



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

**Detección de Temas de Discusión y
Difusión de Opiniones en Comunidades
Online**

Autora: Carmen Bermejo Hernández

Tutor: Oscar Corcho García

Madrid, Octubre 2020

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster

Máster Universitario en Inteligencia Artificial

Título: Detección de Temas de Discursión y Difusión de Opiniones en Comunidades Online

Octubre 2020

Autora: Carmen Bermejo Hernández

Tutor:

Oscar Corcho García
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

El objetivo de este trabajo es proveer de una herramienta para poder entender mejor la difusión de ideas, argumentos y opiniones en comunidades online.

Para ello he desarrollado una aplicación web que permite identificar los temas de debate dentro de una comunidad online y, dentro de esos debates, identificar los distintos argumentos y opiniones que los usuarios publican sobre un tema.

Este trabajo sienta las bases para, en un desarrollo futuro, poder estudiar como se propagan las ideas dentro de las comunidades online con el objetivo de entender mejor estas dinámicas.

La comunidad online elegida para este trabajo ha sido Menéame, una plataforma web tipo foro donde los usuarios comparten noticias de actualidad que otros usuarios comentan, generando hilos de debate. Los datos analizados corresponden al contenido publicado por los usuarios en esta plataforma durante el primer semestre de 2020, que cubre la duración del primer estado de alarma en España debido a la pandemia de COVID-19.

Por medio de la aplicación se han importado las noticias y comentarios generando una estructura de hilos que asocia cada noticia con sus comentarios. A continuación se han agrupado las noticias por medio del algoritmo de clústering AffinityPropagation de Sckit-learn.

Con una búsqueda por palabras clave dentro de los clusters se han identificado aquellos relacionados con una temática escogida como ejemplo: la renta básica. Se ha comprobado la eficacia del clasificador para identificar esta temática utilizando una muestra de 100 ejemplos (50 noticias de los clusters etiquetados como renta básica y 50 noticias al azar) anotada por tres personas con una puntuación de inter-annotator agreement de 0,91, obteniendo una exhaustividad del 100% y una precisión del 77%.

Para identificar los distintos argumentos e ideas expresadas dentro de los debates sobre renta básica en la plataforma se ha hecho un nuevo clústering a partir de todos los documentos (noticias y comentarios) pertenecientes a los hilos de esta temática. Queda como desarrollo futuro seguir trabajando en la clasificación de argumentos.

La plataforma cuenta con información sobre qué usuarios han publicado cada mensaje y en qué momento por lo que, una vez etiquetados los argumentos, se puede extrapolar mucha información sobre la difusión de ideas entre los usuarios y a lo largo del tiempo en la plataforma, así como la influencia que la exposición a distintas ideas tiene en los usuarios, teniendo también en cuenta el punto de partida de los mismos, según las ideas que expresan.

Este trabajo incluye también un pequeño estado del arte de los métodos, modelos y métricas del análisis de influencia, con el fin de sentar las bases para una investigación más amplia que permitirá identificar las mejores herramientas para poder entender las dinámicas de difusión de ideas y consolidación de opiniones dentro de una comunidad.

Abstract

The goal of this project is to provide a tool to improve the understanding of opinion dynamics and idea flows in online communities.

For this I've developed a web application that helps to identify topics of debate inside an online community and the different arguments and opinions published by users in the context of this debates.

This work sets the foundations that will allow, in future developments, to understand better how ideas and opinions spread within this communities.

The case of study chosen is Menéame, a Spanish platform, similar to Reddit and very popular in Spain and Latin America. In this platform, users share news that other users comment, creating discussion threads.

The data analyzed is the content published by users in the platform during the first semester of 2020, that includes the first state of alarm in Spain during the COVID-19 pandemic.

The news and comments has been imported to our web application, generating threads that link each news with its comments. Then the news has been clustered using the AffinityPropagation algorithm, implemented by Sckit-learn.

Using a key word search inside the clusters, I have identify the threads that fit the topic chosen to test the application: the basic income. The efficiency of the classifier has been tested using an anotated sample with 100 instances (50 chosen among the news inside the clusters tagged as Basic Income and 50 randomly chosen among all the news). This sample has been annotated by 3 anotators, obtaining an inter-annotator agreement of 0,91, a recall of 100% and a 77% precission.

To identify the arguments and ideas shared inside the debates about basic income in the platform, a new clustering has been made, using all the documents (news and comments) inside the threads tagged with this topic. It is left for future development to continue working along these lines to improve arguments classification.

The aplication has information about what users has published each message and when so, once it's been tagged, we can infer a lot of information about idea flow among the users and across time in the platform, along with the influence that idea exposition has in the users, taking also into account their starting point, acording to the ideas they share.

This work also includes a small state of the art of methods, models and metrics of influence analysis, intending to set the ground work for a broader investigation in the future, that will allow to identify the best tools to understand the dynamics of idea flows and how opinions take root inside online communities.

1 Tabla de contenidos

1	Introducción	6
2	Prefacio: Evolución del proyecto	1
2.1	Idea inicial	1
2.2	Planificación	1
2.3	Reajuste del alcance del proyecto	2
3	Desarrollo	3
3.1	Plan para la implementación	3
3.2	Estructura de la aplicación	3
3.3	Fuente de los datos	5
3.3.2.1	Noticias (links):	6
3.3.2.2	Comentarios:	7
3.3.3.1	Estructura de los datos dentro de la aplicación	7
3.3.3.2	Módulo de importación de datos: csv_import	10
3.4	Identificación de los temas	11
3.4.1.1	Clasificación supervisada y no supervisada	11
3.4.1.2	Clustering con k-means y Affinity Propagation	12
3.4.2.1	Versión con datasets de Scikit-learn	13
3.4.2.2	Versión sin datasets	15
3.4.3.1	Vectorización	17
3.4.3.2	Palabras comunes (stop words)	18
3.4.3.3	Ajuste de pesos Tf-idf	19
3.4.4.1	Generación de clusters con Affinity Propagation	19
3.4.4.2	Clusters en dos niveles (árboles)	20
3.4.7.1	Inter-Annotator Agreement	26
3.4.7.2	Precisión y exhaustividad	26
3.5	Identificación de argumentos	26
3.6	Comportamiento de los usuarios	28
3.7	Métodos y modelos de análisis de influencia	29
4	Resultados y conclusiones	34
5	Bibliografía	35
6	Anexo: Análisis de Influencia	42
6.1	Breve historia del análisis de influencia	42
6.1.2.1	Las celebridades y la teoría del flujo de dos pasos	43
6.1.2.2	Social Physics	43
6.2	Conceptos teóricos	44
6.2.1.1	Aprendizaje social (social learning)	44
6.2.1.2	Susceptibilidad	44
6.2.1.3	Confianza	45

6.2.1.4 Flujo de ideas (idea flow).....	46
6.2.2.1 Cambios en el comportamiento.....	46
6.2.2.2 Razonamiento individual y flujo de ideas	47
6.2.2.3 Dinámicas grupales	48
6.2.2.4 Sentido común	48
6.2.3.1 Diversidad	48
6.2.3.2 Cámaras de eco	49
6.2.3.3 Innovación.....	49
6.3 Referencias	50

1 Introducción

El objetivo de este trabajo es proveer de una herramienta para poder entender mejor la difusión de ideas, argumentos y opiniones en comunidades online.

Para ello he desarrollado una aplicación que permite identificar los temas de debate dentro de una comunidad online y, dentro de esos debates, identificar los distintos argumentos y opiniones que los usuarios publican sobre un tema.

Este trabajo sienta las bases para, en un desarrollo futuro, poder estudiar como se propagan las ideas dentro de las comunidades online con el objetivo de entender mejor estas dinámicas.

Fuente de los datos:

La comunidad online elegida para este trabajo ha sido Menéame[1], una plataforma web tipo foro donde los usuarios comparten noticias de actualidad que otros usuarios comentan, generando hilos de debate. Los datos analizados corresponden al contenido publicado por los usuarios en esta plataforma durante el primer semestre de 2020, que cubre la duración del primer estado de alarma en España debido a la pandemia de COVID-19.

En el apartado 3.3 *Fuente de los datos*, explico tanto la estructura y formato de los datos proporcionados por Menéame, como la forma de procesarlos e importarlos a nuestra aplicación.

Identificación de temas:

Con el fin de poder entender la difusión de ideas en esta plataforma, primero he agrupado las noticias compartidas por temas, utilizando el algoritmo de clústering AffinityPropagation[2]. Los detalles sobre la implementación de este clustering y la toma de decisiones que llevó a elegir este método se encuentran en la sección 3.4 *Identificación de los temas*.

A continuación he probado a identificar un tema concreto: la renta básica[3]. Me pareció un tema bastante interesante para este estudio ya que, además de estar de actualidad en ese momento debido a que el gobierno español anunció que iban a aprobar una medida en esa línea durante esos meses, hay muchas opiniones enfrentadas al respecto del mismo por lo que genera mucho debate. Esta búsqueda y los resultados se describen en el apartado 3.4.6 *Búsqueda de temáticas con los clusters*.

Para probar la fiabilidad del clasificador, generé una muestra con ejemplos de noticias de los clusters etiquetados como renta básica y noticias del resto de los clusters. Esta muestra fue anotada por tres personas, para a continuación calcular con la aplicación el inter-annotator agreement [4], la precisión y la exhaustividad. Los detalles de esta implementación se encuentran en la sección 3.4.7 *Métricas del clasificador de temas*.

Identificación de argumentos:

Una vez identificados los hilos de debate generados por noticias sobre la temática escogida, el siguiente paso fué identificar los argumentos y opiniones expresados dentro de estas conversaciones. Para esto hice un nuevo clústering con las noticias y comentarios de estos hilos.

En la sección 3.5 *Identificación de argumentos* se puede ver el desarrollo e implementación de esta tarea.

Análisis de influencia:

En línea con el objetivo de este trabajo, además del desarrollo de la herramienta, en el apartado 3.7 he hecho una recopilación de métodos, modelos y métricas de análisis de influencia, un campo en expansión centrado en el estudio de la propagación de ideas online. También en el anexo he incluido una breve historia de este campo y un resumen de conceptos teóricos.

2 Prefacio: Evolución del proyecto

2.1 Idea inicial

La idea surgió durante la realización de un trabajo para la asignatura Ciencia de la Web, que consistía en hacer una presentación de artículos relacionados con la misma. Uno de ellos “Tweets, Death, and Rock ‘n’ Roll” [5] estudiaba lo que llamaban “Social Media Mourning” (luto social), un fenómeno colectivo que se da en ocasiones en redes sociales al fallecer alguien famoso.

En este caso estudiaban la respuesta en redes sociales a raíz del suicidio del vocalista de Linkin Park. Los autores del trabajo estudiaron la difusión de estos mensajes bajo dos marcos: cascadas de información [6] y “herd behaviour” (comportamiento de manada). [7]

Esto me recordó al libro “Social Physics” [8] que había leído años antes y me hizo pensar en lo interesante que sería en el trabajo de fin de máster estudiar los fenómenos sociales colectivos. Tener un mejor conocimiento de las dinámicas que se generan online es clave para entender nuestra sociedad actual.

Me interesaba estudiar como se transmiten y afianzan las ideas dentro de comunidades online, estando especialmente interesada en el fenómeno de la polarización, que es un proceso por el cual la opinión pública tiende a concentrarse en dos extremos opuestos enfrentados. Este fenómeno está cada vez más presente y es un problema cada vez más preocupante.

Le propuse esta idea a Oscar Corcho (profesor de la asignatura y tutor de este TFM) y le pedí ayuda para concretar el trabajo en algo abarcable durante el periodo de tres meses que debe durar la realización de un TFM. Él me propuso que cogiera un modelo sociológico y validara con datos si el modelo funciona.

También me recomendó que utilizara datos de un foro, en vez de redes sociales, ya que los textos suelen ser mas largos y por tanto resultan mas adecuados para esta tarea.

2.2 Planificación

Aterrizando mas la idea, pensé que lo que tenía mas sentido era identificar debates sobre temas concretos dentro del foro (ej: cambio climático, renta básica, etc.), identificar las distintas posturas (ej: negacionistas del cambio climático vs ecologistas, a favor y en contra de la renta básica, etc.) y por último identificar los argumentarios de estas posturas, es decir, los diferentes argumentos que los defensores de cada una de estas posturas suelen repetir constantemente.

A la hora de afrontar esta tarea, planeé utilizar una mezcla de tareas manuales y automáticas.

Lo primero era necesario identificar los temas dentro del foro, luego los debates existentes dentro de cada tema y las posturas dentro de cada debate, para por último identificar los argumentarios.

Una vez identificados los argumentarios, el siguiente paso sería recopilar un set de ejemplos de cada uno de los argumentos, debidamente etiquetados, para por último utilizar este set de ejemplos para entrenar un clasificador.

Una vez ya entrenado el clasificador, este se podría utilizar para detectar y etiquetar el uso de los distintos argumentos en los comentarios de los usuarios del foro.

La información de las publicaciones en los foros incluye la fecha en que fueron publicados y el usuario que lo publica, por lo que, una vez etiquetadas las publicaciones, se puede utilizar esta información para validar el modelo sociológico, ya que lo que tendríamos es la evolución en el tiempo de la expresión de ideas dentro de una comunidad.

Con la intención de encontrar el modelo sociológico que mejor se adaptara a este proyecto, empecé a investigar y recopilar documentación sobre el tema y encontré que existe un campo específico dedicado al estudio de la transmisión de ideas y dinámicas influencia entre usuarios llamado Análisis de Influencia. El resultado de este trabajo de investigación se puede ver en la *sección 3.7* y en el anexo.

2.3 Reajuste del alcance del proyecto

Durante el desarrollo del trabajo, al ir definiendo la implementación de los pasos, fue siendo cada vez más evidente que la realización de todo el plan requería bastante más tiempo de lo que estaba pensado para un trabajo fin de máster, por lo que se decidí recortar el alcance del mismo para dejar los pasos que no se pudieran alcanzar como parte de un desarrollo posterior.

La idea de identificar y etiquetar ejemplos de temas y argumentos para posteriormente entrenar clasificadores ha sido sustituida por un clústering automático, acompañado por una búsqueda de palabras clave.

La recopilación de modelos y metodologías para el análisis de influencia queda plasmada como base para un trabajo futuro.

3 Desarrollo

3.1 Plan para la implementación

Con el objetivo de llevar a cabo la implementación de la idea original, se definió un listado de tareas que se exponen a continuación.

1. **Fuente de los datos:** Encontrar una comunidad online, preferiblemente un foro, de la que obtener los datos para el estudio. Preparar los datos e importarlos al sistema.
2. **Identificar los temas (topics):** Identificar dentro del sistema los temas de los que se ha hablado en esa comunidad durante el periodo a estudiar.
 - 2.1 Seleccionar un tema y obtener un set de ejemplos de entrenamiento debidamente etiquetados del mismo.
 - 2.2 Entrenar con los ejemplos un clasificador capaz de identificar los textos que tratan sobre el tema escogido en el paso anterior y aplicarla para etiquetar los textos del sistema de esa temática.
3. **Identificar los argumentarios:** Esta tarea consiste en identificar, dentro de la temática seleccionada, los debates que se han originado, así como los argumentos a favor y en contra de las diferentes posturas dentro de ese debate. Para esto se aplican los mismos pasos que para la identificación de temas.
4. **Comportamiento de los usuarios:** Inferir qué usuarios han hablado del tema seleccionado y expresado los argumentos identificados, así como qué usuarios están expuestos a esos argumentos a lo largo del tiempo (considerando que un usuario que participa en un hilo donde se ha expresado un argumento, está expuesto al mismo).
5. **Análisis de influencia:** Por último, utilizando un modelo de análisis de influencia, estudiar los fenómenos de influencia dentro de la comunidad así como las dinámicas de la formación de la opinión colectiva y la polarización de la misma si se diera.

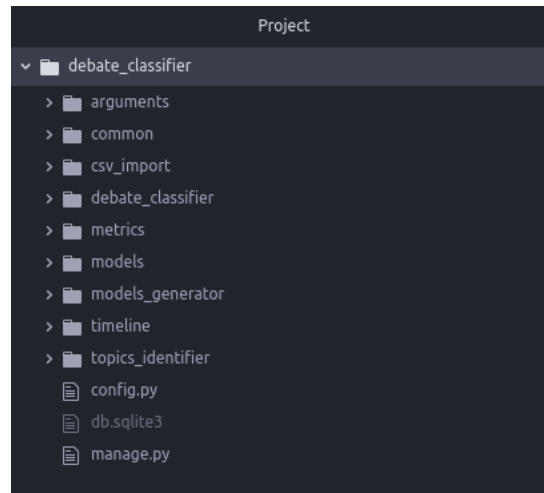
3.2 Estructura de la aplicación

Para facilitar la extracción, procesamiento y posterior visualización de los datos se ha optado por implementar una aplicación web, construida con Python y Django[9]. Para la clasificación de los datos se ha utilizado Scikit-learn[10], una librería de Python enfocada al aprendizaje automático.

El proyecto se llama Debate Classifier[11] y consta de varios módulos (llamados apps en Django):

- **csv_import:** Módulo encargado de importar los datos al sistema.
- **models_generator:** Genera y guarda los modelos entrenados con Scikit-learn.
- **topic_identifier:** Herramienta para identificar las temáticas (topics).
- **arguments:** Herramienta para identificar los argumentos.

- **timeline:** Módulo que contiene los hilos de noticias con sus comentarios, incluidas sus fechas de publicación y los autores que publican cada comentario o noticia.
- **metrics:** Modulo encargado de calcular y mostrar las distintas métricas como la fiabilidad de la muestra (inter-annotator agreement), la precisión y exhaustividad del clasificador.



Además de los módulos (apps), la aplicación contiene otras tres carpetas:

- **common:** Contiene funciones utilizadas en varias de las apps.
- **debate_classifier:** Directorio creado por Django que contiene archivos de configuración del proyecto.
- **models:** Directorio donde se guardan los modelos entrenados con Scikit-learn.

3.3 Fuente de los datos

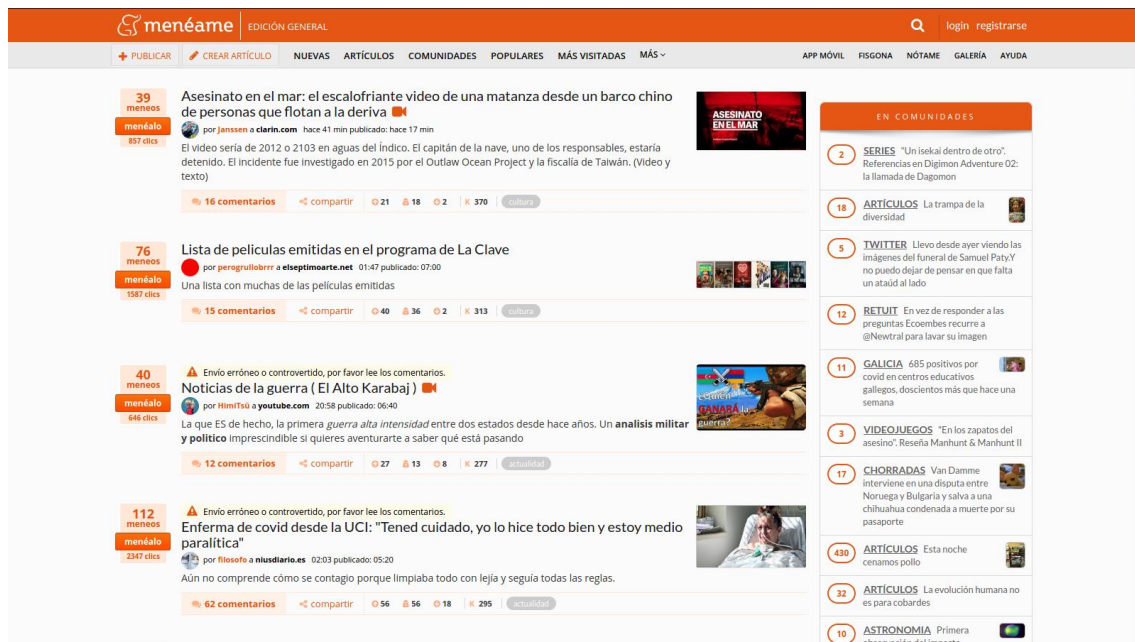
3.3.1 Plataforma escogida

El primer paso fue conseguir una buena fuente de datos que poder analizar.

Al estallar la pandemia y declararse el estado de alarma, pensamos que sería muy interesante trabajar con datos lo más actuales posible, por lo que contactamos a los gestores de la plataforma Menéame para proponerles la idea y pedirles permiso para usar sus datos.

Se eligió esta plataforma porque cuenta con una comunidad de usuarios muy activa, que comparten y discuten noticias a diario, la mayoría de ellas de actualidad. Esto hace de Menéame un caso de estudio idóneo para este trabajo.

Wikipedia [1]: *Menéame es un sitio web y red social basado en la participación comunitaria en el que los usuarios registrados envían historias que los demás usuarios del sitio [...] pueden votar, promoviendo las más votadas a la página principal mediante la aplicación de un algoritmo [...].*



Los gestores de Menéame accedieron a proporcionarnos los datos de las noticias y comentarios publicados en su plataforma durante el primer semestre de 2020, que cubre el periodo del primer estado de alarma y los dos meses previos.

Los datos proporcionados son todos públicos, ya que cualquiera puede acceder a través de su web y leerlos pero, además de ser necesario contar con su permiso para utilizar sus datos, el disponer de ellos en un formato fácil de importar a nuestro sistema ha simplificado mucho el trabajo.

Los datos fueron proporcionados en tres archivos en formato csv, dos archivos de noticias y otro de comentarios, que contienen los registros de la plataforma del primer semestre de 2020, desde el día 1 de enero hasta finales de mayo.

Noticias	archivo:	meneame_news_2020.csv
	tamaño:	7,5 MB. 9.069 registros

	archivo:	missing_news.csv
	tamaño:	32,1 MB. 50.846 registros
Comentarios	archivo:	meneame_comments_2020.csv
	tamaño:	371,9 MB. 1.407.298 registros

El segundo archivo de noticias, “missing_news.csv”, corresponde a un segundo archivo de noticias que se solicitó a Menéame al encontrarnos que había una gran mayoría de comentarios que correspondían a noticias que no se encontraban en el primer archivo.

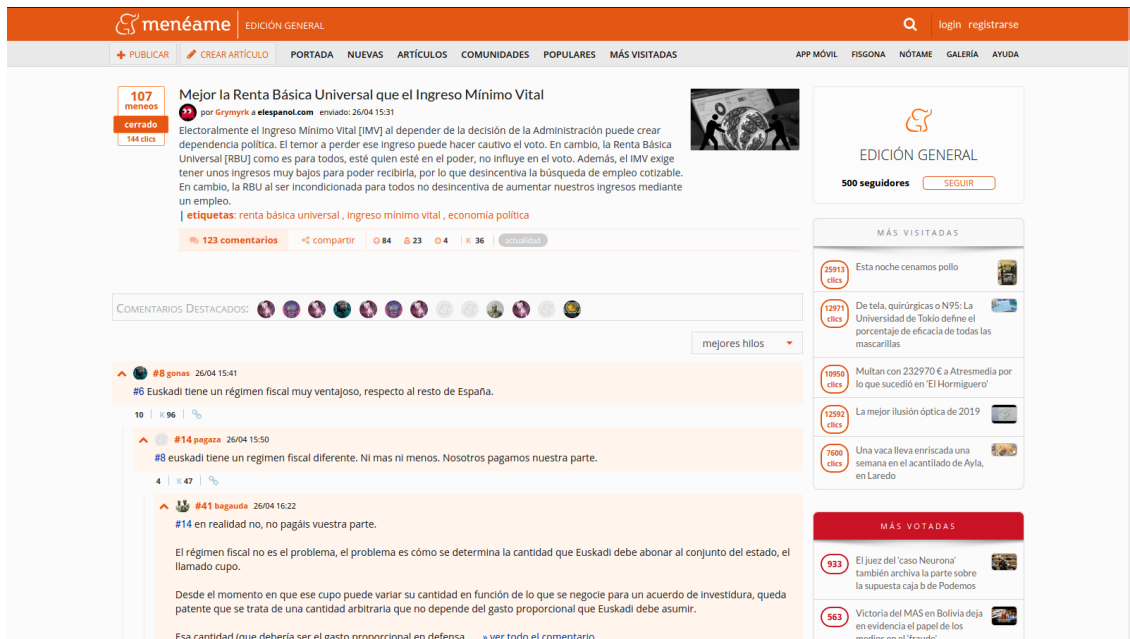
3.3.2 Estructura de los archivos

3.3.2.1 Noticias (links):

Cada noticia es compartida por un usuario. Una vez compartida otros usuarios la votan y la comentan. En Menéame se denominan links en vez de noticias, ya que siempre se trata de un link que referencia a un sitio web externo y estos no siempre se tratan de noticias. A efectos de simplificar la nomenclatura en este trabajo, y ya que la palabra link puede ser fácilmente confundida, llamaremos de ahora en adelante noticias a estos links siempre que hagamos referencia a estos datos.

Los archivos csv de noticias cuentan con las siguientes columnas:

- **link_id:** Número que identifica a la noticia.
- **link_author:** Número que identifica al usuario que ha compartido la noticia
- **link_date:** Fecha y hora en la que se ha compartido la noticia.
- **link_uri:** Identificador único de la noticia. A diferencia de link_id este no es numérico, si no que se saca a partir del título de la noticia.
- **link_url_title:** Título original de la noticia.
- **link_title:** Título de la noticia en Menéame, escrita por el usuario que la ha compartido.
- **link_content:** Descripción de la noticia, escrita por el usuario que la ha compartido.



3.3.2.2 Comentarios:

Una vez colgada la noticia, los usuarios de la plataforma añaden comentarios a la misma y también se responden unos a otros formando conversaciones.

La información disponible en el archivo csv de comentarios es:

- **comment_id:** Número identificador del comentario.
- **comment_link_id:** Número identificador de la noticia a la que hace referencia el comentario. Corresponde a la primera columna "link_id", de los archivos de noticias.
- **comment_user_id:** Número identificador del usuario que hace el comentario.
- **comment_date:** Fecha y hora de publicación del comentario.
- **comment_content:** El comentario en sí.

3.3.3 Importar los datos al sistema

3.3.3.1 Estructura de los datos dentro de la aplicación

Los datos se van a utilizar para varias tareas diferentes, por lo que la información necesaria para esas tareas es distinta.

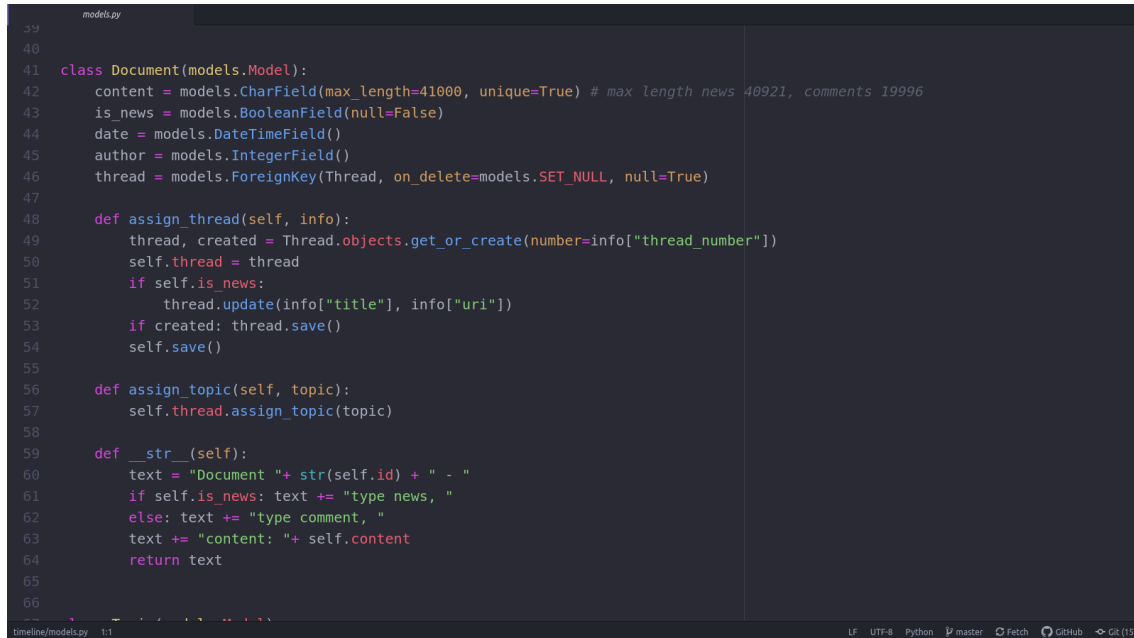
Para las tareas de identificación de temáticas y argumentarios, necesitaremos el contenido compartido por los usuarios durante el periodo a estudiar: link_title y link_content para las noticias, comment_content para los comentarios.

Para el estudio del comportamiento de la comunidad y la difusión e influencia de ideas, no sólo se necesita el contenido compartido, también se necesita saber qué usuarios comparten qué contenido y cuando, además de asociar los comentarios a la noticia sobre la cual están comentando para inferir la exposición de los usuarios al contenido.

Teniendo todo esto en cuenta se decidió la siguiente estructura de los datos dentro de la aplicación:

Clase Document

Esta clase contiene la información tanto de noticias como comentarios. Cada documento apunta a un hilo (thread).



```
39
40
41 class Document(models.Model):
42     content = models.CharField(max_length=41000, unique=True) # max length news 40921, comments 19996
43     is_news = models.BooleanField(null=False)
44     date = models.DateTimeField()
45     author = models.IntegerField()
46     thread = models.ForeignKey(Thread, on_delete=models.SET_NULL, null=True)
47
48     def assign_thread(self, info):
49         thread, created = Thread.objects.get_or_create(number=info["thread_number"])
50         self.thread = thread
51         if self.is_news:
52             thread.update(info["title"], info["uri"])
53             if created: thread.save()
54             self.save()
55
56     def assign_topic(self, topic):
57         self.thread.assign_topic(topic)
58
59     def __str__(self):
60         text = "Document "+ str(self.id) + " - "
61         if self.is_news: text += "type news, "
62         else: text += "type comment, "
63         text += "content: "+ self.content
64         return text
65
66
```

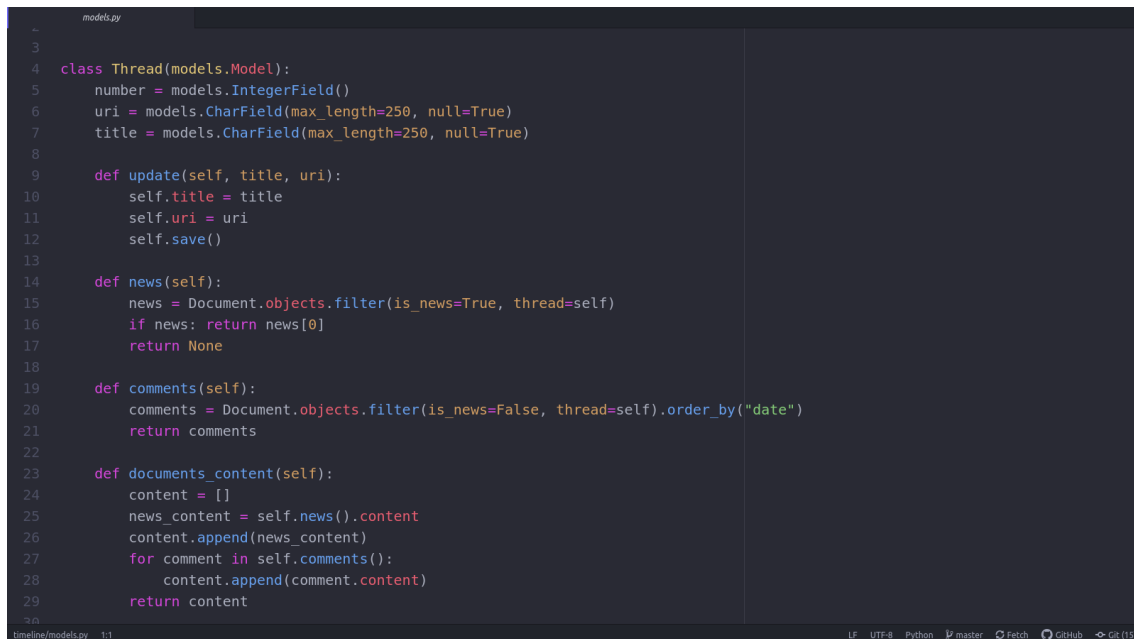
Atributos de la clase:

- content → Contiene comment_content en el caso de los comentarios.
Contiene link_title + salto de línea + link_content para las noticias.
- is_news → True si es una noticia. False si es un comentario.
- date → Fecha de publicación de la noticia o comentario.
link_date para noticias, comment_date para comentarios.
- author → Identificador del usuario que ha publicado la noticia o comentario.
link_author para noticias, comment_date para comentarios.
- thread → id de la clase Thread (hilo) que asocia este documento con el resto de documentos referentes a la misma noticia y por tanto al mismo hilo de discusión. Se asigna utilizando los identificadores contenidos en link_id para las noticias y comment_link_id para los comentarios.

Nota: Se ha optado por incluir el título dentro del contenido de las noticias porque el contenido de los documentos es lo que se proporciona al clasificador, y los textos de los títulos, aunque a veces se repiten dentro de la descripción de la noticia, otras veces no, por lo que pueden contener información útil para el clasificador.

Clase Thread

Esta clase es la encargada de unir cada noticia con sus comentarios. Representa la estructura de hilos de debate. A cada hilo apunta una noticia y los comentarios que responden a esa noticia.



```
models.py
1
2
3
4 class Thread(models.Model):
5     number = models.IntegerField()
6     uri = models.CharField(max_length=250, null=True)
7     title = models.CharField(max_length=250, null=True)
8
9     def update(self, title, uri):
10         self.title = title
11         self.uri = uri
12         self.save()
13
14     def news(self):
15         news = Document.objects.filter(is_news=True, thread=self)
16         if news: return news[0]
17         return None
18
19     def comments(self):
20         comments = Document.objects.filter(is_news=False, thread=self).order_by("date")
21         return comments
22
23     def documents_content(self):
24         content = []
25         news_content = self.news().content
26         content.append(news_content)
27         for comment in self.comments():
28             content.append(comment.content)
29         return content
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
262
```

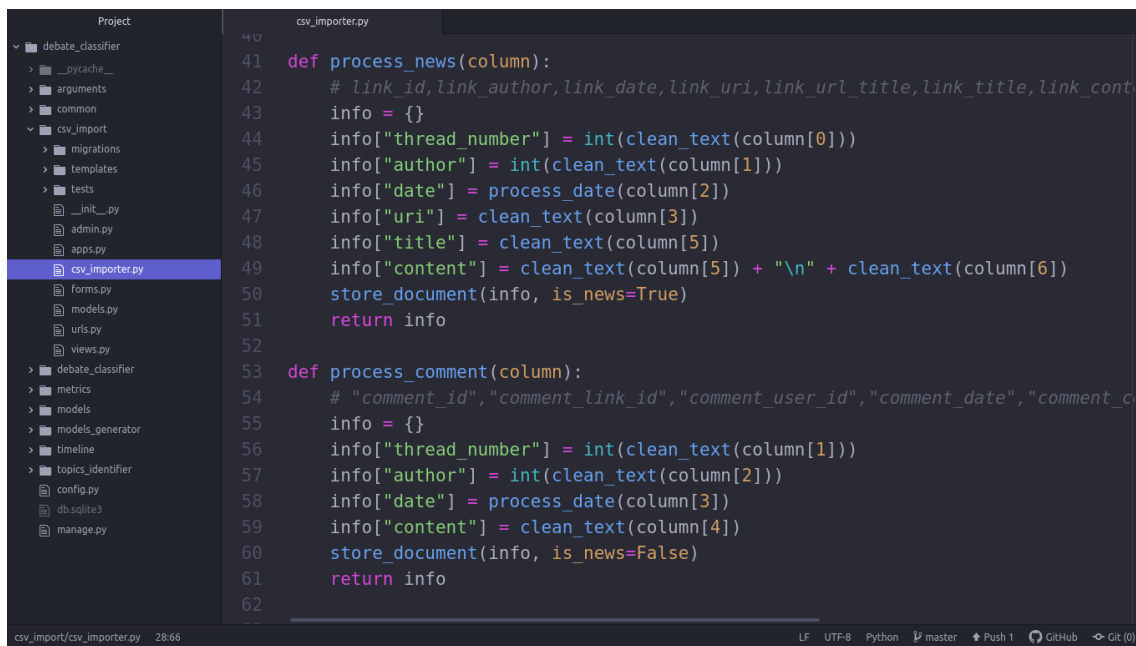
Import CSV file

File:

Examinar... meneame_comments_2020.csv

Upload

Una vez seleccionado el archivo y cargado a través del formulario, este es procesado por `csv_importer.py` del módulo `csv_import`, que genera documentos con el contenido de las noticias y los comentarios, además de generar y asignar los hilos correspondientes.



```
41 def process_news(column):
42     # link_id, link_author, link_date, link_uri, link_url_title, link_title, link_cont
43     info = {}
44     info["thread_number"] = int(clean_text(column[0]))
45     info["author"] = int(clean_text(column[1]))
46     info["date"] = process_date(column[2])
47     info["uri"] = clean_text(column[3])
48     info["title"] = clean_text(column[5])
49     info["content"] = clean_text(column[5]) + "\n" + clean_text(column[6])
50     store_document(info, is_news=True)
51     return info
52
53 def process_comment(column):
54     # "comment_id", "comment_link_id", "comment_user_id", "comment_date", "comment_c
55     info = {}
56     info["thread_number"] = int(clean_text(column[1]))
57     info["author"] = int(clean_text(column[2]))
58     info["date"] = process_date(column[3])
59     info["content"] = clean_text(column[4])
60     store_document(info, is_news=False)
61     return info
62
```

```

csv_importer.py
22
23 def store_document(info, is_news):
24     doc_search = Document.objects.filter(content=info["content"])
25     if document_exist(info):
26         save_duplicated_document(info)
27     else:
28         doc = Document(content=info["content"], is_news=is_news,
29                        date=info["date"], author=info["author"])
30         doc.assign_thread(info)
31

```

En las primeras versiones del código, no se generaban los documentos en la base de datos, si no que el contenido de link_comment, link_title y link_content se guardaba en archivos txt que en pasos posteriores del proceso se utilizaban para generar un dataset.

Los detalles sobre esto se pueden leer en el apartado 3.4.2 *Entrada de datos*.

3.4 Identificación de los temas

En la plataforma con cuyos datos estamos trabajando los usuarios publican noticias que el resto de usuarios comentan generando hilos de discusión. Por tanto es razonable asumir que para la mayoría de los casos cada hilo, con sus noticias y sus comentarios, van a tratar sobre un mismo tema. En algunos casos no será así, ya que a veces las discusiones se desvían y hablan sobre otros temas, pero por simplificar la tarea vamos a ignorar esto.

Por tanto la clasificación por temas se va a realizar por hilos (threads en el código), asumiendo que cada noticia es representativa del tema a tratar. Así que el clustering se ha hecho con las noticias, que se han utilizado para etiquetar con temas (topics en el código) los hilos.

3.4.1 Selección del método de clasificación

3.4.1.1 Clasificación supervisada y no supervisada

Para identificar las temáticas y los argumentarios habría sido interesante poder usar redes neuronales pre-entrenadas para procesamiento de lenguaje natural como por ejemplo BERT[12] pero, a pesar de que este tipo de redes requieren menos entrenamiento al estar ya preparadas para tratar con tareas de reconocimiento de lenguaje, aún así requieren de algo de entrenamiento (llamado fine-tuning) para aprender a realizar la tarea específica que se requiere y no disponíamos de datos etiquetados.

Una posibilidad habría sido realizar el etiquetado de los temas y/o de los argumentos externalizando la tarea con algún servicio de crowdsourcing como Amazon Mechanical Turk[13] o similares, pero esto presentaba varios problemas:

- 1- Para empezar a etiquetar, especialmente si se quiere externalizar, en cuyo caso es mucho más eficiente convertirla en una tarea tipo formulario

con un número de opciones concretas, es necesario tener una idea previa de las categorías a etiquetar, en este caso los temas y los argumentos.

2- Desconocía cuantos ejemplos etiquetados serían necesarios para hacer el fine-tuning y también cuantos documentos de cada categoría podía llegar a haber en los datos de los que disponíamos, pero parecía que, al menos para entrenar una red capaz de clasificar las noticias por temas, no serían suficientes. En el tutorial BERT for dummies [14] se utiliza un dataset de 10.000 ejemplos para un clasificador que sólo distingue una categoría y el total de noticias que disponemos no llega a 60.000.

3- Tanto el tiempo como los recursos disponibles para hacer el TFM eran limitados, por lo que no parecía recomendable intentar esta opción sin antes haber despejado algunas incógnitas.

Por ello, decidí hacer primero una clasificación por clustering con las noticias, ya que, al ser un método no supervisado, podía utilizarlo para hacer una primera clasificación que me facilitara la tarea de identificar los temas o al menos hacerme una idea de cuantas podía haber y cuantos ejemplos había aproximadamente de cada una.

Para esta tarea elegí la herramienta Scikit-learn, una librería de Python enfocada al aprendizaje automático que ya había utilizado durante las prácticas de la asignatura Ingeniería Lingüística y que ofrece varios algoritmos de clustering [15].

3.4.1.2 Clustering con k-means y Affinity Propagation

Primero probé a hacer clustering con el algoritmo k-means [16] utilizando el tutorial “Applying Machine Learning to classify an unsupervised text document” [17]. Como es necesario especificar el número de clusters, lo hice por iteraciones, probando con diferentes números de clusters.

Aquí [18] se puede ver la versión antigua del clasificador implementada con k-means:

```
def cluster_with_kmeans(documents, processed_data, num_clusters):
    # Build the model
    model = KMeans(n_clusters=num_clusters, init='k-means++', max_iter=100, n_init=1)
    # Train the model with the pre processed data
    model.fit(processed_data["vectorized documents"])
    # predict the categories for each document
    documents_predicted_clusters = model.predict(processed_data["vectorized documents"])
    # get the terms selected for each cluster
    clusters_terms = get_clusters_terms(model, processed_data["terms"], num_clusters)
    clustered_documents = group_documents_by_cluster(documents, documents_predicted_clusters, clusters_terms, num_clusters)
    return clustered_documents
```

Los resultados no fueron muy buenos y el método no resultaba práctico para esta tarea, ya que requería repetir el proceso de clustering por cada número de clusters que se quería probar y comparar los resultados de las distintas iteraciones.

A continuación probé con Affinity Propagation[2], cuya ventaja es que el propio algoritmo decide el número de clusters, y conseguí mucho mejores resultados. Este algoritmo se recomienda para sets de datos pequeños y medianos, al ser bastante complejo y requerir de muchos recursos para sets de datos grandes.

El funcionamiento de Affinity Propagation se explica en la sección 3.4.4.1 *Generación de clusters con Affinity Propagation*.

3.4.2 Entrada de datos

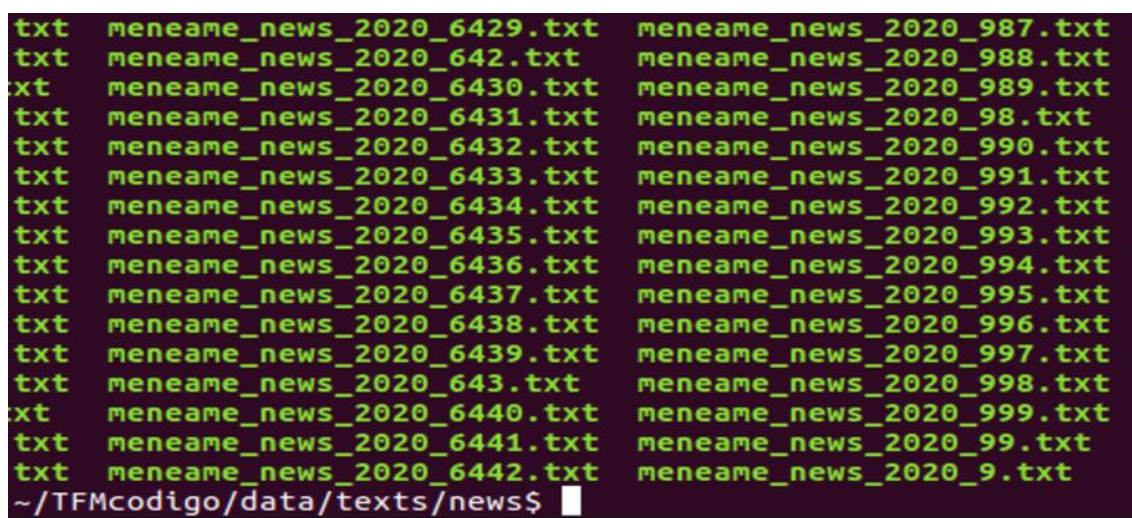
3.4.2.1 Versión con datasets de Scikit-learn

Consultando algunos tutoriales [17] y la documentación oficial de Scikit-learn [19] se puede observar en los ejemplos que la entrada de datos para entrenar los algoritmos son dataset de Scikit-learn [20].

La documentación sobre datasets habla en su mayoría sobre como utilizar datasets ya creados y el punto en el que hablan de importar otros datasets [21] recomienda usar otras herramientas como pandas o numpy para crearlos o utilizar el método de Scikit-learn `datasets.load_files` [22].

El método `datasets.load_files` requiere que se disponga de los documentos en formato txt dentro de carpetas, que la librería utilizará para crear el dataset.

En las primeras versiones de nuestra aplicación esta leía los ficheros csv para generar los ficheros txt, uno por cada registro importado del csv, en las carpetas `/data/texts/news` o `/data/texts/comments`, dependiendo del tipo de archivo csv que se estuviera procesando.



```
txt  meneame_news_2020_6429.txt  meneame_news_2020_987.txt
txt  meneame_news_2020_642.txt  meneame_news_2020_988.txt
xt   meneame_news_2020_6430.txt  meneame_news_2020_989.txt
txt  meneame_news_2020_6431.txt  meneame_news_2020_98.txt
txt  meneame_news_2020_6432.txt  meneame_news_2020_990.txt
txt  meneame_news_2020_6433.txt  meneame_news_2020_991.txt
txt  meneame_news_2020_6434.txt  meneame_news_2020_992.txt
txt  meneame_news_2020_6435.txt  meneame_news_2020_993.txt
txt  meneame_news_2020_6436.txt  meneame_news_2020_994.txt
txt  meneame_news_2020_6437.txt  meneame_news_2020_995.txt
txt  meneame_news_2020_6438.txt  meneame_news_2020_996.txt
txt  meneame_news_2020_6439.txt  meneame_news_2020_997.txt
txt  meneame_news_2020_643.txt   meneame_news_2020_998.txt
xt   meneame_news_2020_6440.txt  meneame_news_2020_999.txt
txt  meneame_news_2020_6441.txt  meneame_news_2020_99.txt
txt  meneame_news_2020_6442.txt  meneame_news_2020_9.txt
~/TFMcodigo/data/texts/news$
```

Una vez hecho esto, se podía generar el dataset por medio de la interfaz web. El dataset se generaba utilizando todos los archivos de texto que se encontraran en la carpeta `texts` dentro de la subcarpeta. Estos archivos se proporcionaban como entrada al método `datasets.load_files` y este devolvía el dataset.

Topics

[Import files](#) | [Generate dataset](#) | [Cluster data](#)

Generate dataset

- This field is required.

Dataset name:

- This field is required.

Description:

A continuación el dataset era guardado por la aplicación en la carpeta data/texts_datasets, en formato NumPy.

Por último se utilizaba el dataset para hacer el clustering. Los cluster resultantes se almacenaban en la base de datos.



Topics

[Import files](#) | [Generate dataset](#) | [Cluster data](#)

- This field is required.

Dataset name:

Esto resultaba tremendamente ineficaz, ya que para generar un dataset con los comentarios era necesario crear una carpeta con mas de un millon de ficheros de texto, que resultaba complicado de gestionar para el sistema operativo (sólo con intentar abrir la carpeta el sistema se ralentizaba seriamente). Además, al almacenar el dataset de noticias (con poco más de 9.000 registros) el fichero resultante ocupaba 1,5 GB, por lo que el dataset con los comentarios (que habría tenido más de un millón de registros) podría haber llegado a ocupar alrededor de 200GB.

Afortunadamente, en pasos posteriores, descubrí que no era necesario generar el dataset para la clasificación no supervisada, ya que esta en realidad recibe como entrada un array de strings con el contenido de los documentos (contenidos en dataset.data cuando se usa el dataset).

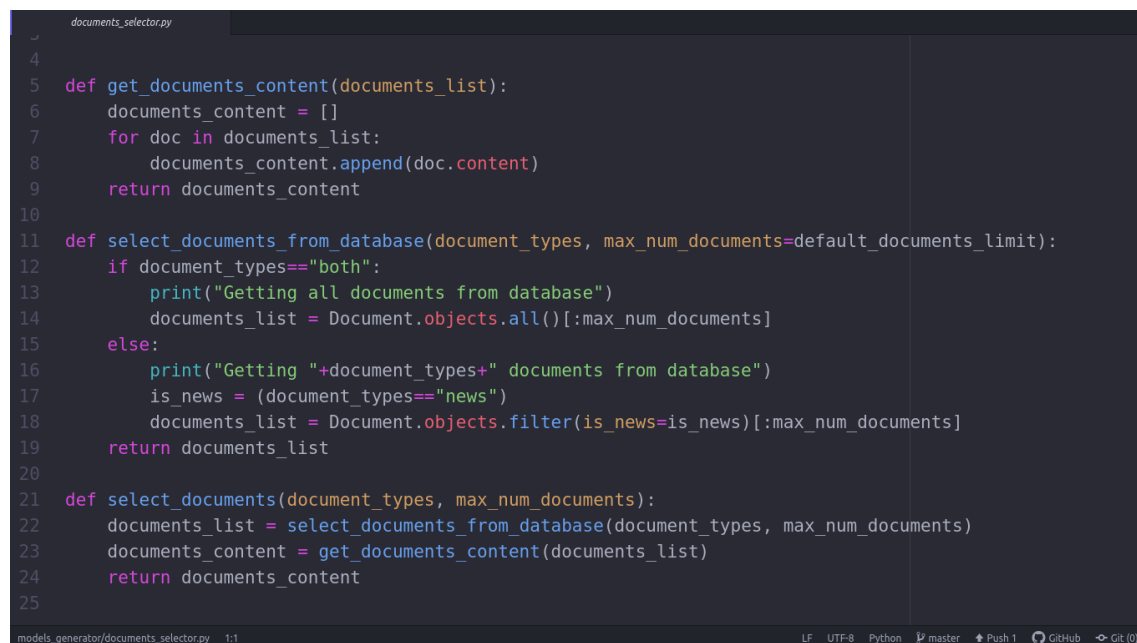
3.4.2.2 Versión sin datasets

La versión actual, sin usar datasets, es muchísimo mas sencilla:

- 1- Se importan los datos como se explica en el punto 2.5.3.
- 2- Cuando es necesario hacer uso de los textos se hace una llamada al método `select_documents` de `documents_selector.py`. Este método devuelve un array de strings con el contenido de los documentos que, dependiendo de si se trata de comentarios o noticias, consistirá en el texto del comentario o el título de la noticia y la descripción de la misma.

Nota: Los módulos `models_generator` y `topics_identifier` cuentan con versiones diferentes de `documents_selector.py`. En un principio se utilizaba el mismo método, pero al tener estos módulos necesidades diferentes, se simplificaba mucho el código haciendo un selector específico para cada uno.

Selector de documentos del módulo `models_generator`:



```
documents_selector.py
4
5 def get_documents_content(documents_list):
6     documents_content = []
7     for doc in documents_list:
8         documents_content.append(doc.content)
9     return documents_content
10
11 def select_documents_from_database(document_types, max_num_documents=default_documents_limit):
12     if document_types=="both":
13         print("Getting all documents from database")
14         documents_list = Document.objects.all()[:max_num_documents]
15     else:
16         print("Getting "+document_types+" documents from database")
17         is_news = (document_types=="news")
18         documents_list = Document.objects.filter(is_news=is_news)[:max_num_documents]
19     return documents_list
20
21 def select_documents(document_types, max_num_documents):
22     documents_list = select_documents_from_database(document_types, max_num_documents)
23     documents_content = get_documents_content(documents_list)
24     return documents_content
25
```

- 3- Por último, el array de strings se proporciona como entrada al vectorizer y al modelo. Mas detalles sobre esto en el siguiente apartado.


```

ModelGenerator.py
6 class ModelGenerator:
7
8     def __init__(self, documents):
9         self.documents = documents
10        self.vectorizer = TfidfVectorizer(stop_words=get_stop_words())
11
12        # Process the documents with the vectorizer.
13    def process_documents(self):
14        print("Procesing "+str(len(self.documents))+ " documents")
15        self.vectorized_documents = self.vectorizer.fit_transform(self.documents)
16        return self.vectorized_documents
17
18    def train_model(self):
19        print("Training model")
20        model = AffinityPropagation()
21        model.fit(self.vectorized_documents)
22        return model
23
24    def generate_model(self):
25        self.process_documents()
26        model = self.train_model()
27        return model
28
models_generator/ModelGenerator.py 1:1
LF UTF-8 Python master Push 1 GitHub Git (0)

```

3.4.3 Extracción de características (Feature extraction)

Tanto para poder utilizar los documentos para entrenar el algoritmo de clustering, como para posteriormente poder clasificar documentos con el modelo entrenado, es necesario preparar previamente estos datos [23].

3.4.3.1 Vectorización

La primera parte de este proceso consiste en convertir el corpus de documentos en representaciones numéricas de los mismos, en forma de vectores. Para ello se siguen tres pasos:

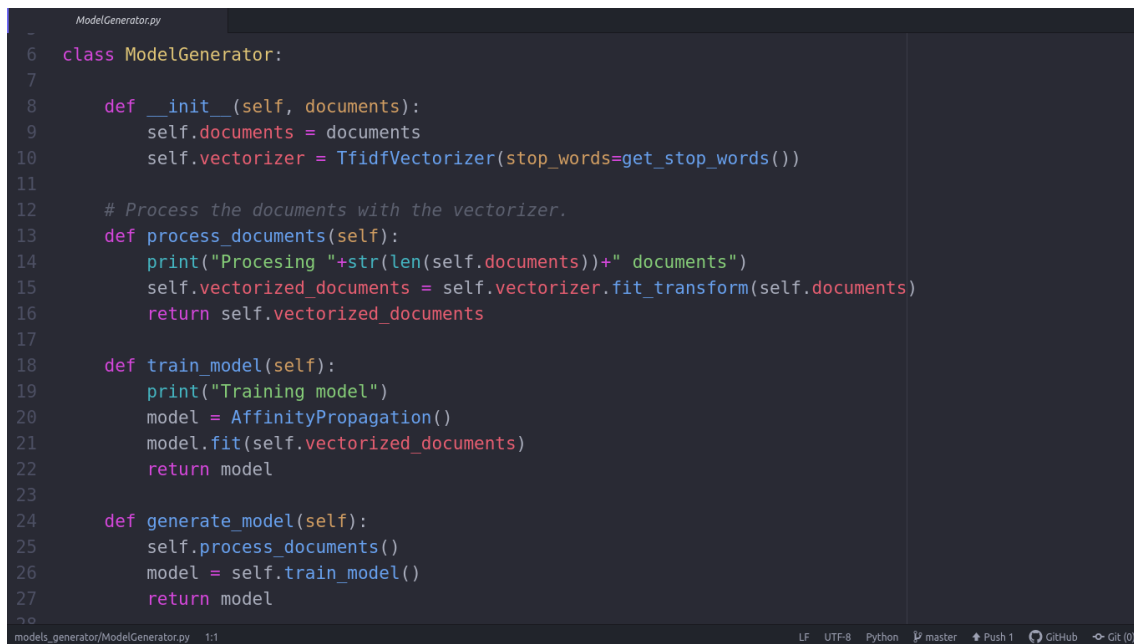
- **Tokenizar:** Se identifican todos los términos (tokens) presentes en la colección de documentos y se asigna un identificador numérico a cada uno.
- **Conteo:** Se cuenta cada una de las veces que cada término aparece en cada uno de los documentos, creando una matriz de términos.
- **Normalizar:** Se normalizan los valores, dando menos peso a los términos que aparecen en la mayoría de los documentos.

Tras realizar este proceso, un corpus de documentos es representado por una matriz con una fila por documento y una columna por cada termino (token).

- Cada fila contiene un vector con todas las frecuencias de términos para un documento (multivariate sample).
- Cada una de estas frecuencias de términos se considera una característica (feature).

Scikit-learn dispone de dos métodos que realizan estas tareas: `CountVectorizer` y `TfidfVectorizer`.

- **CountVectorizer** implementa la tokenización y el conteo en una única clase.
- **TfidfVectorizer** además de la tokenización y el conteo, implementa un ajuste de pesos llamado Tf-idf, que se explica en el apartado 3.4.3.3.



```
ModelGenerator.py
6 class ModelGenerator:
7
8     def __init__(self, documents):
9         self.documents = documents
10        self.vectorizer = TfidfVectorizer(stop_words=get_stop_words())
11
12    # Process the documents with the vectorizer.
13    def process_documents(self):
14        print("Procesing "+str(len(self.documents))+ " documents")
15        self.vectorized_documents = self.vectorizer.fit_transform(self.documents)
16        return self.vectorized_documents
17
18    def train_model(self):
19        print("Training model")
20        model = AffinityPropagation()
21        model.fit(self.vectorized_documents)
22        return model
23
24    def generate_model(self):
25        self.process_documents()
26        model = self.train_model()
27        return model
```

3.4.3.2 Palabras comunes (stop words)

Para mejorar la eficiencia del algoritmo se pueden descartar las palabras más comunes que se espera encontrar en todos los documentos y que no aportan información útil a la hora de realizar la clasificación de los textos.

Estas palabras, denominadas stop words [24], se pueden pasar como parámetro al crear una instancia de `CountVectorizer` o `TfidfVectorizer`.

Las stop words consisten simplemente en un array de strings, que contiene las palabras más comunes. Se puede crear a medida para el tipo de documentos que estamos tratando o se puede utilizar uno de los ya creados para el idioma de los documentos, en este caso el español.

Las stop words que hemos utilizado en la aplicación se encuentran disponibles en github [25].



```
1 stop_words_filename = "models_generator/stop_words/stop_words_spanish.txt"
2
3 def get_stop_words():
4     stop_words = []
5     file = open(stop_words_filename, 'r')
6     words_from_file = file.read().split("\n")
7     file.close()
8     for word in words_from_file:
9         stop_words.append(word)
10    return stop_words
11
```

3.4.3.3 Ajuste de pesos Tf-idf

En un corpus de documentos, aún después de haber eliminado las palabras más comunes en el idioma por medio de las stop words, puede haber algunos términos que sean comunes en muchos de los documentos, y por tanto no resulten muy útiles para agrupar los documentos en clústers.

Con el algoritmo Tf-idf (Term Frequency–Inverse Document Frequency) se realiza un ajuste de pesos sobre la matriz creada con la tokenización y el conteo (ver sección 3.4.3.1), reduciendo la importancia de los términos que aparecen más a menudo en muchos documentos.

La clase **TfidfVectorizer** de Scikit-learn, utilizada en nuestra aplicación, se encarga de tokenizar, realizar el conteo y ajustar los pesos según la técnica Tf-idf.

3.4.4 Agrupación de textos por temáticas

3.4.4.1 Generación de clusters con Affinity Propagation

Una vez tenemos los textos preparados, el siguiente paso es agrupar estos textos por temáticas. Para ello, como se explica en el punto 3.4.1 *Selección del método de clasificación*, se ha optado por usar el algoritmo de clustering Affinity Propagation.

La estrategia de Affinity Propagation es la siguiente:

- Crea los clusters mandando mensajes entre pares de ejemplos hasta que converge. Por medio de estos mensajes se identifica como de bueno es un ejemplo para ser representativo del otro ejemplo.
- Crea un pequeño dataset con unos cuantos ejemplos seleccionados que ha identificado en el paso anterior como los más representativos del total de la muestra.
- El proceso se repite de forma iterativa hasta que converge, actualizando la puntuación que representa cómo de bien cada ejemplo representa al

otro con el que se está comparando teniendo en cuenta los valores del resto de pares de ejemplos.

- El resultado final es una selección de ejemplos que se consideran como los más representativos de la muestra, a los que se llama documentos de referencia, alrededor de los cuales se agrupan los clusters.

El primer clústering con este método sobre el dataset de noticias resultó en algo mas de 1000 clusters.

Muchos de los clusters tenían bastante sentido, agrupando por temáticas relacionadas. Sin embargo me pareció que 1000 clusters era una cantidad difícil de manejar, por lo que se me ocurrió implementar un clustering en arbol, con varios niveles.

3.4.4.2 Clusters en dos niveles (arboles)

Los clusters creados con AffinityPropagation, se generan alrededor de un documento de referencia. La idea del clustering multinivel es coger los documentos de referencia para crear un nuevo dataset (aquí es cuando descubrí que podía crear los datasets manualmente), y aplicar de nuevo Affinity Propagation a este nuevo dataset. A los 176 clusters resultantes les asigné el nivel 1, dejando en el nivel 0 a los 1000 clusters generados anteriormente. Por último generé la estructura de árbol asignando a los clusters de nivel 0 el cluster del nivel 1 que contiene a su documento de referencia.

Tree: news_v1

Document types: news

Level 1 - Cluster 0

Terms:

[‘jodido’, ‘preescolar’, ‘bien’, ‘prenatal’, ‘sanidad’, ‘escuelas’, ‘comida’, ‘guarderías’, ‘saber’, ‘oír’, ‘quieren’, ‘meses’, ‘nueve’, ‘feto’, ‘obsesionados’, ‘conservadores’, ‘defensores’, ‘posicionamiento’, ‘crítica’, ‘ácida’, ‘realiza’, ‘humorista’, ‘carlin’, ‘mujer’, ‘anti’, ‘vida’, ‘pro’, ‘después’, ‘george’, ‘gratis’, ‘sí’]

Reference document:

Pro vida es anti mujer - George Carlin El humorista George Carlin realiza una ácida crítica sobre el posicionamiento de los defensores "pro vida". <i>"Los conservadores pro vida están obsesionados con el feto hasta los nueve meses, pero después no quieren oír ni saber nada de ti. Nada, ni guarderías, ni comida en las escuelas, ni sanidad gratis, nada. Si eres prenatal estás bien, si eres preescolar, estás jodido"</i>

Level 1 - Cluster 1

Terms:

[‘guerra’, ‘amazon’, ‘bezos’, ‘eng’, ‘mundo’]

Reference document:

Bezos contra el mundo (ENG) Amazon está en guerra con todo el mundo.

Tree: news_v1 - Level 1 Cluster 0

Level 1 - Cluster 0

Terms:

[‘jodido’, ‘preescolar’, ‘bien’, ‘prenatal’, ‘sanidad’, ‘escuelas’, ‘comida’, ‘guarderías’, ‘saber’, ‘oír’, ‘quieren’, ‘meses’, ‘nuevo’, ‘feto’, ‘obsesionados’, ‘conservadores’, ‘defensores’, ‘posicionamiento’, ‘crítica’, ‘ácida’, ‘realiza’, ‘humorista’, ‘carlin’, ‘mujer’, ‘anti’, ‘vida’, ‘pro’, ‘después’, ‘george’, ‘gratis’, ‘sí’]

Reference document:

Pro vida es anti mujer - George Carlin El humorista George Carlin realiza una ácida crítica sobre el posicionamiento de los defensores "pro vida". <i>"Los conservadores pro vida están obsesionados con el feto hasta los nueve meses, pero después no quieren oír ni saber nada de ti. Nada, ni guarderías, ni comida en las escuelas, ni sanidad gratis, nada. Si eres prenatal estás bien, si eres preescolar, estás jodido"</i>

Documents:

[George Floyd: Miles de personas protestan por la muerte de un afroamericano a manos de policías en EEUU](#)

[Pro vida es anti mujer - George Carlin](#)

[Un anciano muere en Málaga alcanzado por una bala perdida de un tiroteo entre clanes | Málaga](#)

[«Vidas medievales» una serie documental de Terraviva que explica la vida cotidiana en la Edad](#)

Sub clusters:

Level 0 - Cluster 8

Terms:

[‘billete’, ‘sospecha’, ‘rápidamente’, ‘difundiesen’, ‘vídeos’, ‘facilitó’, ‘ocurridos’, ‘presenciaron’, ‘transeúnte’, ‘inmovilizado’, ‘agentes’, ‘pronunció’, ‘respirar’, ‘grito’, ‘marcharon’, ‘manifestantes’, ‘eeuu’, ‘policías’, ‘protestan’, ‘hechos’, ‘usar’, ‘falso’, ‘bajo’, ‘minutos’, ‘miles’, ‘dólares’, ‘haber’, ‘puedo’, ‘sociales’, ‘redes’, ‘manos’, ‘cuello’, ‘rodilla’, ‘20’, ‘mismo’, ‘lunes’, ‘muerte’, ‘floyd’, ‘george’, ‘afroamericano’, ‘policía’, ‘detenido’, ‘personas’, ‘varios’, ‘supermercado’, ‘intentado’, ‘mientras’]

Reference document:

George Floyd: Miles de personas protestan por la muerte de un afroamericano a manos de policías en EEUU "Los manifestantes marcharon al grito de "¡no puedo respirar!", el mismo que pronunció George Floyd mientras uno de los agentes le tuvo inmovilizado durante minutos con la rodilla sobre su cuello" [...] "Varios transeúnte presenciaron los hechos ocurridos el lunes con Floyd, lo que facilitó que vídeos se difundiesen rápidamente en las redes sociales. La Policía lo había detenido bajo sospecha de haber intentado usar un billete falso de 20 dólares en un supermercado. "

Documents:

[Detenido el policía implicado en el asesinato del afroamericano George Floyd en Mineápolis](#)

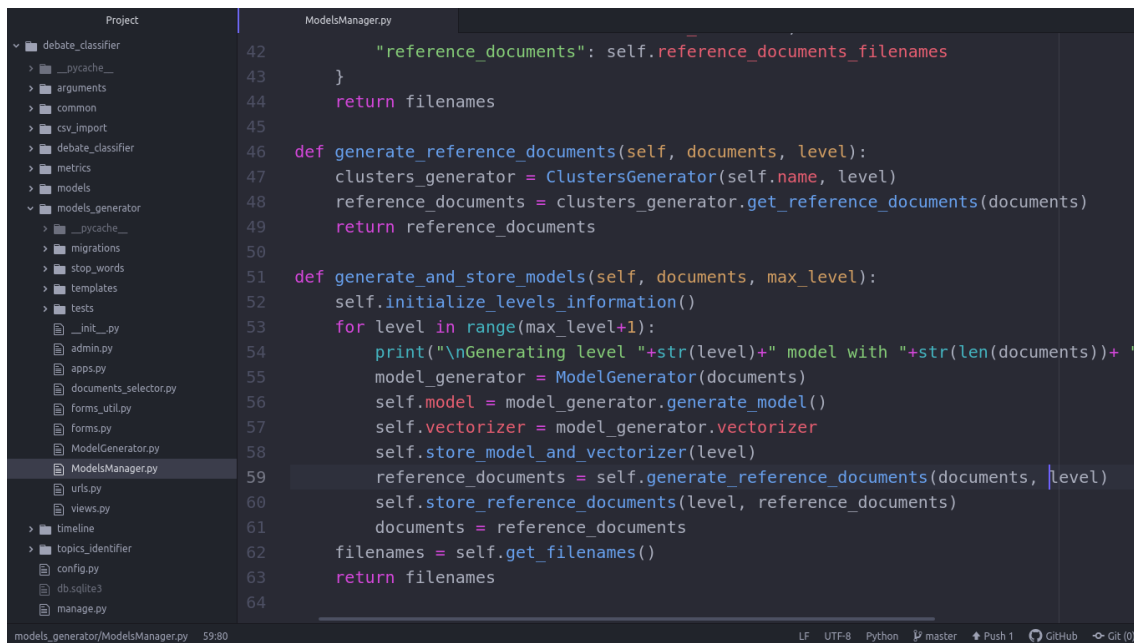
[El policía señalado por la muerte de George Floyd tenía varias denuncias por sus actuaciones policiales](#)

Pensé en crear un cluster de nivel 2, para reducir aún mas el número de clusters, pero lo descarté porque vi que con dos niveles de clusters unido a una búsqueda por palabras clave era suficiente.

3.4.5 Guardar los modelos

Inicialmente generaba el modelo de Affinity Propagation y a continuación el árbol de clusters en la misma operación, pero sin guardar el modelo, lo cual obligaba a volver a generarlo de nuevo cada vez que se quería utilizar.

Decidí separar la generación de los modelos de la generación de árboles, creando el módulo `model_generator`.

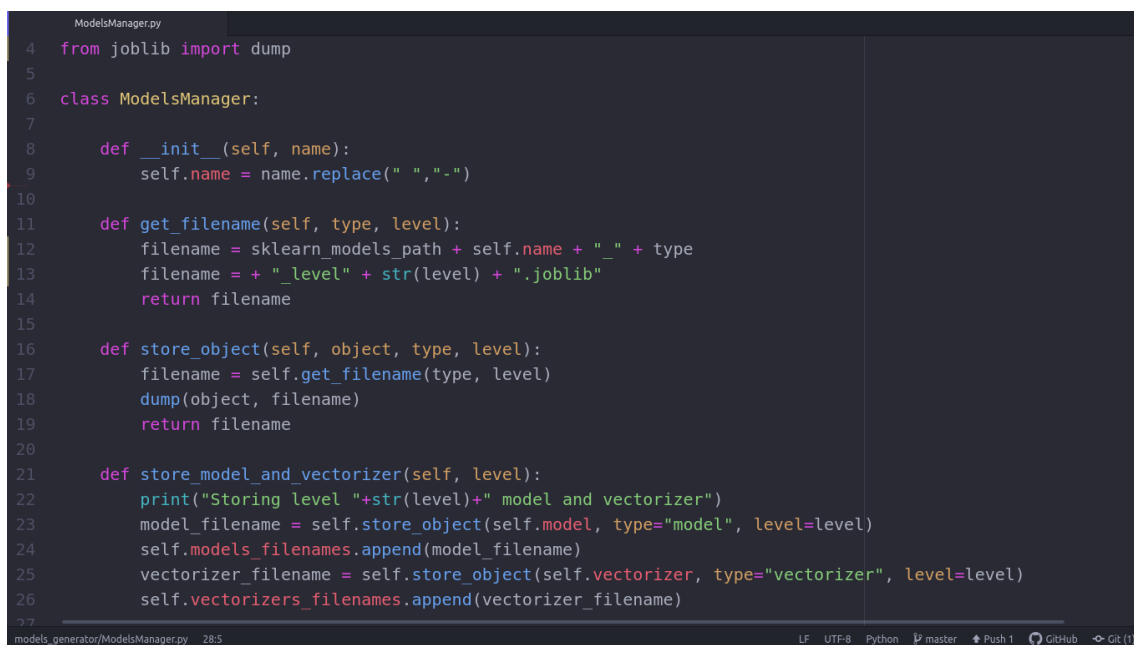


```
42         "reference_documents": self.reference_documents_filenames
43     }
44     return filenames
45
46 def generate_reference_documents(self, documents, level):
47     clusters_generator = ClustersGenerator(self.name, level)
48     reference_documents = clusters_generator.get_reference_documents(documents)
49     return reference_documents
50
51 def generate_and_store_models(self, documents, max_level):
52     self.initialize_levels_information()
53     for level in range(max_level+1):
54         print("\nGenerating level "+str(level)+" model with "+str(len(documents))+ '
55         model_generator = ModelGenerator(documents)
56         self.model = model_generator.generate_model()
57         self.vectorizer = model_generator.vectorizer
58         self.store_model_and_vectorizer(level)
59         reference_documents = self.generate_reference_documents(documents, level)
60         self.store_reference_documents(level, reference_documents)
61         documents = reference_documents
62     filenames = self.get_filenames()
63     return filenames
64
```

Para guardar el modelo una vez entrenado en la documentación de Scikit-learn [26] recomiendan dos opciones: pickle y joblib.

- Pickle convierte el modelo en string, que posteriormente se puede usar para volver a cargar el modelo.
- Joblib permite guardar y cargar el modelo desde un fichero externo.

He escogido joblib al ser, según la documentación de Scikit-learn, el método más eficiente.



```
4 from joblib import dump
5
6 class ModelsManager:
7
8     def __init__(self, name):
9         self.name = name.replace(" ", "-")
10
11     def get_filename(self, type, level):
12         filename = sklearn_models_path + self.name + "_" + type
13         filename = + "_level" + str(level) + ".joblib"
14         return filename
15
16     def store_object(self, object, type, level):
17         filename = self.get_filename(type, level)
18         dump(object, filename)
19         return filename
20
21     def store_model_and_vectorizer(self, level):
22         print("Storing level "+str(level)+" model and vectorizer")
23         model_filename = self.store_object(self.model, type="model", level=level)
24         self.models_filenames.append(model_filename)
25         vectorizer_filename = self.store_object(self.vectorizer, type="vectorizer", level=level)
26         self.vectorizers_filenames.append(vectorizer_filename)
27
```

3.4.6 Búsqueda de temáticas con los clusters

Tras agrupar las noticias en clusters vi que podía encontrar fácilmente las noticias de un tema realizando una búsqueda por palabras clave dentro de los árboles clusters.

Esto se hace buscando la palabra o palabras entre las nubes de términos de los clústers y devolviendo aquellos en los que aparezcan todas las palabras, por lo que se recomienda empezar por una sola palabra y, si devuelve demasiados resultados, afinar la búsqueda.

Topics identifier

[Generate clusters tree](#) | [Clusters trees](#) | [Topics](#) | [Label topic with file](#)

Search terms:

renta

Search

Tree: news_v1

Document types: news

Level 1 - Cluster 0

Terms:

[*'jodido', 'preescolar', 'bien', 'prenatal', 'sanidad', 'escuelas', 'comida', 'guarderías', 'saber', 'oír', 'quieren', 'meses', 'nueve', 'feto', 'obsesionados', 'conservadores', 'defensores', 'posicionamiento', 'crítica', 'ácida', 'realiza', 'humorista', 'carlin', 'mujer', 'anti', 'vida', 'pro', 'después', 'george', 'gratis', 'sí'*]

Reference document:

Pro vida es anti mujer - George Carlin El humorista George Carlin realiza una ácida crítica sobre el posicionamiento de los defensores "pro vida". <i>"Los conservadores pro vida están obsesionados con el feto hasta los nueve meses, pero después no quieren oír ni saber nada de ti. Nada, ni

Así, por ejemplo, si seleccionamos el tema "Renta Básica" podemos buscar por la palabra "renta" con los siguientes resultados:

- 7 clusters nivel 0: 388, 465, 475, 565, 795, 913, 924

De los cuales hablan de la renta básica y/o del ingreso mínimo vital las noticias de los clusters: 388, 475 y 565

- 1 cluster nivel 1: 72
que contiene los clusters de nivel 0: 388, 475, 526, 565, 640 y 795
Es decir, el cluster de nivel 1 número 72 contiene todos los clusters que corresponden a la renta básica, mas algunos otros.

Topics identifier

[Generate clusters tree](#) | [Clusters trees](#) | [Topics](#) | [Label topic with file](#)

Tree: news_v1

Search for: renta

Topic: Renta basica ▾

Select clusters:

- ☒ Level 0 - cluster 388
- ☐ Level 0 - cluster 465
- ☒ Level 0 - cluster 475
- ☒ Level 0 - cluster 565
- ☐ Level 0 - cluster 795
- ☐ Level 0 - cluster 913
- ☐ Level 0 - cluster 924
- ☐ Level 1 - cluster 72

Assign

Level 0 - Cluster 475

Terms:

['beneficiaría', '2030', 'agenda', 'trabaja', 'beneficiará', 'trabajando', 'provocada', 'mínima', 'renta', 'vital', 'derechos', 'crisis', 'anunciado', 'millones', 'ciudadanos', 'frente', 'según', 'sociales', 'coronavirus', 'gobierno', '19', 'covid', 'hacer', 'españa', 'iglesias', 'pablo', 'vicepresidente']

Reference document:

El Gobierno trabaja en una renta mínima vital que beneficiará a más de 5 millones de ciudadanos frente al Covid-19 El Gobierno está trabajando en una renta mínima vital de la que se beneficiaría más de 5 millones de ciudadanos en España para hacer frente a la crisis provocada por el coronavirus COVID-19, según ha anunciado el vicepresidente de Derechos Sociales y Agenda 2030, Pablo Iglesias

Level 0 - Cluster 565

Terms:

['adentra', 'distinción', 'extendería', 'moratoria', 'sumará', 'sobrevivir', 'volverá', 'básica', 'pulso', 'novedad', 'aumentar', 'plantear', 'ministros', 'económica', 'restricciones', 'alquileres', 'renta', 'empezado', 'medida', 'ayudar', 'salir', 'crisis', 'ciudadanos', 'medidas', 'crear', 'debate', 'sector', 'posibilidad', 'situación', 'último', 'consejo', 'abiertas', 'martes', 'gobierno', 'pasado', 'seguridad', 'casi', 'social', 'españa']

Reference document:

El Gobierno debate aumentar las restricciones y crear una renta básica La novedad en este último pulso es que este sector del Gobierno ha empezado a plantear la posibilidad de una renta básica

Una vez seleccionados los clusters (388, 475 y 565), si seleccionamos en el menú “Topic”, después escogemos el topic “Renta básica” y por último seleccionamos en el menú “Label topic”, aparecerá un formulario para seleccionar los documentos que pertenecen al tema “Renta básica” de entre todos los documentos que pertenecen a estos clusters.

Esto sirve para asegurarse manualmente de que todos los documentos etiquetados realmente corresponden al tema, ya que el clasificador, como se puede ver en la sección 3.4.7, no es infalible.

Topic Renta basica

[Topic documents](#) | [Topic clusters](#) | [Label documents](#)

Check the documents that belong to the topic Renta basica

Documents:

- ☐ #67 Será paulatino.. De hecho en muchas producciones, al menos a mí me lo parece, ya empiezan a aparecer actores chinos al estar financiadas desde ese país e imagino que exigirán una cuota mínima de sus nacionales. Es la forma china de hacer las cosas: sin prisa pero sin pausa. Y no dejan pasar ninguna oportunidad.
- ☐ #47 Unos son una democracia con libertad de prensa, donde puedes comprobar los datos sin peligro de ser perseguido y los otros son una dictadura durísima en la que la gente desaparece para siempre sin que nadie de la más mínima explicación y donde preguntar algo o hacer una insinuación significa poner en riesgo, no solo tu propia vida, si no la de todos los que te conocen
- ☐ #10 Una renta mínima para gente que trabaja en negro y no colabora con el sistema
- ? No gracias.

Posible trabajo futuro:

Como idea para desarrollo posterior se podría, utilizando un set de datos algo mas amplio (mas atrás en el tiempo), identificar mas ejemplos de estas temáticas y utilizar un servicio como mecánica turk o similar para obtener un set de datos debidamente etiquetado para distintas temáticas, partiendo de los documentos resultantes de hacer la búsqueda de clústers por palabras clave.

A continuación, con este set de datos se podría entrenar una red neuronal basada en BERT o similar que sería capaz de clasificar las noticias de forma automática, en lugar de la forma semi-manual que he utilizado. También se podría aplicar para clasificar los comentarios, lo que permitiría encontrar comentarios hablando sobre un tema dentro de hilos que inicialmente no corresponden a esa temática.

3.4.7 Métricas del clasificador de temas

Con el objetivo de comprobar la fiabilidad del clasificador, se generó una muestra con 50 ejemplos de noticias elegidas al azar de los clusters etiquetados como renta básica y 50 ejemplos elegidos al azar entre todas las noticias.

A continuación, esta muestra fue anotada por tres personas, etiquetando si los ejemplos trataban o no sobre el tema renta básica. Las anotaciones fueron importadas al sistema por medio de archivos csv.

Por último, se calculó el nivel de acuerdo entre los anotadores (inter-annotator agreement), así como la precisión y la exhaustividad[27] en función de la eficacia del modelo clasificando estos 100 ejemplos, mostrando los resultados por medio de la interfaz web.

Metrics

[Generate sample](#) | [Topics classification metrics](#)

Topic Classification Metrics

Topic: Renta basica

Agreement score: 0.9128371507433053

Precision: 0.7708333333333334

Recall: 1.0

3.4.7.1 Inter-Annotator Agreement

El inter-annotator agreement representa el nivel de acuerdo que hay entre los anotadores a la hora de etiquetar la muestra de ejemplos. En nuestro caso se ha calculado utilizando el algoritmo Fleiss' kappa [28][29].

El resultado obtenido, 0,91 sobre 1, indica que había bastante consenso entre los anotadores, lo que parece indicar que la clasificación es bastante objetiva y la muestra fiable.

3.4.7.2 Precisión y exahustividad

La precisión indica cuantos de los ejemplos que han sido etiquetados positivos realmente lo son. En nuestro caso, cuantos de los ejemplos que han sido clasificados dentro de uno de los clusters que hemos identificado como renta básica en la plataforma son considerados renta básica por los anotadores. Hemos obtenido una precisión del 77%, lo que quiere decir que la mayoría de las noticias en esos clusters son renta básica, pero no todas.

La exahustividad (recall) representa cuantos de los ejemplos positivos han sido clasificados como tal. En nuestro caso, cuantos de los ejemplos que los anotadores consideran renta básica, han sido clasificados dentro de uno de los clusters que hemos identificado como renta básica en la plataforma. Hemos obtenido una exahustividad del 100%, lo que quiere decir que no ha habido ningún ejemplo de renta básica clasificado fuera de estos clústers.

Nota: Hemos considerado dentro de la categoría renta básica las noticias que hablan de un ingreso mínimo vital y una renta mínima vital que, aunque no son lo mismo, forman parte del mismo debate.

3.5 Identificación de argumentos

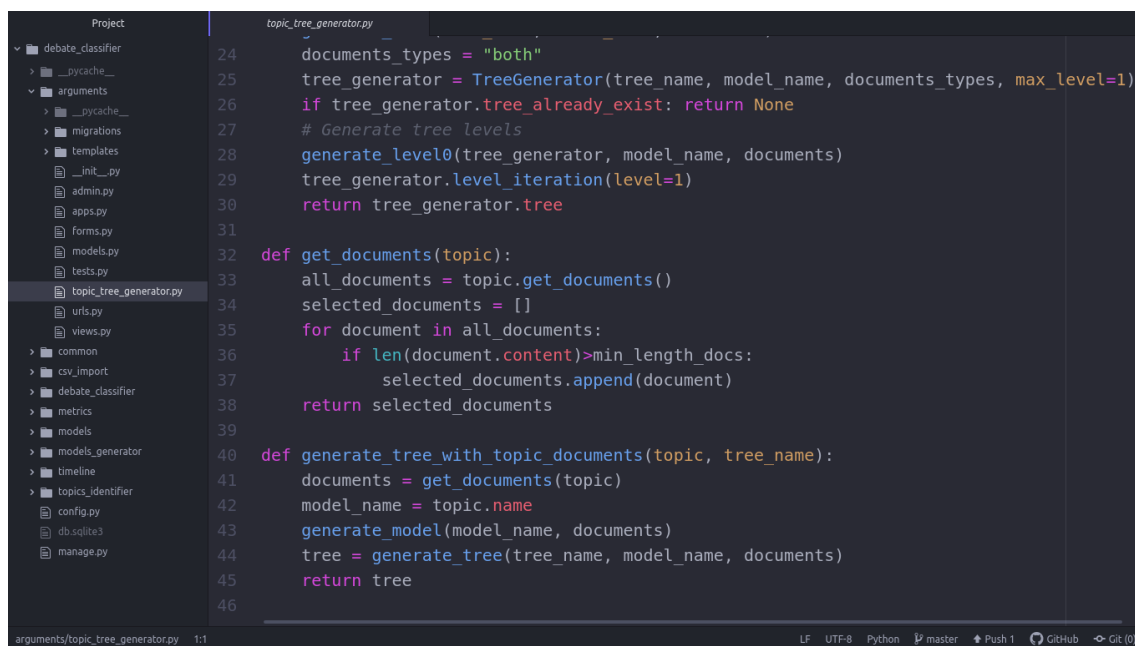
La siguiente tarea es identificar los argumentos a favor y en contra de de las diferentes posturas dentro de un tema. En este caso, los argumentos a favor y en contra de la renta básica.

Para esto partimos del trabajo ya hecho en el punto 3.4.6 Búsqueda de temáticas con los clusters donde se han etiquetado los documentos correspondientes al tema.

A continuación, generamos un nuevo arbol de clusters con todas las noticias y comentarios de esa temática.

El primer intento dio resultados poco satisfactorios, con algunos clusters sin terminos de referencia y una clasificación en general poco útil.

Sin embargo en un segundo intento, descartando los comentarios de menos de 25 caracteres, dio un resultado bastante mejor. Además, los documentos descartados resultaban muy pocos en comparación al total de documentos.



```
Project
├── debate_classifier
│   ├── __pycache__
│   ├── arguments
│   │   ├── __pycache__
│   │   ├── migrations
│   │   ├── templates
│   │   ├── __init__.py
│   │   ├── admin.py
│   │   ├── apps.py
│   │   ├── forms.py
│   │   ├── models.py
│   │   ├── tests.py
│   │   └── topic_tree_generator.py
│   ├── urls.py
│   └── views.py
├── common
├── csv_import
├── debate_classifier
├── metrics
├── models
├── models_generator
├── timeline
├── topics_identifier
├── config.py
├── db.sqlite3
└── manage.py

topic_tree_generator.py
24 documents_types = "both"
25 tree_generator = TreeGenerator(tree_name, model_name, documents_types, max_level=1)
26 if tree_generator.tree_already_exist: return None
27 # Generate tree levels
28 generate_level0(tree_generator, model_name, documents)
29 tree_generator.level_iteration(level=1)
30 return tree_generator.tree
31
32 def get_documents(topic):
33     all_documents = topic.get_documents()
34     selected_documents = []
35     for document in all_documents:
36         if len(document.content) > min_length_docs:
37             selected_documents.append(document)
38     return selected_documents
39
40 def generate_tree_with_topic_documents(topic, tree_name):
41     documents = get_documents(topic)
42     model_name = topic.name
43     generate_model(model_name, documents)
44     tree = generate_tree(tree_name, model_name, documents)
45     return tree
46
```

Arbol generado:

Document types: news and comments

Terms:

[‘mala’, ‘quiere’, ‘nadie’, ‘encuentran’, ‘vela’, ‘sociedad’, ‘trabajo’]

Reference document:

#1 una sociedad que no vela por los que no encuentran trabajo porque nadie les quiere es una mala sociedad

Level 1 - Cluster 1

Terms:

['trabajadores', 'parte', 'exigencias', 'reducción', 'puesto', '2trabajadores', 'gt', 'lt', '1puesto', 'menos', 'pagando', '17', 'trabajo']

Reference document:

#17 Pagando menos. 1 puesto de trabajo &&&&&>>>>2trabajadores /puesto
=reducción de exigencias por parte de los trabajadores

Level 1 - Cluster 2

Al igual que con la detección de temas se podría:

- Utilizar esta clasificación como punto de partida para etiquetar manualmente ejemplos de argumentarios basándonos en la clasificación de los clusters.
- Con estos ejemplos entrenar una red neuronal pre-entrenada para reconocimiento de lenguaje natural basada en BERT[12] o similar.
- Luego usar la red neuronal para identificar los comentarios que contienen estos argumentarios, aplicando la red sobre todos los comentarios disponibles. Esto tendría además la ventaja de que podría identificar cuando alguien está hablando de esto en otros hilos.

El objetivo de este paso es inferir qué usuarios (anonimizados) han hablado de los temas identificados y expresado estos argumentos.

3.6.1 Usuarios y temas

Para obtener la información sobre qué usuarios han participado en qué temas habría que, a partir de un listado de todos los hilos de un tema, obtener los

usuarios autores de los documentos contenidos en esos hilos, junto con la fecha de publicación.

A partir de esta información se podrían hacer estadísticas de participación en temas a lo largo del tiempo, sacando información como por ejemplo qué temas son más populares, si participan muchos usuarios con poca intensidad o pocos usuarios con muchos comentarios, si hay usuarios que tienen a participar en el mismo tipo de temas y la evolución de todo esto a lo largo del tiempo.

3.6.2 Usuarios y argumentos

Del mismo modo, una vez etiquetados los argumentos, se puede extraer qué usuarios han expresado qué argumentos y la evolución de la popularidad de los mismos a lo largo del tiempo.

También se puede extrapolar cómo los usuarios son influidos por la exposición a los argumentos considerando que un usuario que participa en un hilo donde se ha expresado un argumento, está expuesto al mismo.

Desafortunadamente disponemos de información incompleta para medir el grado de exposición de cada usuario a una idea. Primero porque no disponemos de la información sobre los usuarios que han leído cada hilo, sólo podemos saber si han participado en él publicando un comentario.

Segundo porque desconocemos si estos usuarios han estado expuestos a esas ideas en otras plataformas, a través de los medios de comunicación o interactuando en persona con otros individuos.

No obstante, la información que podemos obtener, aunque incompleta, es muy interesante.

3.7 Métodos y modelos de análisis de influencia

En este apartado se exponen un resumen del estado del arte con respecto a las distintas herramientas para medir y enmarcar el análisis de influencia, dejando la implementación de este último paso como desarrollo futuro.

Para confeccionar este listado he tomado como referencia los surveys “Influence analysis in social networks - A survey”[30] y “Social Influence Analysis: Models, Methods, and Evaluation”[31], ambos de 2018.

Además de este listado, en el anexo hay un resumen sobre la historia del análisis de influencia en el apartado 6.1 y un resumen de conceptos teóricos en el apartado 6.2.

3.7.1 Métodos

“Social Influence Analysis: Models, Methods, and Evaluation” [31] presenta un listado de métodos utilizados para resolver problemas de análisis de influencia en redes sociales, a los que denomina SIA methods.

Los divide en cuatro categorías:

- Métodos de **maximización de influencia** (influence maximization): se centran en encontrar el grupo de miembros con mas influencia en la red social.

- Métodos de **minimización de influencia** (influence minimization): intentan minimizar la propagación de contenido considerado negativo o dañino.
- Métodos de **flujo de influencia** (flow of influence): como su nombre indica, estudian el flujo de influencia de ideas.
- Métodos de **influencia individual** (individual influence): analizan la influencia de un usuario sobre otros usuarios o en la red social al completo.

Para este trabajo los métodos más interesantes de analizar son los dos últimos, por lo que a continuación expongo un listado de los artículos centrados en estos métodos.

Flujo de influencia:

Maximizing information or influence spread using flow authority model in social networks. [32]

Content-centric flow mining for influence analysis in social streams [33]

STRIP: Stream learning of influence probabilities.[34]

Collective influence of multiple spreaders evaluated by tracing real information flow in large-scale social networks. [35]

Influence of topology in information flow in social networks [36]

Influencia individual:

Finding influencers in networks using social capital. [37]

TOSI: a trust-oriented social influence evaluation method in contextual social networks. [38]

Credit distribution and influence maximization in online social networks using node features. [39]

3.7.2 Modelos

Los modelos se pueden dividir en tres tipos: modelos de difusión, de evaluación y de análisis.

3.7.2.1 Modelos de difusión

Hay cuatro categorías de modelos de difusión:

- Linear threshold model: propuesto por Granovetter and Schelling en 1978 [40].
- Independent cascade model: propuesto por Goldenberg et al. en 2001 [41].
- Epidemic model [42]. Estos a su vez se clasifican en tres categorías: modelos determinísticos, estocásticos y espacio-temporales.

3.7.2.2 Modelos de evaluación

Los modelos de evaluación se dividen en tres:

- Basados en temáticas (topic based)
- No basados en temáticas (topic oblivious)
- Pairwise: basados en los vínculos e interacciones entre los usuarios.

De estos tres, los que más nos interesan son los **basados en temáticas**:

- Enfocados al marketing viral

Measuring user influence on twitter: the million follower fallacy [43]

An influence strength measurement via time-aware probabilistic generative model for microblogs [44]

Discovering latent influence in online social activities via shared cascade Poisson processes [45]

Modelling influence in a social network: metrics and evaluation. [46]

Scalable topic-specific influence analysis on microblogs [47]

Identifying influential users by their postings in social networks [48]

- Enfocados a redes sociales

Social influence analysis in large-scale networks[49]

A two-level topic model towards knowledge discovery from citation networks[50]

Unsupervised prediction of citation influences [51]

A language-based approach to measuring scholarly impact. [52]

Confluence: conformity influence in large social networks [53]

An author-reader influence model for detecting topic-based influencers in social media [54]

Role-aware conformity influence modeling and analysis in social networks [55]

- Otros

Social influence analysis and application on multimedia sharing websites [56]

Who should share what?: Item-level social influence prediction for users and postsranking [57]

Identifying the influential bloggers in a community [58]

Quantifying sentiment and influence in blogspaces [59]

Mining topic-level opinion influence in microblog [60]
Personalized influential topic search via social network summarization[61]
Identifying high betweenness centrality nodes in large social networks[62]

3.7.2.3 Modelos de análisis

Los modelos de análisis de influencia se pueden dividir en dos categorías: modelos microscópicos y macroscópicos.

- **Modelos microscópicos:** Se enfocan en las interacciones entre individuos. Los ejemplos más característicos son el independent cascade model [63-65] y el linear threshold model [63][66].

- **Modelos macroscópicos:**

Epidemics and rumors [67]
Dynamics of rumor spreading in complex networks [68]
Theory of rumor spreading in complex social networks [69]
Influence of network structure on rumor propagation [70]
A new rumor propagation model on SNS structure [71]
Rumor spreading model with trust mechanism in complex social networks [72]
Rumor spreading model considering hesitating mechanism in complex social networks [73]
An information propagation model considering incomplete reading behavior in microblog [74]
The analysis of an SEIR rumor propagation model on heterogeneous network [75]
SIHR rumor spreading model in social networks [76]
Rumor spreading model considering forgetting and remembering mechanisms in inhomogeneous networks [77]

Posible desarrollo futuro

Un ejemplo de implementación sería, usando el modelo expuesto en el libro Social Physics[8], estudiar la correlación entre la exposición a argumentos a favor y en contra de una postura y el comportamiento posterior.

El libro expone que la probabilidad de que un individuo adquiera un nuevo hábito o adopte una opinión sobre un tema, aumenta con el número de veces que está expuesto a otros individuos que considera sus iguales (peers) y que tengan ese hábito o expresen esa opinión. Esto, partiendo de la implementación propuesta en el apartado 3.6.2 *Usuarios y argumentos*, se traduce en validar que la probabilidad de que un usuario de un foro exprese

una idea aumenta cuando este usuario ha estado expuesto a esa misma idea expresada por otros usuarios dentro del foro.

4 Resultados y conclusiones

Durante el desarrollo de este trabajo se ha logrado construir una herramienta capaz de identificar temas de debate dentro de una comunidad online.

Las métricas obtenidas para el tema escogido como ejemplo, la renta básica, han sido de una exhaustividad del 100%, lo que quiere decir que no ha habido ningún ejemplo de renta básica clasificado fuera de los clústers identificados como tal.

Metrics

[Generate sample](#) | [Topics classification metrics](#)

Topic Classification Metrics

Topic: Renta basica

Agreement score: 0.9128371507433053

Precision: 0.7708333333333334

Recall: 1.0

La precisión del 77% no es satisfactoria, pero se ha conseguido compensar esto mediante un etiquetado semi-manual, partiendo de la clasificación proporcionada por los clusters.

Además, se ha conseguido hacer un primer intento de clasificación de argumentos dentro de los documentos de la temática renta básica, con resultados prometedores.

Como idea para desarrollo posterior se podría, partiendo de los documentos resultantes de hacer la búsqueda de clústers por palabras clave, identificar mas ejemplos de temas y argumentos utilizando un servicio de crowdsourcing para obtener un set de datos debidamente etiquetado.

Este set de datos se podría utilizar para hacer el entrenamiento (fine-tuning) de una red neuronal pre-entrenada para tareas de reconocimiento de lenguaje natural, con el fin de usarla para identificar los distintos debates y flujo de ideas dentro de la comunidad objeto de estudio.

5 Bibliografía

- [1] Menéame. Wikipedia. <https://es.wikipedia.org/wiki/Menéame>
- [2] Documentación de Scikit-learn. AffinityPropagation. <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html#sklearn.cluster.AffinityPropagation>
- [3] Renta básica universal. Wikipedia. https://es.wikipedia.org/wiki/Renta_b%C3%A1sica_universal
- [4] Inter-rater reliability. Wikipedia. https://en.wikipedia.org/wiki/Inter-rater_reliability
- [5] Xinyuan Xu, Terhi Nurmikko-Fuller, and Bernardo Pereira Nunes. 2018. Tweets, Death, and Rock 'n' Roll: Social Media Mourning on Twitter and Sina Weibo. In Proceedings of 10th ACM Conference on Web Science (WebSci'18). ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3201064.3201079>
- [6] Sushil Bikhchandani, David Hirshleifer, Ivo Welch, and others. 2005. Information Cascades and Observational Learning. Charles A. Dice Center for Research in Financial Economics, Fisher College of Business, Ohio State University.
- [7] Boğaçhan Çelen and Shachar Kariv. 2004. Distinguishing informational cascades from herd behavior in the laboratory. American Economic Review 94, 3 (2004), 484–498.
- [8] “Social Physics. How Social Networks Can Make Us Smarter”. By Alex Pentland. MIT Press. 2015
- [9] Django <https://www.djangoproject.com/>
- [10] Scikit-learn <https://scikit-learn.org/>
- [11] Debate Classifier en github. https://github.com/Adavideo/debate_classifier
- [12] What is BERT <https://github.com/google-research/bert#what-is-bert>
- [13] Amazon Mechanical Turk <https://www.mturk.com/>
- [14] BERT for dummies <https://towardsdatascience.com/bert-for-dummies-step-by-step-tutorial-fb90890ffe03>
- [15] Documentación de Scikit-learn. Clustering. <https://scikit-learn.org/stable/modules/clustering.html#clustering>
- [16] Documentación de Scikit-learn. KMeans. <https://scikit-learn.org/stable/modules/clustering.html#k-means>

- [17] Applying Machine Learning to classify an unsupervised text document <https://towardsdatascience.com/applying-machine-learning-to-classify-an-unsupervised-text-document-e7bb6265f52>
- [18] Versión del clasificador con KMeans. https://github.com/Adavideo/debate_classifier/blob/a4da06a81f5e80e32c2a717585d99892bbad3164/topics/topics_identifier/data_classifier.py
- [19] Documentación de Scikit-learn. Clustering text documents using k-means. https://scikit-learn.org/stable/auto_examples/text/plot_document_clustering.html
- [20] Documentación de Scikit-learn. Datasets. <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets>
- [21] Documentación de Scikit-learn. 7.5.4. Loading from external datasets <https://scikit-learn.org/stable/datasets/index.html?highlight=sklearn%20datasets%20base%20bunch#loading-other-datasets>
- [22] Documentación de Scikit-learn. `sklearn.datasets.load_files` https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_files.html
- [23] Documentación de Scikit-learn. Feature extraction. https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction
- [24] Documentación de Scikit-learn. Stop words. https://scikit-learn.org/stable/modules/feature_extraction.html#stop-words
- [25] Stop words para el lenguaje Español. Github. Usuario Alir3z4. <https://github.com/Alir3z4/stop-words/blob/master/spanish.txt>
- [26] Documentación de Scikit-learn. Model persistence https://scikit-learn.org/stable/modules/model_persistence.html
- [27] Precisión y exhaustividad. Wikipedia. https://es.wikipedia.org/wiki/Precisi%C3%B3n_y_exhaustividad
- [28] Fleiss' kappa. Wikipedia. https://en.wikipedia.org/wiki/Fleiss%27_kappa
- [29] Inter-Annotator Agreement (IAA) by Louis de Bruijn. <https://towardsdatascience.com/inter-annotator-agreement-2f46c6d37bf3>
- [30] Peng, Sancheng & Zhou, Yongmei & Cao, Lihong & Yu, Shui & Niu, Jianwei & Jia, Weijia. (2018). Influence analysis in social networks: A survey. *Journal of Network and Computer Applications*. 106. 10.1016/j.jnca.2018.01.005.
- [31] Li, Kan & Zhang, Lin & Huang, Heyan. (2018). Social Influence Analysis: Models, Methods, and Evaluation. *Engineering*. 4. 10.1016/j.eng.2018.02.004.

- [32] Faisan, Mohamed & Surampudi, Durga. (2014). Maximizing Information or Influence Spread Using Flow Authority Model in Social Networks. 233-238. 10.1007/978-3-319-04483-5_24.
- [33] Subbian, Karthik & Aggarwal, Charu & Srivastava, Jaideep. (2013). Content-centric flow mining for influence analysis in social streams. 841-846. 10.1145/2505515.2505626.
- [34] Kutzkov, Konstantin & Bifet, Albert & Bonchi, Francesco & Gionis, Aristides. (2013). STRIP: stream learning of influence probabilities. 10.1145/2487575.2487657.
- [35] Teng, Xian & Pei, Sen & Morone, Flaviano & Makse, Hernán. (2016). Collective Influence of Multiple Spreaders Evaluated by Tracing Real Information Flow in Large-Scale Social Networks. Scientific Reports. 6. 36043. 10.1038/srep36043.
- [36] Chintakunta, Harish & Gentimis, Thanos. (2016). Influence of topology in information flow in social networks. 67-71. 10.1109/ACSSC.2016.7868995.
- [37] Subbian K, Sharma D, Wen Z, Srivastava J. Finding influencers in networks using social capital.
- [38] Liu, Guanfeng & Zhu, Feng & Zheng, Kai & Liu, An & Li, Zhixu & Zhao, Lei & Zhou, Xiaofang. (2016). TOSI: A trust-oriented social influence evaluation method in contextual social networks. Neurocomputing. 210. 10.1016/j.neucom.2015.11.129.
- [39] Deng, Xiaoheng & Pan, Yan & Wu, You & Gui, Jingsong. (2015). Credit Distribution and influence maximization in online social networks using node features. 2093-2100. 10.1109/FSKD.2015.7382274.
- [40] (Granovetter, 1978) Granovetter, M., 1978. Threshold models of collective behavior. Am. J. Sociol. 83 (6), 1420–1443. [http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref38](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref38)
- [41] Goldenberg et al. (2001) Goldenberg, J., Libai, B., Muller, E., 2001. Talk of the network: a complex systems look at the underlying process of word-of-mouth. Market. Lett. 12 (3), 211–223. [http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref35](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref35)
- [42] Peng, S., Yu, S., Yang, A., 2014. Smartphone malware and its propagation modeling: a survey. IEEE Commun. Surv. Tutor. 16 (2), 925–941. [http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref85](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref85)
- [43] Cha et al. (2010) Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.P., 2010. Measuring user influence on twitter: the million follower fallacy. In: Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM 2010), Washington, DC, USA, pp. 10–17. [http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref15](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref15)

- [44] Ding et al. (2013b) Ding, Z., Jia, Y., Zhou, B., Zhang, J., Han, Y., Yu, C., 2013. An influence strength measurement via time-aware probabilistic generative model for microblogs. In: Proceedings of the 15th Asia-Pacific Web Conference, Sydney, Australia, pp. 373–383.
[http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref25](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref25)
- [45] Iwata et al. (2013) Iwata, T., Shah, A., Ghahramani, Z., 2013. Discovering latent influence in online social activities via shared cascade Poisson processes. In: KDD 2013, Chicago, Illinois, USA, pp. 266–274.
[http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref50](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref50)
- [46] Hajian and White (2011) Hajian, B., White, T., 2011. Modelling influence in a social network: metrics and evaluation. In: 2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing, pp. 497–500. DOI: 10.1109/PASSAT/SocialCom.2011.118
- [47] Bi et al. (2014) Bi, B., Tian, Y., Sismanis, Y., Balmin, A., Cho, J., 2014. Scalable topic-specific influence analysis on microblogs. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM 2014), New York, USA, pp. 513–522. DOI: 10.1145/2556195.2556229
- [48] Sun and Ng (2012) Sun, B., Ng, V.T., 2012. Identifying influential users by their postings in social networks. In: MSM 2012, Milwaukee, Wisconsin, pp. 1–8. [http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref102](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref102)
- [49] Tang et al. (2009) Tang, J., Sun, J., Wang, C., Yang, Z., 2009. Social influence analysis in large-scale networks. In: Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009), New York, NY, USA, pp. 807–816.
<https://doi.org/10.1145/1557019.1557108>
- [50] Guo et al. (2014) Guo, Z., Zhang, Z., Zhu, S., Chi, Y., Gong, Y., 2014. A two-level topic model towards knowledge discovery from citation networks. *IEEE Trans. Knowl. Data Eng.* 26 (4), 780–794.
[http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref40](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref40)
- [51] Dietz et al. (2007) Dietz, L., Bickel, S., Scheffer, T., 2007. Unsupervised prediction of citation influences. In: Proceedings of the 24th International Conference on Machine Learning. ICML, Corvallis, Oregon, USA. DOI: 10.1145/1273496.1273526
- [52] Gerrish and Blei (2010) Gerrish, S.M., Blei, D.M., 2010. A language-based approach to measuring scholarly impact. In: Proceedings of the 26th International Conference on Machine Learning (ICML 2010), Haifa, Israel, pp. 375–382. [http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref34](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref34)
- [53] Tang et al. (2013) Tang, J., Wu, S., Sun, J., 2013. Confluence: conformity influence in large social networks. In: Proceeding of the 19th ACM SIGKDD

- International Conference on Knowledge Discovery and Data Mining (KDD 2013), Chicago, Illinois, USA, pp. 347–355. DOI: 10.1145/2487575.2487691
- [54] Herzig et al. (2014) Herzig, J., Mass, Y., Roitman, H., 2014. An author-reader influence model for detecting topic-based influencers in social media. In: Proceedings of the 25th ACM Conference on Hypertext and Social Media (HT 2014), Santiago, Chile, pp. 46–55.
<https://doi.org/10.1145/2631775.2631804>
- [55] Zhang et al. (2014) Zhang, J., Tang, J., Zhuang, H., Leung, C.W., Li, J., 2014. Role-aware conformity influence modeling and analysis in social networks. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 958–965.
- [56] Sang and Xu (2013) Sang, J., Xu, C., 2013. Social influence analysis and application on multimedia sharing websites, *ACM Transactions on Multimedia Computing. Commun. Appl.* 9 (1s), 1–24. [http://refhub.elsevier.com/S1084-8045\(18\)30019-5/sref93](http://refhub.elsevier.com/S1084-8045(18)30019-5/sref93)
- [57] (Cui et al., 2011) Cui, P., Wang, F., Liu, S., Ou, M., Yang, S., Sun, L., 2011. Who should share what?: Item-level social influence prediction for users and postsranking. In: Proceedings of the 34th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2011), Beijing, China, pp. 185–194. DOI: 10.1145/2009916.2009945
- [58] (Agarwal et al., 2008) Agarwal, N., Liu, H., Tang, L., Yu, P.S., 2008. Identifying the influential bloggers in a community. In: Proceedings of the 1th ACM International Conference on Web Search and Data Mining (WSDM 2008). PaloAlto, California, USA, pp. 207–217.
<https://doi.org/10.1145/1341531.1341559>
- [59] (Hui and Gregory, 2010) Hui, P., Gregory, M., 2010. Quantifying sentiment and influence in blogspaces. In: Proceedings of the the 1st Workshop on Social Media Analytics, New York, USA, pp. 53–61.
<https://doi.org/10.1145/1964858.1964866>
- [60] Li et al., (2012) Li, D., Shuai, X., Sun, G., Yang, J., Ding, Y., Luo, Z., 2012. Mining topic-level opinion influence in microblog. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012), Maui, USA, pp. 1562–1566. DOI: 10.1145/2396761.2398473
- [61] Li et al., (2016) Li, J., Liu, C., Yu, J.X., Chen, Y., Sellis, T., Culpepper, J.S., 2016. Personalized influential topic search via social network summarization. *IEEE Trans. Knowl. Data Eng.* 28 (7), 1820–1834. doi: 10.1109/TKDE.2016.2542804
- [62] (Kourtellis et al) Kourtellis, N., Alahakoon, T., Simha, R. et al. Identifying high betweenness centrality nodes in large social networks. *Soc. Netw. Anal. Min.* 3, 899–914 (2013). doi: 10.1007/s13278-012-0076-6

- [63] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2003 Aug 24–27; Washington, DC, USA; 2003. p. 137–46.
- [64] Leskovec J, Mcglohon M, Faloutsos C, Glance NS, Hurst M. Patterns of cascading behavior in large blog graphs. In: Proceedings of the 2007 SIAM International Conference on Data Mining; 2007 Apr 26–28; Minneapolis, MN, USA; 2007.
- [65] Gruhl D, Guha R, Liben-Nowell D, Tomkins A. Information diffusion through blogspace. In: Proceedings of the 13th International Conference on World Wide Web; 2004 May 17–20; New York, NY, USA; 2004. p. 491–501.
- [66] Granovetter M. Threshold models of collective behavior. *Am J Sociol* 1978;83 (6):1420–43. [http://refhub.elsevier.com/S2095-8099\(17\)30805-6/h0130](http://refhub.elsevier.com/S2095-8099(17)30805-6/h0130)
- [67] Daley DJ, Kendall DG. Epidemics and rumors. *Nature* 1964;204(4963):1118.
- [68] Moreno Y, Nekovee M, Pacheco AF. Dynamics of rumor spreading in complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 2004;69(6 Pt 2):066130.
- [69] Nekovee M, Moreno Y, Bianconi G, Marsili M. Theory of rumor spreading in complex social networks. *Phys A* 2007;374(1):457–70.
- [70] Zhou J, Liu Z, Li B. Influence of network structure on rumor propagation. *Phys Lett A* 2007;368(6):458–63.
- [71] Wang H, Deng L, Xie F, Xu H, Han J. A new rumor propagation model on SNS structure. In: Proceedings of the 2012 IEEE International Conference on Granular Computing; 2012 Aug 11–13; Hangzhou, China; 2012. p. 499–503.
- [72] Wang Y, Yang X, Han Y, Wang X. Rumor spreading model with trust mechanism in complex social networks. *Commun Theor Phys* 2013;59(4):510–6.
- [73] Xia L, Jiang G, Song B, Song Y. Rumor spreading model considering hesitating mechanism in complex social networks. *Phys A* 2015;437:295–303.
- [74] Su Q, Huang J, Zhao X. An information propagation model considering incomplete reading behavior in microblog. *Phys A* 2015;419:55–63.
- [75] Liu Q, Li T, Sun M. The analysis of an SEIR rumor propagation model on heterogeneous network. *Phys A* 2017;469:372–80.
- [76] Zhao L, Wang J, Chen Y, Wang Q, Cheng J, Cui H. SIHR rumor spreading model in social networks. *Phys A* 2012;391(7):2444–53.

[77] Zhao L, Qiu X, Wang X, Wang J. Rumor spreading model considering forgetting and remembering mechanisms in inhomogeneous networks. *Phys A* 2013;392 (4):987–94.

6 Anexo: Análisis de Influencia

¿Cómo se forman las opiniones individuales y colectivas? ¿Hasta qué punto influye nuestro entorno a la hora de definirnos por una postura en sobre un tema? El análisis de influencia es un tema ampliamente estudiado desde muchas perspectivas distintas, que ha cobrado una especial relevancia con la aparición de las redes sociales y dispositivos móviles, que permiten medir en tiempo real cómo nos relacionamos y la información que compartimos.

6.1 Breve historia del análisis de influencia

El análisis de influencia se puede dividir en dos grandes bloques: el estudio de cómo el entorno social afecta al individuo, entendiendo este entorno como algo externo que le afecta y actúa sobre él, y el estudio de las dinámicas grupales, entendiendo estos fenómenos como algo colectivo. Ambos enfoques son útiles y se complementan.

Tenemos ejemplos bastante interesantes de trabajos siguiendo ambos enfoques a partir de mediados del siglo pasado.

6.1.1 El individuo y el entorno social

Los primeros trabajos en este campo datan de la década de los 50. En el artículo “Opinions and Social Pressure”[1], publicado en 1955, Solomon Asch se preguntaba “How and to what extent do social forces constrain people’s opinions and attitudes?”. Pocos años después Herbert C. Kelman publicaba “Compliance, identification, and internalization: Three processes of attitude change”[2].

Uno de los trabajos más impactantes en este campo es “Los Peligros de la Obediencia”[3] de Stanley Milgram, publicado en la década de los 70. Este artículo que resumía los resultados de una serie de experimentos conocidos como “El experimento de Milgram”[4]. Estos experimentos causaron tanto impacto que se sigue hablando de ellos aún en la actualidad, ya que sus resultados ponen en tela de juicio creencias fuertemente arraigadas sobre cómo actuamos y tomamos decisiones los seres humanos, mostrando como la presión de nuestro entorno social puede llegar a hacernos actuar en contra de nuestros propios principios.

6.1.2 La influencia como fenómeno colectivo

6.1.2.1 Las celebridades y la teoría del flujo de dos pasos

En las décadas de 1940 y 1950 un grupo de investigadores (Paul Lazarsfeld, Elihu Katz y colegas)[5] popularizaron una teoría sobre la formación de la opinión pública denominada “two-step flow of communication”.

Como se explica en la introducción del artículo “Influentials, Networks, and Public Opinion Formation”[6], esta teoría pretendía reconciliar el rol de la influencia de los medios de comunicación con la importancia de los individuos altamente influyentes:

“[...] reconcile the role of media influence with the growing realization that, in a variety of decision-making scenarios, ranging from political to personal, individuals may be influenced more by exposure to each other than to the media.”[6]

Resulta interesante que ya en los años 50 se encontraban en un escenario similar al nuestro, con una innovación tecnológica (en su caso la televisión) que estaba teniendo un gran impacto en como se transmitía la información y cómo nos relacionábamos con ella.

De acuerdo con esta teoría, una pequeña minoría de líderes de opinión actuaban como intermediarios entre los medios de comunicación de masas y la mayoría de la sociedad. Según esto, la información fluía desde los medios, a través de los líderes de opinión hacia sus seguidores.

Según Gitlin [7] A finales de los 60, esta teoría se consideraba una de las más influyentes y el paradigma dominante en sociología de los medios.

En el artículo anteriormente mencionado “Influentials, Networks, and Public Opinion Formation”[6] de 2007, Watts y Sheridan cuestionan el modelo two-step flow argumentando que aunque el proceso ha sido de sobra documentado, no existen pruebas de que la influencia individual sea realmente relevante:

although the dual concepts of personal influence and opinion leadership have been extensively documented, it is nevertheless unclear exactly how, or even if, the influentials of the two-step flow are responsible for diffusion processes, technology adoption, or other processes of social change. [6]

6.1.2.2 Social Physics

En el libro Social Physics (2015) [8], donde exponen los resultados de varios experimentos llevados a cabo en el MIT [9], argumentan que la probabilidad de que un individuo adquiriera un nuevo hábito o adopte una opinión sobre un tema, aumenta con el número de veces que están expuestos a otros individuos que consideran sus iguales (peers) y que tengan ese hábito o expresen esa opinión.

“The search for new ideas and information, like the formation of new habits, appears driven primarily by social exposure. [...] It wasn’t just direct interactions that mattered; it was the amount of all exposure to the behavior of people who gained weight, including both direct interaction and indirect observation.”[8]

6.2 Conceptos teóricos

Para entender mejor el marco teórico del análisis de influencia, son convenientes algunas definiciones.

Los conceptos mostrados a continuación han sido extraídos mayoritariamente del libro Social Physics [8], con algunas referencias otras fuentes que se indican en cada caso.

6.2.1 Conceptos básicos

6.2.1.1 Aprendizaje social (social learning)

El proceso del aprendizaje social implica que si hay mucha interacción entre alguien exhibiendo un comportamiento (el modelo a seguir o “role model”) y otra persona, y esta última es susceptible, es bastante probable que esta nueva idea sea adoptada por ella y cambie su comportamiento.

6.2.1.2 Susceptibilidad

La susceptibilidad depende de varios factores incluyendo el nivel de confianza entre las dos personas, si la persona exhibiendo el comportamiento es lo suficientemente similar a la otra como para que el nuevo comportamiento sea útil para ella, y la consistencia entre la nueva idea y los comportamientos previamente aprendidos.

Según los resultados del experimento expuesto en el artículo “Social Influence and the Collective Dynamics of Opinion Formation”[10], los individuos son más fácilmente influenciados por otros individuos con opiniones similares a las suyas, mientras que las opiniones que son muy lejanas a su opinión actual, las descartan pensando que la otra persona está equivocada.

6.2.1.3 Confianza

Cuando la gente ve a otros adoptando estrategias similares a las suyas, a menudo adquieren más confianza en las suyas y es más probable que inviertan mas en esa estrategia. Las decisiones de las personas son una mezcla de la información propia y la información social (aprendida de las personas de nuestro entorno). Cuando tenemos poca información, tendemos a apoyarnos mas en la información social.

En el experimento antes mencionado[10] cuando la confianza de un individuo era igual o inferior al otro individuo, el primero tendía a adaptar su opinión.

Se podían distinguir tres zonas en el mapa de influencia, dependiendo del grado de confianza de cada individuo y de la similitud entre las opiniones de ambos:

- **Zona de confirmación:** cuando ambos individuos tienen una opinión similar, ambos tienden a mantener su opinión, independientemente de cual sea el nivel de confianza del otro. Además tienden a aumentar su nivel de confianza en su opinión.
- **Zona de influencia:** La influencia tiende a ser mucho más fuerte en niveles intermedios de disenso. En esta zona, la mayor parte de las personas tienden a llegar a un compromiso entre ambas opiniones cuando la otra persona tiene el mismo nivel de confianza o mayor y la otra opinión difiere lo suficiente como para motivar una revisión pero no tanto como para ser descartada. Los niveles de confianza de ambos participantes tienden a permanecer igual después de esta interacción.
- **Opiniones muy alejadas:** Cuando la distancia entre las opiniones de ambos individuos es muy grande, la fuerza de la influencia social va disminuyendo progresivamente, probablemente porque asumen que la otra persona está equivocada. Aún así, la otra opinión no es totalmente ignorada. La mayoría de las personas tienden a llegar a una solución de compromiso cuando la otra persona tiene mucho mas nivel de confianza en la suya. Incluso la gente que tiene un alto nivel de confianza tiende a empezar a dudar de su opinión.

6.2.1.4 Flujo de ideas (idea flow)

El flujo de ideas es la difusión de las mismas entre un conjunto de personas, ya sea por medio de comportamiento observado o transmitido verbalmente.

La forma de medir el flujo de ideas es mediante la probabilidad de que una persona cambie su comportamiento cuando una nueva idea aparece en su entorno social.

La probabilidad de que el comportamiento de una persona cambie cuando una nueva idea aparece en su entorno social no depende sólo de interacciones directas, si no de la cantidad de exposición a ese nuevo comportamiento tanto con interacciones directas como indirectas (comportamiento observado en nuestro entorno). De hecho, en algunos casos, la probabilidad de copiar un nuevo comportamiento depende mas de lo que la gente realmente hace (comportamiento observado) que lo que dicen que hacen. El flujo de ideas depende del aprendizaje social, por lo que nuestro comportamiento puede predecirse en cierto grado en función a la exposición a comportamientos de otras personas.

La exposición al comportamiento de nuestro entorno es el factor más importante para la transmisión de ideas. Esto es debido a que aprender de otros individuos

es mucho mas eficiente que depender únicamente de nuestras propias experiencias. Modelos matemáticos de aprendizaje en entornos complejos sugieren que la mejor estrategia para aprender en ellos es invertir el 90% de nuestro tiempo en la exploración, encontrando y copiando a otros a los que parece irles bien [11] y un 10% en experimentar y pensar por cuenta propia[12].

6.2.2 El individuo y el entorno

6.2.2.1 Cambios en el comportamiento

Para la transmisión de conocimientos objetivos, una sola exposición a ese conocimiento por parte de un individuo de confianza suele ser suficiente.

Sin embargo los cambios de comportamiento requieren estar expuesto en varias ocasiones dentro de un periodo corto de tiempo. Suele ser necesario que un individuo esté expuesto en varias ocasiones a ver que un nuevo comportamiento genera una ventaja (como por ejemplo aprobación social) para que sea probable que dicho individuo adopte el nuevo comportamiento.

Cuando todos nuestros semejantes a nuestro alrededor están exhibiendo un comportamiento similar, como por ejemplo ganar o perder peso, la uniformidad de los ejemplos a nuestro alrededor tiende a influirnos fuertemente, tanto en decisiones conscientes como en hábitos inconscientes. [13]. La influencia social puede fomentar tanto comportamientos buenos como malos.

“We all sail in a stream of ideas, [...] examples and stories of the peers who surround us; exposure to this stream shapes our habits and beliefs. [...] We can resist the flow if we try, [...] but most of our behavior is shaped by the ideas we are exposed to. The idea flow within these streams binds us together into a sort of collective intelligence [...] comprised of the shared learning of our peers.” - Social Physics [8]

Sin embargo, los individuos pueden elegir cambiar su entorno para cambiar los comportamientos a los que están expuestos.

6.2.2.2 Razonamiento individual y flujo de ideas

Para entender el rol del razonamiento individual en el flujo de ideas necesitamos analizar como los hábitos y las creencias son creados.

Kahneman y Simon [14][15] definieron un modelo de funcionamiento de la mente humana con dos mecanismos de pensamiento complementarios:

- El rápido y automático, intuitivo y en gran parte inconsciente.
- El pensamiento lento y consciente, que usa el pensamiento racional, combinado con nuestras creencias y conocimientos para llegar a nuevas conclusiones.

El pensamiento rápido parece jugar un rol importante en crear sociedades sanas. Hay estudios psicológicos que muestran que las decisiones rápidas, tomadas en el momento, tienden a ser mas altruistas y cooperativas que las decisiones pensadas con calma.

La información de la que disponemos para saber lo que queremos y valoramos, además de cómo actuar para obtenerlo, está constantemente cambiando y evolucionando en función de nuestras interacciones con otros individuos. Nuestros deseos y preferencias están fuertemente basadas en lo que nuestra comunidad está de acuerdo en que es valioso, más que en nuestras reflexiones racionales o nuestros impulsos biológicos. [16]

“we are now coming to realize that human behavior is determined as much by social context as by rational thinking or individual desires. Rationality [...] means that we know what we want and act to get it. [...] research shows that both people’s desires and their decisions about how to act are often [...] dominated by social network effects. [...] we have biases and cognitive limitations that prevent us from realizing full rationality.”
Social Physics [8]

6.2.2.3 Dinámicas grupales

Casi todas las decisiones que afectan al grupo en su conjunto son tomadas en situaciones sociales [17].

Compartiendo ideas podemos tomar mejores decisiones que con la mejor de las decisiones individuales. [18-20] La estrategia de aprendizaje de grupo de ir retroalimentando la mejor idea existente produce un efecto de sabiduría de las masas (“wisdom of the crowds”) que funciona incluso en grupos pequeños [21].

Cada comunidad tiene su propio flujo de ideas que permite a los miembros incorporar innovaciones de otros individuos del grupo pudiendo llegar incluso a crear una cultura propia.

6.2.2.4 Sentido común

El sentido común que forma parte de una comunidad viene del flujo de ideas. Con el paso del tiempo, una comunidad con miembros que interactúan activamente unos con otros crea un grupo con valores y hábitos compartidos.

Nuestros ancestros entendían que nuestra cultura y los hábitos de nuestra sociedad son contratos sociales, y ambos dependen sobre todo del aprendizaje social.

La mayoría de nuestras decisiones son fuertemente influenciadas por el sentido común, los hábitos y las creencias que tenemos en común con nuestros semejantes, y estos hábitos en común toman forma a través de nuestras interacciones con otras personas.

Aprendemos el sentido común de forma casi automática, observando y copiando los comportamiento más comunes entre nuestros semejantes. [22]

6.2.3 Innovación y diversidad

6.2.3.1 Diversidad

Parece que la clave para recolectar ideas que llevan a grandes decisiones es aprender de los éxitos y fracasos de otros, a la vez que asegurarse de que las oportunidades de aprendizaje son lo suficientemente diversas.

Cuando el flujo de ideas en la comunidad incorpora una corriente constante de ideas externas, los individuos de la comunidad toman mejores decisiones de las que podrían tomar individualmente.

“we can think of each stream of ideas as a swarm or collective intelligence, flowing through time, with all the humans in it learning from each other’s experiences in order to jointly discover patterns of preferences and habits of action that best suit the surrounding physical and social environment.”
- Social Physics

6.2.3.2 Cámaras de eco

El mecanismo de aprendizaje social sólo mejora cuando los participantes cuentan con información individual distinta. Cuando las fuentes de información externa son demasiado similares el pensamiento grupal se vuelve un verdadero peligro. Es fácil pensar que todo el mundo a nuestro alrededor ha llegado a las mismas conclusiones de forma independiente y volvernos más confiados en estas estrategias o creencias de lo que deberíamos.

Cuando un grupo altamente cohesionado se retroalimenta reforzando sus creencias, se considera que el grupo está atrapado en una cámara de eco.

El éxito depende en gran medida de la calidad de nuestra exploración de ideas y depende de la diversidad e independencia de nuestra información y nuestras fuentes.

En un entorno lleno de cámaras de eco, es mucho más difícil tomar buenas decisiones. Esto sugiere que necesitamos prestar mucha mas atencion a las fuentes de las que vienen nuestra información e ideas.

6.2.3.3 Innovación

La gente más creativa y perspicaz invierte una gran cantidad de tiempo en conocer gente nueva, con diferentes puntos de vista e ideas.

Filtran y mejoran las ideas que han descubierto mas recientemente compartiéndolas con otras personas con las que se encuentran. La diversidad

de puntos de vista y experiencias es un factor muy importante en el proceso de creación de ideas innovadoras.

“One reason human culture grows [...] we occasionally choose to row against the flow of ideas surround us and dip into another stream. [...] by crossing what sociologist Ron Burt called the ‘structural holes’ [23] within the fabric of society, we can create innovation.” - Social Physics.

Esta teoría se basa en la idea de que la homogeneidad de la información, las nuevas ideas y el comportamiento es en general mayor en un sólo grupo de personas comparado con dos grupos de personas. Un individuo que actúa como mediador entre dos o mas grupos que están muy conectados entre sí, puede obtener importantes ventajas.

6.3 Referencias

- [1] Asch SE (1955) Opinions and social pressure. Scientific American 193: 33–35.
- [2] Kelman HC. Compliance, identification, and internalization: Three processes of attitude change. Journal of Conflict Resolution. 1958;2 (1) :51-60.
- [3] The Perils of Obedience. Stanley Milgram (1974).
<http://www.physics.utah.edu/~detar/phys4910/readings/ethics/PerilsofObedience.html>
- [4] El experimento de Milgram.
https://es.wikipedia.org/wiki/Experimento_de_Milgram
- [5] Katz, Elihu and Paul Felix Lazarsfeld (1955), Personal Influence; the Part Played by People in the Flow of Mass Communications, Glencoe, IL: Free Press.
- [6] Watts, Duncan & Dodds, Peter. (2007). Influentials, Networks, and Public Opinion Formation. Journal of Consumer Research. 34. 441-458.
10.1086/518527.
- [7] Gitlin, T. (1978). Media Sociology: The Dominant Paradigm. Theory and Society. 6. 205-253.
- [8] Social Physics. How Social Networks Can Make Us Smarter. By Alex Pentland. MIT Press. 2015
- [9] Reality Commons. MIT Human Dynamics Lab.
<http://realitycommons.media.mit.edu/>
- [10] Moussaïd, Mehdi & Kämmer, Juliane & Analytis, Pantelis & Neth, Hansjörg. (2013). Social Influence and the Collective Dynamics of Opinion Formation. PloS one. 8. e78433. 10.1371/journal.pone.0078433.
- [11] Rendell et al. 2010
- [12] Lazer and Friedman 2007; Grinton et al. 2010; Anghel et al. 2004; Yamamoto et al. 2013; Sueur et al. 2012; Farrell 2011.
- [13] Stanley Milgram work on social conformity.
https://en.wikipedia.org/wiki/Milgram_experiment
- [14] Simon 1978; Kahneman 2002.
- [15] Kahneman, D. (2011) Thinking, Fast and Slow
- [16] Haidt 2010.
- [17] Buchanan 2007.
- [18] Collective Intelligence – James Surowiecki 2004
- [19] Dall et al. 2005
- [20] Lorenz et al. 2011
- [21] King et al. 2012.
- [22] Hassin et al. 2005
- [23] Structural holes. Wikipedia.
https://en.wikipedia.org/wiki/Structural_holes