que encontramos fue la decisión de haber eliminado la categoría other, ya que debido a esto hubo películas que no encajaban directamente en ninguna de las clasificaciones pero tuvimos que asignarle una de todas formas. Si hubiésemos dejado la clasificación “other” la precisión de los clasificadores hubiera aumentado, aunque el recall hubiese bajado ya que las películas que no fueran fuertemente asociadas a una emoción serían etiquetadas como other, emoción que no sería mostrada para ser filtrada en la aplicación. Igualmente esto es mejor para nuestro caso de estudio, ya que priorizamos una buena precisión antes que un buen recall, por lo que quedaría como trabajo a futuro.

Otro problema que encontramos fue que obtuvimos películas con descripciones muy cortas que no aportaban nada y no se les podía asignar una categoría tan fácilmente. También obtuvimos películas del lado opuesto, con descripciones demasiado largas que llegaban a tener palabras que podían hacer que la película cayera en más de una clasificación. Creemos que si hubiéramos descartado películas de este tipo se podrían haber obtenido mejores resultados.
Otra cosa que creemos que hubiese servido para obtener mejores resultados sería haber usado valores porcentuales para max_df y min_df con respecto al corpus, en lugar de usar números fijos como hicimos. Los números fijos no funcionan mal debido a que no teníamos un corpus muy grande, pero aun así se podría haber mejorado un poco los resultados de la otra forma. Con un corpus grande se tendrían que usar valores porcentuales obligatoriamente para poder obtener buenos resultados.
Por último, creemos que otro problema pudo haber sido usar un vector de palabras genérico para realizar word embedding, en lugar de haber creado uno propio específicamente para esto, con palabras que estuvieran más relacionadas a descripciones de películas.

Acercamiento Word Embedding

Se propuso otra forma de evaluar las descripciones de las películas tomando un vector de palabras en inglés[8]. La idea es calcular los vectores centroides de las películas para entrenar según las palabras de sus descripciones y también calcular los vectores centroides de las películas para verificar los resultados. Luego utilizamos el clasificador KNN (K Near Neighbors) para determinar la clasificación de cada película según los vectores centroides de sus 3 vecinos más cercanos, tomando el resultado más frecuente entre sus vecinos (por ejemplo, si 2 de sus 3 vecinos fueron clasificados con fear, la película será clasificada con fear también).

Utilizando el corpus original y los dos corpus nuevos luego del re-etiquetado, se obtuvieron los siguientes resultados:

Word Embedding(with KNN)	K	F-Score
Versión Original	3	0.224
Versión 5E	3	0.397
Versión Auto	3	0.340

Se puede observar que con el entrenamiento con el corpus original se obtuvo el peor resultado, el cual fue un F-Score de 0.224. Era esperable que este resultado fuera bastante malo debido a todas las pruebas que se habían realizado con anterioridad utilizando el mismo corpus. Para los nuevos corpus se obtuvieron resultados un poco mejores, pero aun así consideramos que no son suficientemente buenos.

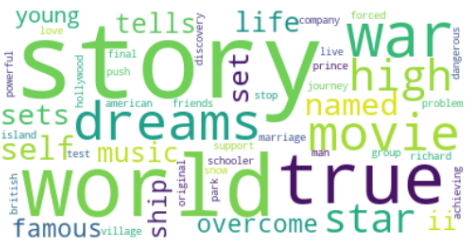
Acercamiento Clustering K-Means

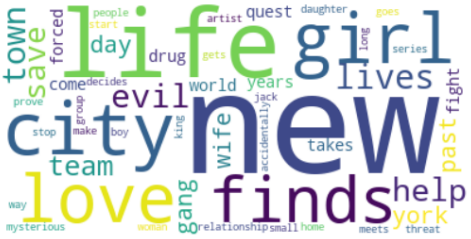
Como todos los clasificadores utilizados anteriormente utilizaban aprendizaje supervisado decidimos realizar pruebas con un clasificador no supervisado. Para esto se decidió utilizar K-Means siguiendo un tutorial visto en [9]. El clasificador K-Means consta de separar el conjunto de películas a verificar en k clusters agrupados según la relevancia de las palabras utilizando el vectorizador TF-IDF.

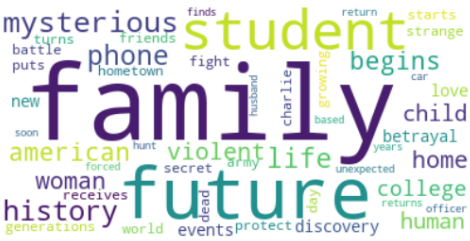
Para nuestras pruebas decidimos que k fuera la cantidad de clasificaciones de películas, es decir la cantidad de emociones elegidas para cada corpus. Para el corpus original k es 6, 5 para el cluster V2 E5 y 4 para el V2 automated.

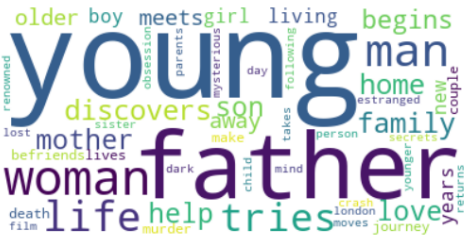
También, luego de realizar el clustering es posible visualizar en una imagen las palabras más representativas de cada cluster, la cual es generada como parte del output del algoritmo, guardando en formato ".png".

Resultados para el corpus original



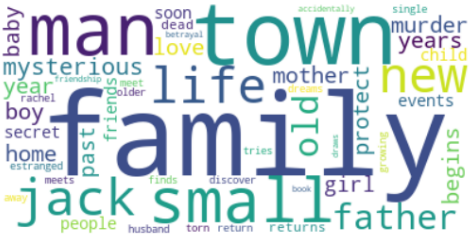


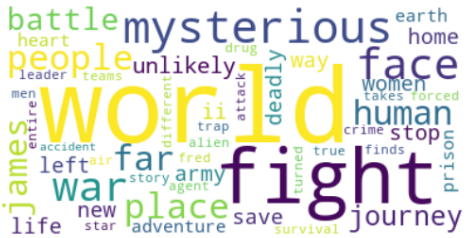


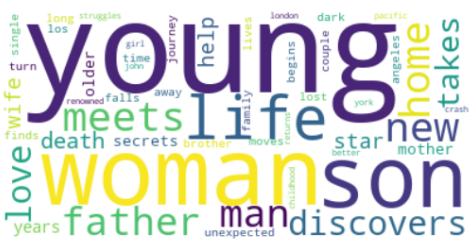


Podemos ver que los clusters obtenidos para este corpus no son fáciles de diferenciar entre las distintas emociones debido a que tienen muchas palabras sin una relación clara. Por ejemplo, en la cuarta foto podemos ver palabras como ‘Love’ y ‘Girl’ que podrían asociarse al sentimiento love, pero luego podemos ver que también tiene palabras como ‘Gang’, ‘Drug’ y ‘Evil’ que no son muy comunes en películas puramente románticas.

Resultados para el corpus V2 E5

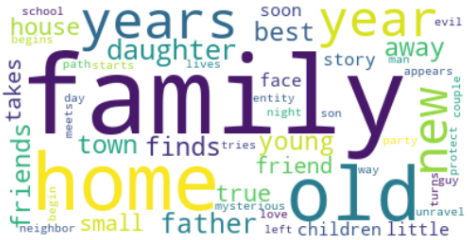


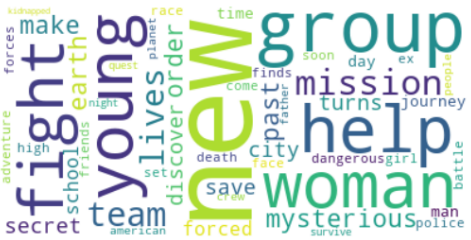


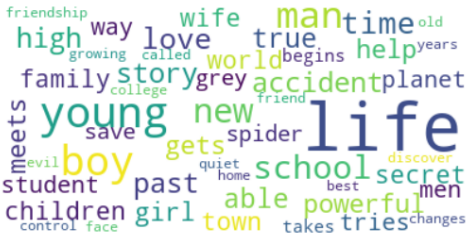


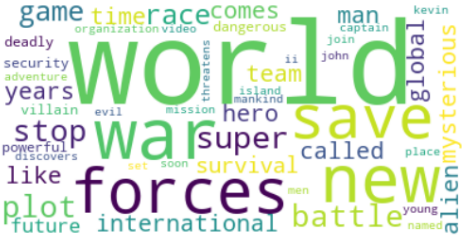
Para este corpus podemos ver que algunos clusters se pueden comenzar a asociar con las distintas emociones, pero la mayoría siguen teniendo el mismo problema visto para el corpus original. En particular podemos inferir que la última imagen representa el cluster para la emoción ‘love’, ya que tiene palabras como ‘Love’, ‘Girl’, ‘Boy’, ‘School’, ‘Falls’, etc. Sin embargo, este es el único cluster al cual se le puede inferir una emoción.

Resultados para el corpus V2 automated









Por último, para este corpus pudimos inferir que la primera imagen hace referencia a la emoción ‘happines’, la tercera a ‘love’ y las 2 restantes podrían hacer referencia a excitement y fear. Estas 2 son difíciles de diferenciar con estos clusters debido a que tienen palabras que pueden llegar a aparecer en los mismos contextos.

Trabajo futuro

A partir del trabajo hecho se puede iterar con algunas mejoras. Primero que nada, se puede seguir intentando mejorar el clasificador. Para esto se podría entre otras cosas elegir un conjunto distinto de stopwords o conseguir y clasificar manualmente un corpus más grande. Otro cambio con el que se podría probar es el de agregar distintos inputs, como lo son utilizar el género de la película en conjunto con la descripción de la misma, para poder mejorar así la precisión del clasificador. Por otro lado, se podría buscar reseñas para las

películas y utilizarlas para intentar analizar las emociones que la película le hizo sentir a la persona que escribió la reseña. Para esto habría que tener cuidado porque hay muchas reseñas que son bastante técnicas y formales y por lo tanto no reflejan de gran manera los sentimientos de la persona. Si se quisiera utilizar este enfoque habría que buscar reseñas y comentarios sobre las películas en lugares no muy formales, por lo que se descartan imdb y rotten tomatoes. En su lugar se podría buscar comentarios de personas en redes sociales o letterboxd, la cual es una web bastante popular para hacer reseñas de películas a la cual todo el mundo puede tener acceso y donde la gente generalmente deja reseñas más personales y no tan técnicas.
Adicionalmente se podría agregar a la clasificación, analizadores semánticos de texto.
Para finalizar, se podrían agregar los análisis de los posters de las películas para analizar a la decisión final del clasificador de emoción.
Todas estos analizadores podrían aplicarse de forma separada (permitiendo reutilizar las herramientas ya expuestas en este informe), en donde un clasificador final tome un vector de resultados de los clasificadores previos y posiblemente su porcentaje de confianza de dicha clasificación para cada clasificador, y decida a qué categoría pertenece finalmente esa película.

Conclusiones

Concluimos luego de obtener todos los F-score para cada clasificador probado, que el mejor acercamiento a este problema es el de clasificar manualmente un gran corpus de entrenamiento, como fue hecho para la clasificación V2 E5, forzando a que las descripciones de dichas películas estén altamente relacionadas con la emoción que se busca clasificar. En adición, mantener las emociones seleccionadas para la clasificación V2 E5, añadiendo la emoción “other” para que aumentar la importancia de la precisión a costo del recall, ya que la cantidad de películas existentes es muy grande. Seguido de esto se podría desbalancear el corpus de entrenamiento poco a poco agregando una mayor cantidad de películas asociadas a la emoción “other” para lograr llegar al punto deseado de recall y precisión de manera en que a la hora de mostrar películas por emoción en la aplicación a desarrollar, estas sean con alta probabilidad correctamente asociadas a la emoción por la cual se filtra.

De esta manera, es muy posible que se logre una precisión mayor a que simplemente buscar el género que con mayor probabilidad estará asociado a la emoción que se busca experimentar.

Creemos que el acercamiento de clasificar las películas por clustering no fue exitoso, y que no es necesario profundizar el mismo ya que las asociaciones o clusters que son generados por dicho acercamiento no tienen porqué estar relacionadas con las emociones que provocan las películas.

Por otro lado, creemos que el acercamiento de word embedding es prometedor, aún si su F-score presentado no fue el mejor dentro de todos los clasificadores explorados, debido a que este no fue entrenado específicamente para la asociación de descripciones de películas con sus respectivos sentimientos.

Bibliografía

[1] - https://scikit-learn.org/stable/modules/naive_bayes.html
[2] - https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html#sklearn.linear_model.Perceptron
[3] - https://scikit-learn.org/stable/modules/tree.html
[4] - https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees
[5] - https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html?highlight=vectorizer#sklearn.feature_extraction.text.CountVectorizer
[6] - https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html?highlight=vectorizer#sklearn.feature_extraction.text.TfidfVectorizer
[7] - https://www.themoviedb.org/documentation/api
[8] - https://fasttext.cc/docs/en/english-vectors.html
[9] - https://www.kaggle.com/code/aybukehamideak/clustering-text-documents-using-k-means
[10] - https://en.wikipedia.org/wiki/F-score
[11] - https://en.wikipedia.org/wiki/Word_embedding
[12] - https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
[13] - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
[14] - https://radimrehurek.com/gensim/models/keyedvectors.html
[15] - https://pythonspot.com/nltk-stop-words/