Facultad de Ingeniería

Universidad de la república

Estudio de menciones a personalidades públicas basado en clustering aplicado a tweets

proyecto de grado

Federico Kauffman Piñeiro
Fabián Andrés Larrañaga Fagián

Tutores

Dra. Ing. Aiala Rosá
Dr. Ing. Guillermo Moncecchi

URUGUAY, Montevideo. Julio 2017

Agradecimientos

Agradecemos a todos los que nos dieron apoyo y aliento durante la elaboración de este proyecto. Entre ellas a nuestras familias, amigos y novias por la comprensión y aguante. A WyeWorks por permitirnos llevar a cabo nuestras reuniones internas en la oficina así como las facilidades de estudio. También agradecemos a Aiala y Guillermo, nuestros tutores, por la confianza para trabajar en un área no abordada hasta el momento y por la buena energía y disposición.

Resumen

Las redes sociales tal y como las conocemos hoy en día son un recurso inagotable para la comunicación e información. Las personas pasamos una gran parte del día revisando nuestras redes, compartiendo información y comunicándonos a través de ellas. Sin embargo, el análisis de esta información es relativamente escaso.

En este proyecto, para hacer uso de esta información se propone analizar menciones a personas en la red social Twitter con el objetivo de conocer los temas más relevantes relacionados a políticos y deportistas uruguayos. Para ello se construye un sistema de recolección de tweets con el cual se genera un corpus de aproximadamente 420 mil ejemplares. Posteriormente se aplican y evalúan dos algoritmos de clustering: K-Means y DBSCAN. Además se experimenta con algunas técnicas de expansión de tweets logrando superar los resultados obtenidos en la línea base en la mayoría de los experimentos realizados.

Palabras clave: análisis de tweets, aprendizaje no supervisado, cluster, clustering, anotación, corpus, tweets, validación.

Índice general

Capítulo 1

Introducción

El ser humano es un animal inherentemente social [Lieberman, 2013, 1–19]. Su capacidad de expresión es sin dudas uno de los aspectos más sobresalientes de la especie humana. En la "era de la información" esta destreza ha sido magnificada por la facilidad de comunicación que las nuevas tecnologías proveen. Entre estas tecnologías se encuentran las redes sociales. Una red social en el contexto de las nuevas tecnologías es un sitio o aplicación que permite a las personas, entidades u organizaciones conectarse entre sí con el fin de compartir intereses o actividades.

Un caso particular de red social es Twitter. Este sitio permite a los usuarios descubrir qué ocurre en el mundo momento a momento, al igual que compartir intereses relacionados a múltiples temas como la música, deportes, política, entre otros. La unidad de información utilizada para compartir información se denomina tweet. Desde sus orígenes en el año 2006, Twitter, ha ganado terreno a nivel mundial. Actualmente existen alrededor de 330 millones de usuarios activos, con el estimado de un tweet por persona generado al día. Es decir, 330 millones de tweets por día [Statista, 2017].

Estas cifras han despertado mucho interés en el marco de la lingüística computacional por varios motivos. La inmediatez de la información, el tamaño de los tweets, el lenguaje utilizado y el uso de emojis son algunos de los motivos, siendo uno de los principales la cantidad de datos sobre distintos temas que se generan a diario.

En el marco de los proyectos del grupo de Procesamiento de Lenguaje Natural de la Facultad de Ingeniería — Universidad de la República — hay interés en tener en una línea de tiempo con los sucesos más importantes relacionados con una persona. Esto permitiría, por ejemplo, construir reputaciones a través de la utilización de técnicas de análisis de sentimientos. En este proyecto, yendo en esa dirección, luego de identificada una persona, se busca identificar los principales tópicos o aquellos de mayor repercusión vinculados a esa persona. Una posible forma de conocer estos sucesos es analizar conjuntos de tweets para cada personalidad a estudiar y una estrategia válida es analizar estos conjuntos aplicando diferentes técnicas de clustering.

El clustering en general se lo puede definir como el proceso de agrupar elementos de un dominio bajo un cierto criterio. Estas agrupaciones se conocen como clusters. Un cluster es un conjunto de entidades que son parecidas, y además entidades de conjuntos

distintos no son parecidas [Everitt, 2011].

En este contexto el clustering se centra en la agrupación de tweets de temas similares. Esta tarea de agrupar presenta la necesidad de contar con tweets para cada personalidad objeto de estudio y por este motivo se hace necesaria la construcción de un corpus que contenga opiniones públicas variadas.

Para poder comprender el problema de la identificación o agrupación de tópicos que se intenta resolver se considera el siguiente ejemplo extraído de Twitter:

Ejemplo 1. *Ejemplo de dificultad presente en Twitter.*

i) "Diego Godín está en la nómina de los 30 futbolistas con chance de ganar el #BalondeOro. Felicitaciones @diegogodin https://t.co/7zFkleFfh0"

ii) "Diego Godin nominado entre los 30 mejores futbolistas x la revista France Football https://t.co/FsCEyS9JuZ"

iii) "Tiene que ser tuyo @diegogodin"

A primera vista, parecería ser directo agrupar los tweets i) y ii) en un mismo conjunto. Sin embargo, ¿cuál es la decisión a tomar con el tweet iii)? Algunas personas afirmarían que el tweet iii) debe pertenecer al mismo subgrupo que i) y ii). Pero, ¿cuál es el criterio que lo determina así? ¿existe criterio? Perfectamente este tweet podría estar haciendo referencia a que "un penal", "un auto de regalo de algún patrocinio", "un cabezazo de final de partido que termine en gol", etc. tiene que ser de él, pero también podría tratarse del balón de oro. La situación no es clara, no existe contexto y no existe a priori manera de saber la forma correcta de agrupar fácilmente tweets. Por este motivo, se desea construir un algoritmo capaz de considerar y determinar la mejor manera de agrupar tweets relacionados a deportistas y políticos uruguayos. Se trata de una tarea cuya resolución no es escalable de manera manual y donde la cantidad de datos afecta el resultado directamente. Al mismo tiempo, el algoritmo debe ser capaz de reflejar múltiples características de un solo tweet así como interpolarlas con otras características presentes en los demás tweets.

El objetivo de este proyecto es estudiar, diseñar e implementar diferentes formas para clusterizar tweets, construir un corpus específico sobre políticos y deportistas uruguayos, investigar y utilizar medidas de evaluación del clustering y poder representar los resultados de manera gráfica y amigable.

El resto de este documento se encuentra dividido en cinco capítulos. El segundo capítulo se focaliza principalmente en detallar el marco teórico necesario así como presentar diferentes formas de clustering y ámbitos donde es aplicable. El capítulo tres presenta el proceso de construcción y la descripción del corpus. El cuarto capítulo, abarca el proceso de clustering así como la construcción y ajuste de las diferentes variantes. En los últimos dos capítulos se tiene el análisis de resultados y las conclusiones de este proyecto así como el trabajo a futuro.

Existe un glosario el cual se puede consultar en cualquier momento para obtener información de algunos términos pocos usuales o desconocidos por el lector.

Capítulo 2

Marco Teórico

El presente capítulo intenta conceptualizar y definir el marco teórico necesario para abordar el problema de clustering a resolver. También se da a conocer el estado-del-arte del clustering, presentando aspectos teórico-técnicos y los algoritmos más conocidos al día de la fecha. Se dedica una subsección a los diferentes modelos y técnicas utilizadas en la validación de clustering. Por último, se da a conocer el estado del arte del clustering de Tweets, donde se presentan además algunos trabajos previos en el área.

Según se expresa en [Pinker, 1997, 12]: *"Un ser inteligente no puede tratar cada objeto como una entidad única y diferente a todo el resto de objetos del universo. Debe poder categorizarlos en ciertos grupos y así aplicar conocimientos de objetos similares a aquellos objetos sin categorizar"*. Esto entre otras cosas sugiere una suerte de agrupación necesaria en la vida real para poder diferenciar conjuntos de cosas. En el contexto de este proyecto, y en el área de aprendizaje automático y Procesamiento del Lenguaje Natural, a esta tarea de agrupar se la conoce como clustering de datos.

Este problema ha sido objeto de estudio por más de 20 años en las áreas de minería de datos y aprendizaje automático debido a las numerosas aplicaciones para aprendizaje, resumen y segmentación de datos que posee. Además, [Gan, 2011] la establece como una de las seis tareas esenciales en la explotación de información.

2.1. Clustering

El clustering es la agrupación de objetos basada en los datos que estos poseen y sus relaciones. El objetivo de esta técnica es poder obtener conjuntos de objetos que logren resaltar y capturar la estructura natural de los datos. Por este motivo es muchas veces utilizado como punto de partida para resolver otros problemas en ámbitos de la psicología, biología, estadística y sistemas de información en áreas como recuperación de datos, aprendizaje automático y minería de datos.

Para lograr su principal objetivo, agrupar datos, se busca identificar conjuntos de objetos tales que aquellos pertenecientes a una misma agrupación sean parecidos o estén relacionados de cierta manera. Esto no es todo, sino que además, objetos en conjuntos distintos sean lo más diferente posible.

Si bien el tema de agrupar por similitud parece ser algo simple y que resulta natural para el ser humano, en muchas ocasiones la noción de cluster no es fácil de establecer. Para poder evaluar similitud, es necesario definir una medida de distancia. Esta medida debe lograr condensar la información y permitir obtener conjuntos que tengan sentido y utilidad. Y así, poder atacar el problema que se busca resolver.

Consideremos el siguiente ejemplo presentado por [Tan, 2006]

Sea un conjunto de 20 puntos (véase figura 2.1), el cual se quiere particionar sin ningún criterio específico. A priori es difícil determinar el número de particiones, podrían ser dos, cuatro, seis, o cualquier cantidad entre 1 y 20. Determinar esta cantidad con cierto sentido generalmente depende de los datos y los resultados que se esperan en caso que se tenga conocimiento.

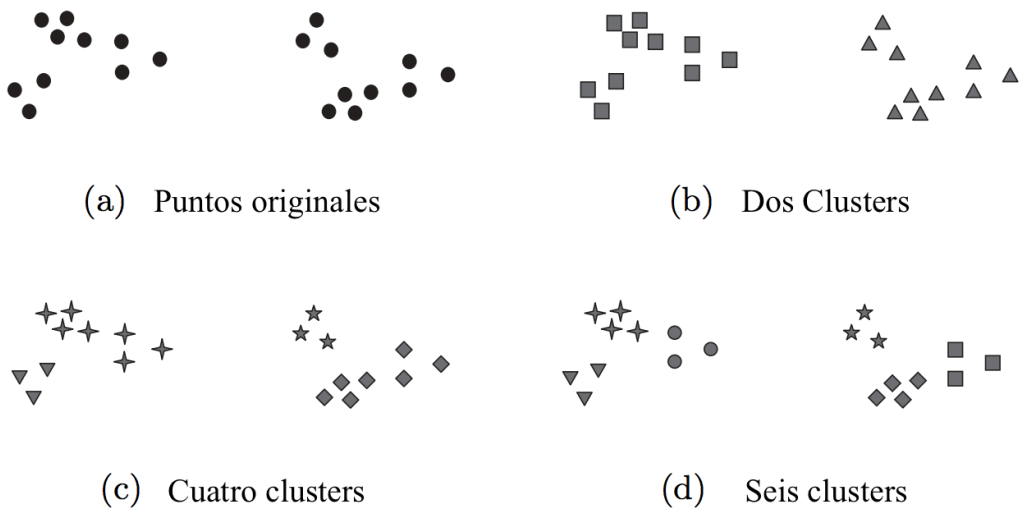


Figura 2.1: Diferentes formas de clusterizar el mismo conjunto de puntos.

El proceso de clustering también se puede considerar un proceso de clasificación (asignar etiquetas predefinidas a objetos de un corpus), donde las etiquetas son generadas solo en base a los datos internos. A diferencia de la clasificación supervisada, en clustering, las etiquetas suelen ser desconocidas y es por este motivo que el clustering puede considerarse un proceso de clasificación no supervisada.

Los clusters se pueden definir de muchas formas, sin embargo todas intentan finalmente expresar el mismo concepto. A continuación se establecen algunas de las posibles definiciones [Jain, 1988] extraídas originalmente de [Everitt, 2011].

1) Un cluster es un conjunto de entidades que son parecidas, y además entidades de clusters distintos no son parecidas.

2) Un cluster es una aglomeración de puntos en el espacio de datos tal que la distancia

entre dos puntos cualesquiera en el cluster es menor que la distancia entre un punto del cluster y uno cualquiera afuera de este.

3) Los clusters se pueden describir como regiones conexas del espacio multidimensional conteniendo una alta densidad de puntos, separados de otras regiones por zonas de baja densidad.

Las definiciones dos y tres dependen ambas de una noción de distancia. Este es uno de los problemas más importantes a resolver en el clustering, como se representan los datos y cómo se miden las distancias entre entidades. Se puede decir entonces que en definitiva el concepto clave del clustering se centra en tener un buen cálculo de la distancia y similitud entre objetos.

A continuación se presentan algunos ejemplos con el fin de aclarar el concepto y mostrar algunos de sus usos en las distintas áreas.

Ejemplo 1: Se considera una compañía de venta de artículos para el hogar que está interesada en saber las cantidades de venta de sus distintos productos. De esta forma poder conocer aquellos que están siendo rentables y aquellos que no. El problema es optimizable si se logra agrupar aquellos productos que están dando menos ganancias y los que están dando más. En este sentido, lo único que debería preocupar a la empresa sería analizar el conjunto que genera menos ingresos y no la totalidad de sus productos. En conclusión se ganaría tiempo, reduciría el esfuerzo monetario/humano y facilitaría ampliamente el proceso de mining al resolver un problema localizado.

Casos similares se dan en el ámbito financiero y de marketing [Tudor, 2013, 176–194]. Se plantea la asistencia a vendedores para distinguir grupos diferentes de clientes, y así, conociendo sus preferencias como compradores permitir desarrollar programas de ventas de manera focalizada y precisa.

Ejemplo 2: Usando texto para generar hipótesis sobre enfermedades. La empresa Hearst (1999) plantea:

Durante más de una década, Don Swanson ha argumentado elocuentemente por qué es plausible esperar que nueva información sea derivable de las colecciones de texto: los expertos sólo pueden leer un pequeño subconjunto de lo que se publica en sus campos y con frecuencia desconocen los desarrollos en campos relacionados. Por lo tanto, debería ser posible encontrar vínculos útiles entre la información en literaturas relacionadas. Swanson ha demostrado cómo las cadenas de causas de enfermedades dentro de la literatura médica pueden conducir a hipótesis de causas de enfermedades raras, algunas de las cuales han recibido evidencia experimental como apoyo [Swanson, 1987, 228–233]; [Smalheiser, 1994, 1–9].

Existen otros autores y bibliografías que justifican el uso de clustering en el ámbito de la medicina de diferentes formas, tal es el caso de [Kalyani, 2012, 1–4] que plantea su utilización para la partición de datos médicos así como también en la toma de decisiones.

Los ejemplos anteriores son nada más que algunas de las posibles aplicaciones del clustering. Si bien cada uno de ellos presenta un común denominador, la agrupación de información, los ámbitos y datos a analizar son de distintas índoles. Esto conduce a la necesidad de distintas técnicas y modelos de clustering. A continuación se detallan algunos de los modelos de clustering más utilizados al día de la fecha.

2.1.1. Métodos basados en particiones

Este modelo de clustering se basa en el principio de particiones de los conjuntos. El clustering basado en particiones se caracteriza por seguir una lógica de reubicación de los elementos, donde cada uno de ellos es movido de una agrupación a otra según corresponda. Para hacer posible esta mecánica es necesario considerar un estado base que represente el particionamiento inicial y a partir de éste comenzar a construir los clusters [Äyrämö, 2006].

La siguiente sección presenta algunos de los tipos de clustering basados en particiones.

Algoritmos basados en minimización de errores

La idea de fondo de estos algoritmos consiste en minimizar un cierto criterio de error utilizado para medir la distancia entre cada elemento a clusterizar y su valor más representativo. Para esto, es necesario disponer de un representante por cada partición existente. La elección de los representantes y las medidas de distancia son fundamentales en estos métodos y son quienes regulan el comportamiento del algoritmo específico que se utilice. En cada iteración del algoritmo se asignan elementos a aquellos conjuntos donde se minimice el error respecto al representante [Rokach, 2005, (321–352)].

El criterio más conocido y utilizado para estos modelos es la suma de errores al cuadrado (SSE en inglés) que mide la distancia euclídea total entre una instancia y su representante.

$$SSE = \sum_{i=1}^{n} ||x_i - \bar{x}||^2 \tag{2.1}$$

Siendo $\bar{x}$ el representante de una partición.

K-Means: Un algoritmo sencillo y usado muy frecuentemente que hace uso de la Suma de Errores al Cuadrado es K-Means. Este algoritmo particiona los datos en K clusters $(C_1, C_2, ..., C_k)$, donde $C_i \ \forall i = 1, ..., k$ son los K centroides. Estos centros, se calculan como el punto medio de todas las instancias pertenecientes al cluster. K-Means posee la peculiaridad de comenzar con un conjunto de centroides C_k predefinidos, ya sea aleatoriamente o de acuerdo a alguna heurística determinada. Una vez finalizado, el conjunto resultante de clusters será de la misma aridad K elegida. Básicamente lo que ocurre en cada iteración del algoritmo es que cada instancia o elemento es asignado al cluster más cercano. Esto a partir de calcular la distancia entre los centroides y la instancia dada.

Posterior a asignar todos los elementos a algún cluster, se prosigue a re-calcular los centroides como el promedio o elemento medio de las instancias del cluster [Rokach, 2005, (321–352)].

Calculándose de la siguiente manera:

$$\mu_k = \frac{1}{N_k} \sum_{q=1}^{N_k} x_q \tag{2.2}$$

donde N_k es la cantidad de instancias pertenecientes al cluster K.

La siguiente figura, (fig. 2.2) [Manning, 2009] muestra gráficamente como funciona K-Means.

Se muestran los 5 grandes estados por los que debe pasar un algoritmo basado en particiones. El primero de ellos consiste en seleccionar y definir los K centroides ($K = 2$ en este caso particular). El siguiente paso representa la primer iteración del algoritmo y en esencia es lo que se estará repitiendo hasta que termine el clustering. Este paso implica asociar las instancias a su cluster más cercano. El tercer estado de K-Means consiste en reasignar los nuevos centroides del cluster, para ello se consideran todos las instancias del cluster y se consigue el elemento que minimiza la distancia a todos los elementos. La repetición de este paso para cada cluster genera K nuevos centros. Una vez obtenido los nuevos centroides, es momento de re-asociar las instancias al cluster cuyo centroide sea el más cercano (paso 2), y así sucesivamente.

El algoritmo debe disponer de condiciones de parada. Lo más habitual es considerar una cantidad máxima de iteraciones y al mismo tiempo un entorno de variación de los clusters que permita identificar que las instancias de un cluster se mantienen invariadas en un cierto período de tiempo. El resultado final para el ejemplo, se muestra en la cuarta sub-imagen luego de 9 iteraciones. La última imagen muestra el movimiento de los centroides a lo largo de toda la ejecución.

Existen otros dos algoritmos muy similares a K-Means, ellos son: K-Medians y K-Medoids.

K-Medians: A diferencia de K-Means, al momento de asignar instancias a un cluster, calcula la media en cada dimensión haciendo que este método sea menos sensible a ruido y elementos aislados. Al igual que con K-Means el representante del cluster (centroide) se calcula en base a los datos y no tiene porque ser un elemento particular.

K-Medoids: En este caso se selecciona el centroide considerando una instancia. Este método es un poco más lento que los anteriores porque debe no solo calcular el "medio" del cluster sino que además debe utilizar la instancia más cercana a este para designarla como centroide.

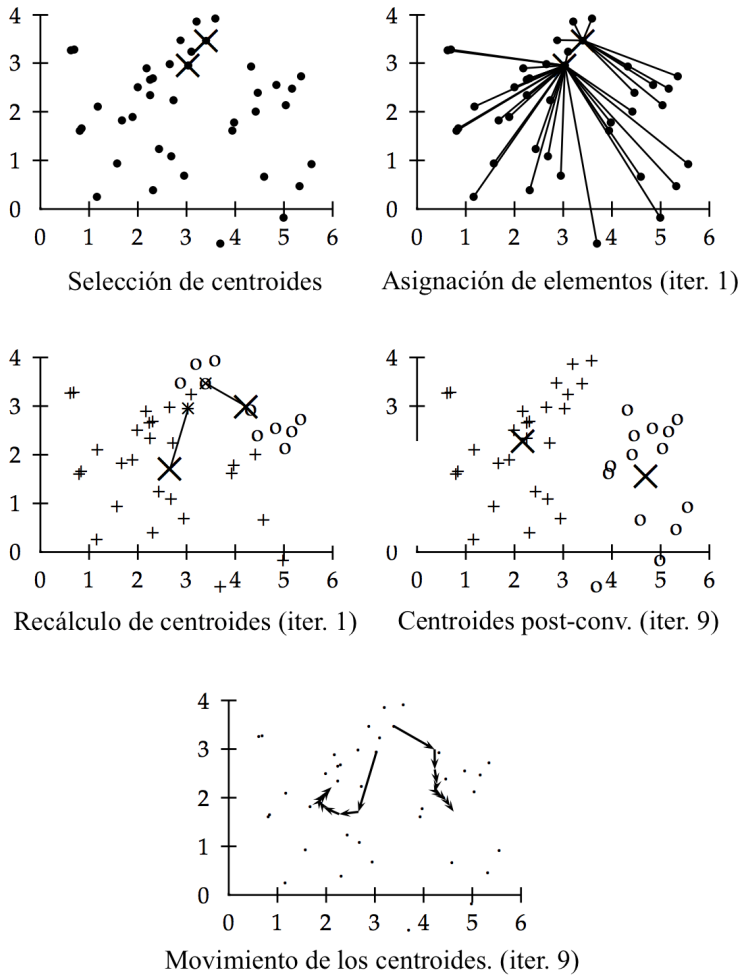


Figura 2.2: Ejemplo de K-Means para $K = 2$ en $\mathbb{R}^2$. La posición de los centroides converge tras nueve iteraciones.

2.1.2. Métodos basados en densidad o grillas

Los métodos basados en densidad suponen que los elementos pertenecientes a cada cluster parten de una distribución de probabilidad específica [Banfield, 1993, 803–821]. Se supone que la distribución general de los datos es una mezcla de varias distribuciones. El objetivo de estos métodos es caracterizar las distribuciones que generan los puntos de los clusters a través de sus parámetros. Este tipo de algoritmos está diseñado para descubrir clusters de forma arbitraria que no necesariamente sean convexos, a diferencia de los métodos anteriores. Es decir, dados $x_i, x_j \in C_k$, no necesariamente $\alpha * x_i + (1-\alpha) * x_j \in C_k$

El funcionamiento de este tipo de algoritmos consiste en incrementar el tamaño de un cluster dado, siempre y cuando la densidad (número de instancias) en el "vecindario" supere algún umbral. Esto quiere decir que dentro de un cierto perímetro dado, se tiene

que superar una cantidad mínima de elementos.

La mayor parte de trabajo en el área se ha basado en el supuesto de que las densidades son Gaussianas (en caso de elementos numéricos) o Multinomiales (en caso de datos nominales - datos basados en el etiquetado o codificación de información en categorías -). A continuación se detalla el algoritmo más popular de este modelo de clustering.

Density Based Spatial Clustering of Applications with Noise (DBSCAN)

Este algoritmo [Ester, 1996] es uno de los primeros que emplea este enfoque basado en densidad. Comienza seleccionando un punto t arbitrario, en caso de que t sea un punto central, se empieza a construir un cluster alrededor de él. De esta forma se trata de descubrir componentes denso-conectadas. En caso de no ser un elemento central, se elige otro elemento del conjunto de datos. Para comprender esta idea, se define elemento central, borde (o frontera) y elemento ruido:

- Puntos centrales son aquellos tales que en su vecindad de radio Eps, poseen una cantidad de puntos mayor o igual que un umbral $MinPts$ especificado.
- Un elemento borde o frontera tiene menos puntos que $MinPts$ en su vecindad, pero pertenece a la vecindad de un punto central.
- Un punto ruido es aquel que no es ni central ni borde.

La figura 2.3, [González, 2010] ilustra cada uno de estos conceptos considerando $4 \leqq MinPts \leqq 6$. Esto significa que A es un elemento central, B es un elemento borde y C es un elemento ruido.

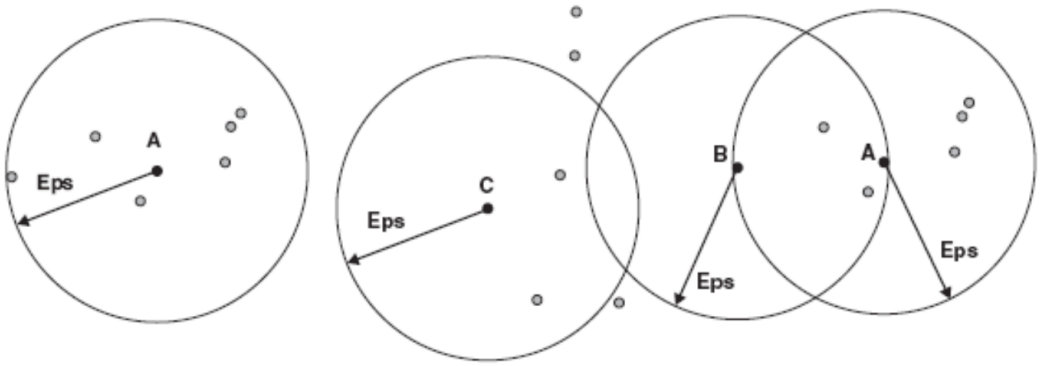


Figura 2.3: Definiciones de punto central, borde y ruido.

En este algoritmo lo primero que se realiza es etiquetar cada elemento bajo alguna de las tres categorías anteriores. Luego, se prosigue a eliminar aquellos elementos ruido. Con los elementos restantes, es decir aquellos que no son elementos ruido, para cada elemento central que no haya sido asignado a un cluster, se crea un nuevo cluster y todos

aquellos puntos denso-conectados se los agrega. De esta misma forma, se asocian aquellos elementos borde al cluster del elemento centro más cercano.

Dado que estos métodos se basan en separar entre zonas densas y no densas, donde cada zona densa representa un cluster, permiten descubrir clusters como los de la figura 2.4:

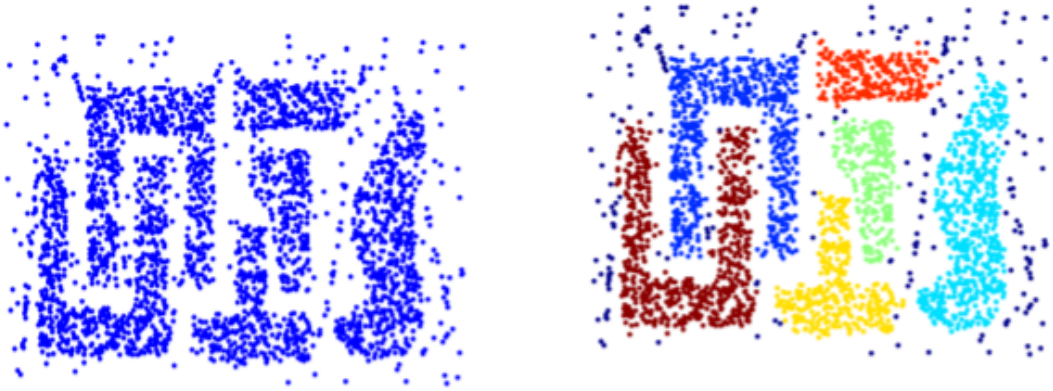


Figura 2.4: Clustering mediante DBSCAN. Seis clusters.

Métodos basados en grillas

Los métodos basados grillas son una subclase de los métodos basados en densidad. Difieren principalmente en que el cálculo de los clusters se basa en el espacio de datos que rodea a cada elemento a agrupar y no en los elementos en sí. Un algoritmo clásico basado en el modelo de grillas consiste de cinco pasos [Grabusts P., 2002].

- Crear la estructura de grilla. Por ejemplo, particionar el espacio de datos en un número finito de celdas.
- Calcular la densidad de celda por cada celda creada.
- Ordenar las celdas de acuerdo a las densidades calculadas en el paso anterior.
- Identificar centroides.
- Relación o cruce de celdas vecinas.

Una ventaja importante de estos métodos es que, dado que exploran el espacio de datos con un alto nivel de granularidad, pueden usarse para reconstruir toda la forma de la distribución de los datos.

El principal reto de los métodos basados en densidad es que están definidos naturalmente en puntos de datos en un espacio continuo. Por lo tanto, a menudo no se pueden utilizar de manera significativa en un espacio discreto o que no sea euclidiano; a menos que se lo ajuste.

2.1.3. Métodos basados en Jerarquía

Los modelos jerárquicos se caracterizan por construir una jerarquía de agrupamientos representada a través de dendogramas (árbol) a diferentes niveles de granularidad [Duda, 2001] [Jain, 1999]. Existen dos formas de crear esta jerarquía y a eso se deben sus nombres. Las aglomerativas y las divisivas.

Las estrategias jerárquicas aglomerativas más conocidas basadas en distancias son: Single Link (SL) [Sibson, 1972, 30–34], Average Link (AL) [hees, 1986] y Complete Link (CL) [Defays, 1977].

En cada nivel de la jerarquía, se unen los dos grupos más cercanos y la métrica entre elementos se debe generalizar a los subconjuntos de elementos. Las diferencias entre los métodos surgen debido a las diferentes maneras de definir distancia (o similitud) entre grupos.

Aglomerativos

En las técnicas aglomerativas, el dendograma se genera de "abajo hacia arriba" (bottom-up). Se parte generalmente de grupos unitarios y sucesivamente se van uniendo los clusters hasta conseguir el grupo formado por todos los elementos o hasta que algún criterio de parada se ejecute.

Divisivos

Las estrategias jerárquicas divisivas utilizan un enfoque opuesto a los aglomerativos. El enfoque es de "arriba hacia abajo" (top-down). Es decir comienzan generalmente con todos los puntos en un cluster y van dividiendo en cada nivel los grupos de acuerdo a algún criterio prefijado. La partición divisiva permite mayor flexibilidad tanto de la estructura jerárquica del árbol como del nivel de equilibrio en los diferentes grupos.

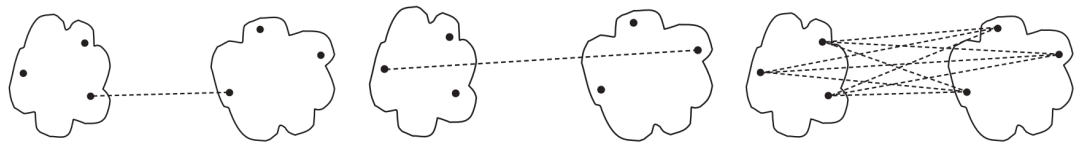
Las siguientes definiciones tomadas de [Tan, 2006] contrastan las tres formas de medir distancia en el clustering jerárquico.

Single Link (SL): En SL, la distancia entre grupos se define como la distancia entre los dos elementos más cercanos (uno de cada cluster) o, empleando terminología de grafos, el enlace más corto entre dos nodos en diferentes subconjuntos de nodos.

Average Link (AL): En el caso del algoritmo AL, la distancia entre dos grupos es el promedio de las distancias entre todos los pares de puntos (uno de cada conjunto).

Complete Link (CL): La distancia entre dos grupos en el CL es la máxima distancia entre los pares de puntos (uno de cada conjunto), es decir, en cada nivel se unirán los dos grupos cuya unión tiene diámetro mínimo o, empleando terminología de grafos, el enlace más largo entre dos nodos de diferentes subconjuntos de nodos.

La figura 2.5 describe estos conceptos gráficamente.



(a) MIN (single link.) (b) MAX (complete link.) (c) Group average.

Figura 2.5: Definiciones de proximidad entre clusters.

El siguiente ejemplo (fig. 2.6), [Cimiano, 2004, 435–439] muestra los dos enfoques distintos de clustering jerárquico.

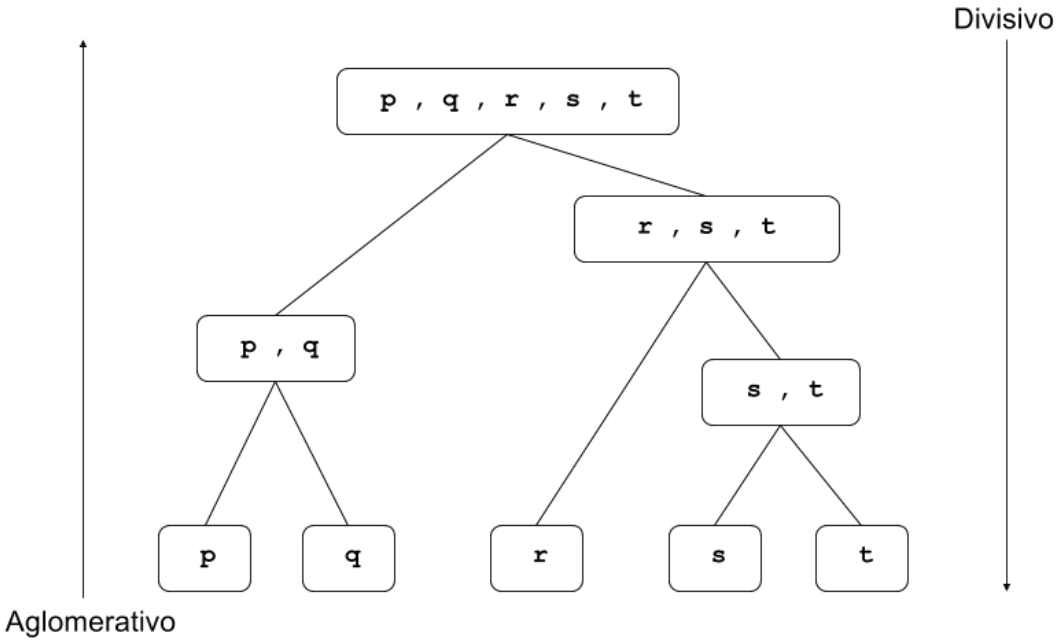


Figura 2.6: Clustering Jerárquico Aglomerativo vs Divisivo.

La figura 2.6 muestra cómo el modelo de clustering jerárquico aglomerado produce una serie de particiones de los datos $P_n, P_{n-1}, ..., P_1$. El primer P_n consta de n clusters de un solo elemento (en este caso particular, cinco clusters {p}, {q}, {r}, {s} y {t}) y donde el último P_1 es un solo cluster que contiene los n (5 en este caso) elementos existentes ({ p, q, r, s, t }).

En cada etapa particular, el método une los dos grupos que están más juntos. En la primera etapa, esto supone unir los dos objetos que están más próximos entre sí, ya que en la etapa inicial cada grupo tiene sólo un objeto.

2.1.4. Métodos basados en Modelos

Los métodos basados en modelos intentan optimizar la semejanza entre el conjunto de datos y algunos modelos matemáticos. A diferencia del clustering convencional, que identifica grupos de elementos, los métodos de clustering basados en modelos encuentran también descripciones características para cada cluster; y cada cluster representa un concepto o una clase. Cada cluster es modelado con alguna distribución matemática como por ejemplo Gaussiana o Normal. Las dos estructuras más utilizadas son los árboles de decisión y las redes neuronales.

Árboles de decisión

En el caso de los árboles de decisión, los datos se representan por un árbol jerárquico, donde cada hoja se refiere a un concepto y contiene una descripción probabilística de ese concepto. Varios algoritmos producen árboles de clasificación para representar los datos no etiquetados. Los algoritmos más conocidos son COBWEB y CLASSIT.

COBWEB: Este algoritmo asume que todos los atributos son independientes (un supuesto tal vez no demasiado correcto). Su objetivo es lograr una alta previsibilidad de los valores de las variables nominales, dado un cluster.

En la estructura de datos COBWEB, cada nodo representa un concepto. Cada concepto representa un conjunto de objetos y cada objeto que se representa se lo hace como una lista de propiedades de valor binario. Los datos asociados a cada nodo del árbol (los conceptos) son los recuentos de propiedades de enteros para los objetos en ese concepto [Sahoo, 2006, 357–366].

A modo de ejemplo, se considera la figura 2.7.

Sea el Concepto C_1 que contiene 4 objetos (los repetidos son válidos): [1 0 1], [0 1 1], [0 1 0] y [0 1 1], donde las tres propiedades en juego pueden ser [es_macho?, tiene_alas?, es_nocturno?]. Lo que se guarda en este concepto es la propiedad basada en contar las repeticiones ([1 3 3]) lo cual indica que uno de los objetos dentro de este concepto C_1 es macho, tres objetos tienen alas y tres son nocturnos.

La descripción del concepto es la categoría-probabilidad condicional (verosimilitud) de las propiedades en el nodo. Dado que el objeto es un miembro del concepto, la probabilidad de ser macho es $\frac{1}{4}$ y tanto la probabilidad de tener alas como de ser nocturno es de $\frac{3}{4}$. De esta forma la probabilidad condicional del concepto C_1 es $P(x|C_1) = (0{,}25, 0{,}75, 0{,}75)$.

La figura 2.7, muestra en su completitud 5 conceptos, C_0 es el concepto raíz que contiene los 10 objetos en el conjunto de datos y luego se tienen hijos de C_0 y así sucesivamente.

CLASSIT: Es una extensión de COBWEB para datos de valores continuos. Dado que son algoritmos que no fueron propuestos para resolver problemas de tipo minería de texto, se decide no profundizar más en el tema. Sin embargo se puede encontrar una sección muy interesante en [Sahoo, 2006, 357–366].

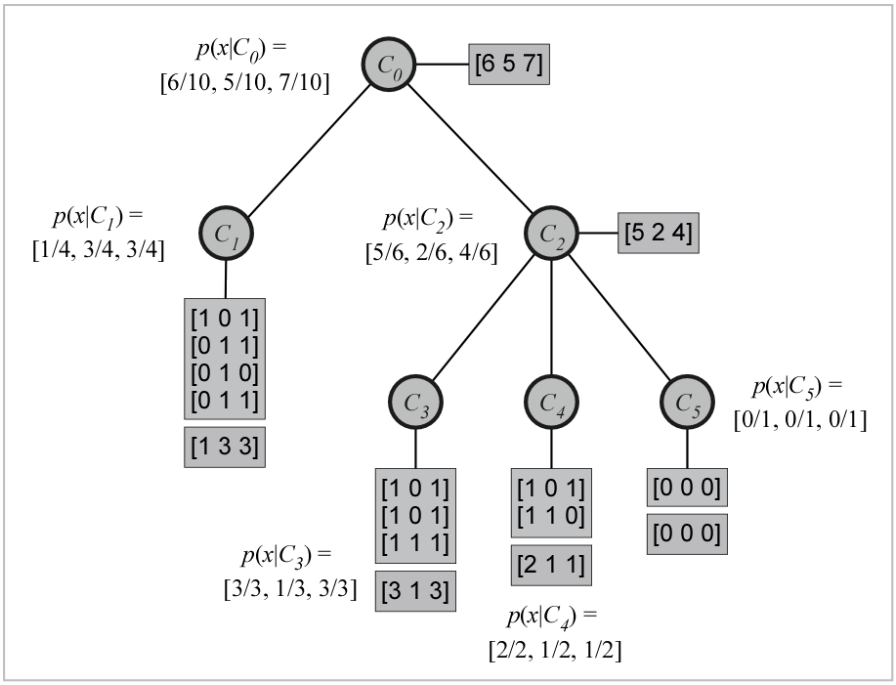


Figura 2.7: Definiciones de proximidad entre clusters. [Wikipedia, 2017]

Redes neuronales

Un algoritmo neuronal muy popular para el clustering es el Mapa Auto-organizado conocido en su versión original como Self-Organizing Map (SOM).

Self-Organizing Map (SOM): Este algoritmo construye una red de una sola capa. El proceso de aprendizaje se lleva a cabo de una manera "ganador-se lleva-todo" que se define a continuación:

- Las neuronas prototipo compiten por la instancia actual. El ganador es la neurona cuyo vector de pesos es el más cercano a la instancia presentada actualmente.
- El ganador y sus vecinos aprenden al ajustar sus pesos.

El algoritmo SOM se utiliza con éxito para la cuantificación vectorial y reconocimiento de voz. Es útil para visualizar datos de alta dimensión en el espacio 2D o 3D, al igual que con los algoritmos basados en árboles de decisión, no resulta estar en su naturaleza el trabajo en áreas de la minería de textos. Por este motivo, una vez más, se invita a consultar [Vesanto, 2000].

2.2. Evaluación de validez del clustering

Uno de los problemas más importantes en el área del clustering computacional es la evaluación de los clusters resultantes. Lograr la mejor partición de los datos y la cantidad óptima de clusters para un conjunto de datos dado es el objetivo fundamental de fondo y es muy difícil conocer la respuesta. El procedimiento de evaluar los resultados obtenidos de clusterizar un conjunto de datos se conoce como validación del cluster.

2.2.1. Técnicas de validación

Las técnicas de validación han sido objeto de estudio de varios investigadores. Existen dos grandes categorías que conforman el subgrupo de técnicas más utilizadas a nivel mundial. Se las conoce como validación externa y validación interna. Se podría decir que existe una tercer categoría, la validación relativa [Jain, 1988];[Xie, 1991];[Theodoridis, 2003]. Esta categoría surge como necesidad de lograr algo intermedio a la validación interna y la externa.

La validación de las particiones es un proceso formal que evalúa el resultado del análisis de manera cuantitativa y objetiva. Permite evaluar la corrección de las particiones de un conjunto de datos [Halkidi, 2001a]. Por otro lado [Gath, 1989] proponen tres principios para definir una partición óptima, basada en la noción de densidad. Se enuncian a continuación:

- Maximización de la separación entre las clases resultantes.
- Minimizar el volumen de cada clase resultante.
- Maximizar el conjunto de datos concentrado en la proximidad de cada centroide resultante.

Dos de las grandes preguntas que surgen de los principios anteriores, planteados por Jain (1988), cuestionan:

1) ¿Cuántos grupos hay presentes en el conjunto de datos objetivo?
2) Los grupos resultantes, ¿Son una partición válida?

Con el objetivo de lograr resolver estas interrogantes es que se plantean los tres enfoques de validación.

2.2.1.1. Validación Externa

La validación externa intenta evaluar la capacidad que tiene un algoritmo de clustering para reconocer estructuras o conjuntos de datos a partir de una agrupación hecha en forma manual (Gold Standard). Como la validación externa mide la calidad del agrupamiento conociendo información externa de antemano, es principalmente usada para

escoger un algoritmo de clustering óptimo sobre un conjunto de datos donde ya se conocen los clusters.

El índice de validación externa es quien permite valorar la similitud entre particiones. Muchos de estos índices pueden ser usados como índices internos, lo que varía es la forma en que se los aplica. A continuación se listan algunos de los índices más utilizados en cuanto a validación externa respecta.

Adjusted Rand Index (ARI): Este índice parte de la base de que un elemento es asignado a un único cluster tanto por el algoritmo de clustering como de forma manual. Por esta razón es posible recurrir a medidas de consenso entre dos resultados de realizar clustering [Yeung, 2001].

[Wagner, 2007] [Wikipedia., 2017] proponen que dado un conjunto de N objetos $X = \{x_1, ..., x_n\}$ y dos particiones de X a comparar, $U = \{u_1, ..., u_k\}$ y $V = \{v_1, ..., v_{k'}\}$ si se consideran lo siguiente:

- a es el número de pares de elementos en X que están en el mismo subgrupo en U y en el mismo subgrupo en V.
- b es el número de pares de elementos en X que están en diferentes subgrupos en U y en diferentes subgrupos en V.
- c es el número de pares de elementos X que están en el mismo subgrupo en U y en diferentes subgrupos en V.
- d es el número de pares de elementos en X que están en diferentes subgrupos en U y en mismo subgrupos en V.

Se puede construir el Rand Index (RI) como:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{2.3}$$

En este caso, $a + b$ puede considerarse como el número de consensos entre U y V, además $c + d$ es la cantidad de discrepancia entre U y V. Por último $a + b + c + d$ es la cantidad total de pares de elementos. En resumen el Rand Index es la frecuencia de ocurrencias de conformidades en la asignación de elementos a un cluster sobre el total de pares o también visto como la probabilidad de que U y V estén de acuerdo sobre pares elegidos aleatoriamente.

El RI resulta en un valor entre 0 y 1, donde 0 indica que los clusters no concuerdan en ningún par de elementos y 1 indica que los clusters son exactamente los mismos. Un problema que presenta este índice es que para agrupaciones aleatorias el resultado no toma un valor constante. Para mejorar este resultado y reducir el factor de aleatoriedad es que se ajusta el cálculo del índice, logrando la versión conocida como Adjusted Rand Index (ARI) [Hubert, 1985].

El ARI tiene el resultado de generar índices entre valores de [-1, 1]. Esto se debe principalmente a que se basa en información de superposición entre los subgrupos U y V. Esta superposición se representa a través de una matriz de contingencia como muestra la tabla 2.1, extraída de [Theodoridis, 2003].

U/V	v_1 v_2 ... $v_{k'}$	Sumas
u_1	$n_{1,1}$ $n_{1,2}$ ... $n_{1,k'}$	a_1
u_2	$n_{2,1}$ $n_{2,2}$ ... $n_{2,k'}$	a_2
.	.	.
.	.	.
u_k	$n_{k,1}$ $n_{k,2}$ ... $n_{k,k'}$	$a_{k'}$
Sumas	b_1 b_2 ... b_k	

Cuadro 2.1: Matriz de Contingencia, registra la información común entre subgrupos.

n_{ij} denota el número de objetos que son comunes entre el subgrupo u_i y v_j $|u_i\ v_j|$. Los a_i son el número de objetos que hay en la clase u_i y los b_j son el número de elementos que hay en la clase v_j.

El índice ARI se define como $\frac{índice - índice_{esperado}}{índice_{máximo} - índice_{esperado}}$ y más precisamente como:

$$\frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}\right]/\binom{n}{2}} \tag{2.4}$$

$n_{ij}, a_i\ y\ b_j$ valores de la matriz de contingencia.

2.2.1.2. Validación Interna

La validación interna intenta evaluar la calidad de las estructuras generadas desde un punto de vista teórico, es decir, que tan bien cumple un cluster con su definición; a diferencia de la externa que intenta determinar si las estructuras generadas son las correctas. En este tipo de validaciones solo se mide la calidad basándose en información de los datos sin necesidad de conocimiento externo al problema, esto hace que sean las técnicas más utilizadas.

La mayoría de los índices utilizados para la validación interna se basan en la cohesión y la separación de los clusters. No existe un criterio genérico que determine qué índice tiene mejor comportamiento para cada caso [Introini, 2011].

Se entiende por cohesión de un cluster a cuán relacionados están los elementos de un mismo cluster y por separación qué tan separados están los clusters entre sí.

Algunos índices son:

- **Calinski-Harabasz:** Este índice está definido como la razón entre la dispersión interna de los clusters y la dispersión entre clusters. El objetivo es maximizar el

valor de la función CH definida por:

$$CH(P) = \frac{(N - |P|)inter_{CH}(P)}{(|P| - 1)intra_{CH}(P)} \tag{2.5}$$

$inter_{CH}(P)$ denota el error suma de cuadrados entre diferentes clusters.

$$inter_{CH}(P) = \sum_{k=1}^{K} |C_k| \| \bar{C}_k - \bar{x} \|^2 \tag{2.6}$$

K es la cantidad de clusters. $|C_k|$ es la cantidad de elementos en el cluster k. $\bar{C}_k$ es el centroide del cluster k y $\bar{x}$ es el centro de todo el conjunto de datos [Desgraupes, 2013].

$intra_{CH}(P)$ denota la diferencia al cuadrado de todos los objetos en un cluster con respecto a su centroide ($\bar{C}_k$).

$$intra_{CH}(P) = \sum_{k=1}^{K} \sum_{i=1}^{N} w_{k,i} \| x_i - \bar{C}_k \|^2 \tag{2.7}$$

- **Davies-Bouldin:** Se emplea como medida de cohesión de un cluster la media de las distancias de sus puntos a su centroide, mientras que como medida de separación utiliza la distancia entre clusters. La elección de la cantidad óptima de clusters se toma minimizando el índice de Davies Bouldin dado que eso significa que los clusters son más compactos y están más separados [Davies, 2000, 224–227]. Esta es una desventaja del índice, porque al calcular la compacidad según la distancia de los puntos a los centros, no detecta bien la forma de los clusters y si además, éstos están solapados, no es posible determinar las fronteras entre los clusters, dando resultados erróneos acerca del número de grupos.

  El índice se calcula de la siguiente forma:

$$DB(P) = \frac{1}{|P|} \sum_{C_k \in P} \max_{c_l \in P/C_k} \left\{ \frac{S(C_k) + S(C_l)}{d(\bar{C}_k, \bar{C}_l)} \right\} \tag{2.8}$$

$$S(C_i) = 1/|C| \sum_{x \in C_i} d(x, \bar{C}_i) \tag{2.9}$$

Donde $S(C_i)$ denota la distancia media de todos los elementos del clúster i al centroide $\bar{C}_i$. Por otro lado, $d(\bar{C}_i, \bar{C}_j)$ denota la distancia entre los centroides $\bar{C}_i$ y $\bar{C}_j$. $|C|$ denota la cantidad de clusters.

- **Dunn:** El índice Dunn corresponde a la proporción de la distancia más pequeña entre las observaciones de diferentes clusters y la distancia inter-cluster más grande [Dunn, 1974, 95–104]. El número de clusters que maximiza $Dunn(P)$ se considerará el correcto y se calcula del siguiente modo:

$$Dunn(P) = \frac{inter_{Dunn}(P)}{intra_{Dunn}(P)} \tag{2.10}$$

$$inter_{Dunn}(P) = \min_{C_k \in P}\{\min_{c_1 \in P/C_k}\{\delta(C_k, C_1)\}\} \tag{2.11}$$

$$\delta(C_k, C_1) = \min_{x_i} \in C_k\{\min d(x_i, x_j)\} \tag{2.12}$$

$$intra_{Dunn}(P) = \max_{c} \in P\{\max_{x_i, x_j \in C} d(x_i, x_j)\} \tag{2.13}$$

- **Silhouette:** El índice silhouette, al igual que el índice Dunn, mide cuán compactos y separados están los clusters [Rousseeuw, 1987, 53–65]. Este índice corresponde al promedio del valor Silhouette de cada elemento y mide que tan similar es un elemento con el cluster al que pertenece (cohesión) en comparación con otros clusters (separación). El índice resulta en un valor comprendido entre -1 y 1, donde los valores próximos a 1 indican que los elementos del cluster se encuentran correctamente asignados (alta cohesión y alta separación). En cambio si la mayoría de los índices son lejanos a 1, significa que la agrupación no es la más adecuada y existen elementos que en el mejor caso se encontrarían en otro cluster. Para un elemento i resultado de clusterizar con un algoritmo X y obtener k clusters, el índice de Silhouette se define de la siguiente manera:

$$S(i) = \frac{b_i - a_i}{\max\{b_i, a_i\}} \tag{2.14}$$

En la ecuación 2.14 se tiene que a_i es el promedio de disimilitud de i con todos los otros elementos en el mismo cluster. Esto es "cuán bien i es asignado al cluster que pertenece". Por otro lado se tiene que b_i es la disimilitud promedio más baja de i a cualquier otro cluster (al cual no pertenece). El cluster con esta disimilitud promedio más baja es conocido como el cluster "vecino" porque es la siguiente mejor asignación de i a otro cluster.

Según [Rousseeuw, 1987, 53–65], el índice de Silhouette también se puede interpretar como:

$$S(i)\begin{cases} 1 - a_i/b_i, & \text{if } a_i < b_i \\ 0, & \text{if } a_i = b_i \\ b_i/a_i - 1, & \text{if } a_i > b_i \end{cases} \tag{2.15}$$

La ecuación 2.15 muestra como para que $S(i)$ sea cercano a 1, se debe dar que $a_i \ll b_i$. Dado que a_i mide cuán disimilar es i a su propio cluster, un valor chico es lo deseable. Además un valor grande de b_i implica que i no es parecido a los elementos del cluster "vecino".

Las métricas de validación interna pueden usarse para escoger el mejor algoritmo de clustering, así como el número de clusters óptimo sin ningún tipo de información adicional.

2.2.1.3. Validación Relativa

La principal característica de la validación relativa es que se basa en la medición de la consistencia de los algoritmos, comparando los clusters obtenidos por un mismo algoritmo bajo diferentes condiciones [Brun, 2005, 807–824]. Es decir, su operación parte de evaluar un cierto cluster con respecto a varios esquemas de clustering y/o bien de ejecutar un mismo procedimiento de clustering con diferentes parámetros en reiteradas ocasiones.

La base de los métodos de validación externa e interna es la evaluación estadística. Esto tiene la gran desventaja de tener una demanda computacional muy elevada. Un acercamiento que erradica esta dificultad es la validación relativa. A continuación se plantea una definición más formal del problema que se quiere resolver con esta validación:

"Sea P_{alg} el conjunto de parámetros asociados a un algoritmo de clustering específico (por ejemplo: el número de centroides nc). Entre las diferentes posibles centros $C_i, i = 1, ..., nc$, definidos para algún algoritmo específico, para diferentes valores de los parámetros en P_{alg}, elegir el que mejor se ajuste al conjunto de datos" [Halkidi, 2001a, 413–441].

Aquí se pueden considerar algunos problemas, como ser:

1. P_{alg} no contiene el número de centroides, nc, como parámetro.
   Encontrar el óptimo nc se reduce a correr el algoritmo en varias ocasiones hasta encontrar un rango de valores donde nc se mantenga constante.

2. P_{alg} contiene nc como parámetro.
   En este caso, el proceso de identificar el mejor esquema de clustering se basa en un índice de validación.

Algunos de los índices más utilizados para la validación relativa son el índice de Dunn y su versión generalizada [Bezdek, 1998, 301–315].

Índice de Dunn generalizado (V_{GD})

Sea $X \subset R^l$, un conjunto de datos, sea $C_j \subset X, j = 1, 2, ..., k$ donde se cumple que $\cap_{j=1}^{k} C_j = \emptyset$ y $\cup_{j=1}^{k} C_j = X$, sea $\delta_i : P(R^l) * P(R^l) \to R^+$ y $\Delta_j : R^l \to R^+$ funciones

que miden la distancia entre las clases y diámetro, respectivamente. Entonces se define el índice de Dunn generalizado como:

$$V_{GD}(U) = V_{\delta_i \Delta_j}(U) \tag{2.16}$$

$$V_{GD}(U) = \min_{1 \leqslant S \leqslant k} \left\{ \min_{1 \leqslant t \leqslant k, t \neq s} \left\{ \frac{\delta_i(C_s, C_t)}{\max_{1 \leqslant h \leqslant k} \{\Delta_j(C_h)\}} \right\} \right\} \tag{2.17}$$

El índice de Dunn se basa en δ_1 y Δ_1 definidas como sigue:

$$\delta_1(S,T) = \delta_{min}(S,T) = \min \{d(x,y) | x \in S, y \in T\} \tag{2.18}$$

$$\Delta_1(S) = \max \{d(x,y) | x, y \in S\} \tag{2.19}$$

Con esto se busca que la ecuación 2.17 tenga su máximo valor en la partición óptima. Este índice tiene algunas desventajas como el tiempo de cálculo y la sensibilidad a datos "ruidosos". Al mismo tiempo, el diámetro del cluster (Δ_1) se ve afectado por la adición o eliminación de un dato del cluster. Como forma de contrarrestar estos problemas surge la necesidad de generalizar el índice de Dunn.

La generalización del índice tiene como objetivo plantear una definición más correcta para el diámetro del cluster Δ_y la distancia entre clusters δ y así poder validar clusters de diferentes tipos sin penalizaciones propias del índice. La generalización de Dunn permite además disminuir la sensibilidad al ruido y validar clusters volumétricos de formas complejas.

2.3. Clustering de documentos de texto

El clustering de documentos no es más que el problema de clustering general tratado hasta el momento pero instanciado en un ámbito de documentos de texto. Es decir, lo que se intenta agrupar son textos. Este problema suele ser interesante cuando se necesitan procesar numerosos documentos. De esta forma se pueden ahorrar recursos teniendo textos similares agrupados según ciertos criterios, esto eventualmente facilita futuras etapas tras homogeneizar el área de estudio.

2.3.1. Representación de documentos

Como se menciona al comienzo de este capítulo, la representación de los objetos a agrupar es fundamental para resolver el problema. En el caso de los documentos de texto una técnica muy utilizada es la representación mediante "Bag of Words" (BOW). Es decir, se representa cada documento con un vector, de largo todas las palabras del documento. Por ejemplo, indicando si una palabra está presente o no.

También existen otras alternativas, como ser la "Matriz de Frecuencia de Términos" (TF-IDF). Esta es muy similar al BOW pero se tienen en cuenta las frecuencias a la hora de formar los vectores y se obtienen así mejores representaciones. También se tiene en cuenta si una palabra aparece muchas veces o pocas. De este modo se puede identificar palabras que no aportan a la agrupación y que pueden ser eliminadas de la representación sin pérdida de información. Un ejemplo posible puede ser la ocurrencia de una palabra exactamente una vez en cada documento. Es de imaginarse que esta no agrega información valiosa y por ende no sería necesario tenerla en cuenta.

Consideremos el siguiente ejemplo:

Ejemplo 2. *Ejemplo de TF-IDF.*

d1 = *"Hoy es jueves"* $\rightarrow (1,1,1,0,0) \rightarrow v_1 = (1,1,0,0)$

d2 = *"Mañana es viernes"* $\rightarrow (0,1,0,1,1) \rightarrow v_2 = (0,0,1,1)$

d3 = *"Es viernes"* $\rightarrow (0,1,0,0,1) \rightarrow v_3 = (0,0,0,1)$

En este caso la palabra "es" aparece en todos los documentos y se puede eliminar obteniendo así un vector más distintivo para cada documento.

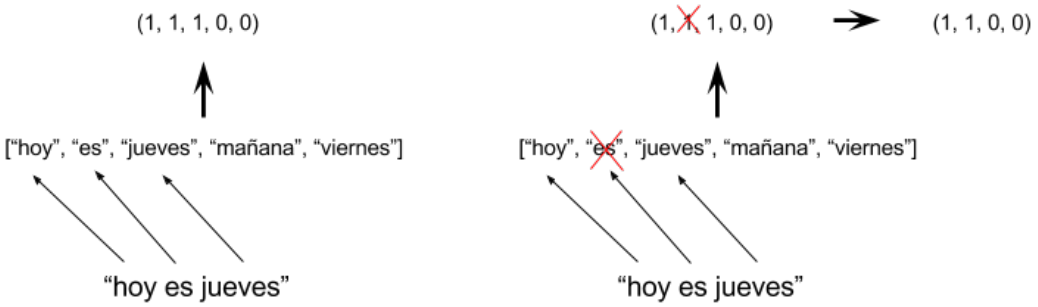


Figura 2.8: Representación "hoy es Jueves"

Definición de distancia

El segundo problema a resolver es la definición de distancia entre documentos. Debido a que los documentos son representados como vectores, es trivial utilizar la distancia euclídea que se define a continuación. Sin embargo, esta distancia no suele ser utilizada ya que por lo general para reconocer que dos documentos mencionan al mismo tema basta saber si los vectores tienen la misma dirección y sentido, para esto se suele utilizar la medida del ángulo que forman los vectores, también conocida como la distancia coseno.

La distancia coseno suele definirse como (1 - la similaridad), siendo la similaridad el producto punto de los vectores divido por el producto de las normas de cada uno de ellos:

$$coseno(d_1, d_2) = \frac{d_1 \bullet d_2}{\|d_1\|\|d_2\|} \tag{2.20}$$

La distancia euclídea entre los vectores $d_1 = (d_{1_0}, d_{1_1}, ..., d_{1_n})$ y $d_2 = (d_{2_0}, d_{2_1}, ..., d_{2_n})$ se define como:

$$euclídea(d_1, d_2) = \sqrt{\sum_{i=1}^{n}(d_{1_i} - d_{2_i})^2} \tag{2.21}$$

Particularmente en el ejemplo de los días, considerando los vectores $v_1 = (1, 1, 0, 0)$ y $v_2 = (0, 0, 1, 1)$ se tiene:

$$coseno(d_1, d_2) = coseno((1, 1, 0, 0), (0, 0, 1, 1)) = \frac{(1, 1, 0, 0) \bullet (0, 0, 1, 1)}{\|(1, 1, 0, 0)\|\|(1, 1, 0, 0)\|} \tag{2.22}$$

$$= \frac{0}{\sqrt{(1^2+1^2)}*\sqrt{(1^2+1^2)}} \Rightarrow 1 - 0 = 1$$

$$euclídea(d_1, d_2) = euclídea((1, 1, 0, 0), (0, 0, 1, 1)) = \sqrt{1^2 + 1^2 + (-1)^2 + (-1)^2} = \sqrt{4} = 2 \tag{2.23}$$

De esta forma también se computan (v_1, v_3) y (v_2, v_3). Obteniendo los siguientes resultados (tabla 2.2)

	Euclídea	Coseno
(v_1, v_2) "Hoy es jueves" vs "Mañana es viernes"	2	1
(v_1, v_3) "Hoy es jueves" vs "Es viernes"	1.7	1
(v_2, v_3) "Es viernes" vs "Mañana es viernes"	1	0.3

Cuadro 2.2: Ejemplo de distancias.

Para este caso particular se tiene que la distancia coseno permite distinguir mejor los vectores que la euclídea. Para la distancia coseno se tiene que "Hoy es jueves" esta a la misma distancia de "Mañana es viernes" que de "Es viernes". Sin embargo la distancia euclídea no permite identificar esta relación tan directamente.

2.4. Representaciones vectoriales de palabras

Son una forma de modelar palabras en un documento mediante vectores de alta dimensión conformados por números reales. Lo que se busca es obtener una representación numérica de tamaño manejable de las palabras en un documento. Esto permite poder

aplicar distintas técnicas de procesamiento de lenguaje natural sobre los vectores resultantes.

Existen múltiples métodos para lograr obtener estos vectores en base a un corpus. Aquellos más utilizados involucran redes neuronales, reducción de dimensionalidad y matrices de coocurrencia.

Un posible modelo presentado hace algunos años por investigadores de Google es el modelo "Word2Vec" propuesto por [Mikolov, 2013] el cual proporciona una implementación eficiente de los modelos de bow-of-words y skip-gram para computar representaciones vectoriales de palabras. El modelo Word2Vec se genera mediante redes neuronales, y debido a su alta eficiencia es apto para ser utilizado con conjuntos de textos de varios órdenes de magnitud mayores a los utilizados para entrenar el modelo Skip-gram. Algo que se destaca del modelo es la "capacidad" para lograr aprender relaciones entre palabras, lo cual sin duda brinda muchas posibilidades. De un corpus suficientemente grande y diverso se pueden incluso realizar operaciones matemáticas con vectores y que los resultados tengan sentido.

En la figura 2.9, extraída de un tutorial de Tensor Flow, se pueden observar algunas relaciones posibles que son extraídas directamente del modelo aprendido de un corpus particular.

Relaciones aprendidas

- Mujer es a reina, como hombre es a rey.
- caminando es a caminado, como nadando es a nadado.
- La distancia entre la palabra que representa un país y su capital es aproximadamente igual para todos los países con sus respectivas capitales.

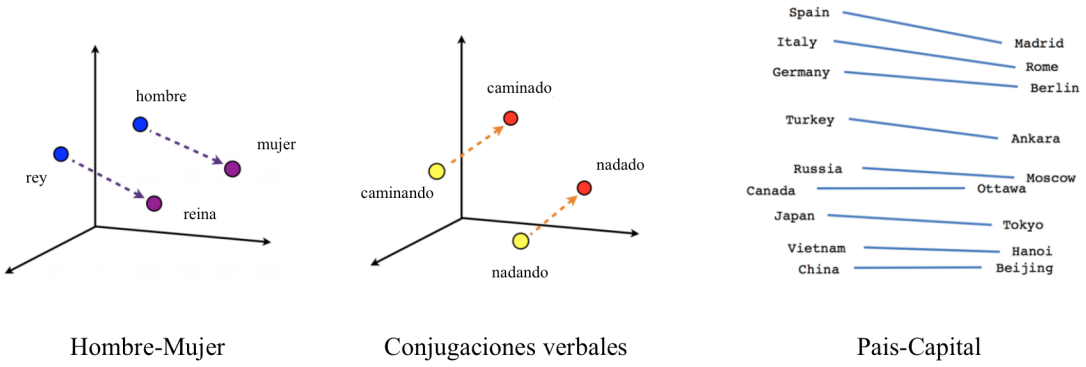


Figura 2.9: Relaciones entre palabras de conjugaciones verbales, sexo y ciudades-capitales.

Hasta este punto, se han presentado los modelos de clustering de mayor renombre y uso en el área de la minería de datos. También se establecieron los criterios de validación

existentes para estos modelos y se ha tratado el tema de clustering de texto y sus técnicas de representación. A continuación, se presentan algunos trabajos realizados por diferentes autores que hacen uso de lo antedicho, y que dan pie al proceso de clustering de tweets de entidades uruguayas de este trabajo.

2.5. Trabajos previos

En esta sección se analizan algunos trabajos previos que son de gran utilidad para basar nuestro estudio. Existen varios enfoques, algunos utilizan métodos supervisados, otros no supervisados y también combinaciones de estos. Los diferentes enfoques que se presentan son de gran ayuda y plantean ideas interesantes para ser combinadas y así tratar de obtener mejores resultados para la implementación que se realizará a posteriori. Del mismo modo constituye una fuente de información importante a la hora de tomar decisiones sobre qué estrategia utilizar.

Topical Clustering of Tweets [Dela Rosa, 2011]

En este trabajo el objetivo final es explotar y agrupar un gran conjunto de tweets en seis grandes categorías. Estas seis categorías suponen seis clases o tópicos predefinidos. Cada tema se encuentra definido por un conjunto de hashtags determinado y que se suponen ser identificadores del tema que se quiere considerar. El corpus de tweets cuenta con aproximadamente 1,1 millones de tweets y se los procesa previo al clustering con el fin de eliminar ruido indeseado. Es importante destacar que de los tweets recolectados todos poseen al menos un hashtag. Este hashtag, además, es parte de una lista de 30 hashtags. La lista supone un cubrimiento de características adecuado para los seis temas objetivo.

Una vez procesados los tweets se aplican dos técnicas de clustering no supervisado, LDA y K-Means. Posteriormente se comparan los resultados con un Gold Standard basado en hashtags. Para este experimento los resultados obtenidos son bastante malos. Los índices obtenidos se presentan en la tabla 2.3. En el caso de la agrupación por hashtag, es decir el nivel "más fino" utilizado, los clusters son malos. Para el caso de los clusters por tema los resultados de evaluación son mejores ya que es más probable que un tweet esté en la clase correcta si hay menos clases, pero aun así las medidas siguen siendo bajas.

LDA	**Purity**	**Pairwise F-Score**
Por hashtag	0.131	0.048
Por tema	0.304	0.143
K-Means	**Purity**	**Pairwise F-Score**
Por hashtag	0.285	0.058
Por tema	0.412	0.143

Cuadro 2.3: Resultados según los índices de validación utilizados.

El segundo análisis consiste en aplicar una técnica de clasificación supervisada, un

clasificador Rocchio [Manning, 2009]. Se utiliza como método de representación el bag-of-words y se lo evalúa a través de métricas estándares, Precisión, Recall y Medida F. En esta ocasión el desempeño es mejor según los autores, pero no se presentan los índices nuevos de la misma forma que en la primera etapa y por lo tanto no son comparables. También en este estudio se analiza cómo la cercanía temporal de los conjuntos de tweets afecta la performance. Luego de este análisis se concluye que la separación temporal de los tweets afecta negativamente el desempeño ya que las tendencias en Twitter suelen fluctuar. Por ese motivo es bueno mantener los conjuntos de datos con separaciones temporales de tiempo razonables.

Otra observación considerable es la aplicación de "expansiones" de los tweets. A aquellos tweets poseedores de URLs en su contenido (aproximadamente 39 %) se les agrega el contenido del sitio referenciado en la URL como parte del tweet. Esto resulta en efectos negativos sobre el algoritmo de clustering. Se observa un descenso en las medidas utilizadas por los autores. Según un análisis posterior se determinó que la mayoría de las URLs no tienen relación con el contenido del tweet y se podían considerar "spam", generando tan solo ruido en los tweets.

Interpreting Twitter Data From World Cup Tweets [Godfrey, 2014]

En este trabajo, se utilizan técnicas de clustering para explorar y extraer patrones de Twitter durante un evento particular, el mundial de fútbol del año 2014 en Brasil.

En este estudio además de los ya mencionados algoritmos existentes para realizar clustering, también se utiliza una técnica conocida como "Clustering por consenso". La misma se basa en la utilización de múltiples algoritmos (o variantes de un mismo algoritmo) para lograr identificar de forma más precisa los clusters. Si dos tweets se mantienen juntos tras procesarlos con distintos algoritmos, se puede afirmar con más certeza si deben pertenecer al mismo cluster o no.

Para realizar este estudio se recolectan tweets que fueron emitidos durante el transcurso del mundial y que contenían las palabras "world cup". Inicialmente se recolectaron casi 30 mil tweets, sin embargo luego de analizar y procesarlos solo se consideraron útiles aproximadamente 17 mil ($\tilde{5}7\,\%$). Particularmente se decide no mantener los re-tweets ya que se piensa que podrían afectar negativamente los conteos de palabras. Además se eliminan tweets ruidosos basándose en la matriz de consenso. Para esto último se ejecuta K-Means con múltiples valores de K y se eliminan aquellos tweets tales que no pertenecen al mismo cluster menos del 10 % de las veces.

A pesar de que en el trabajo no se menciona cómo se realizó la validación de los clusters es interesante destacar como se muestran los datos. La visualización es una parte importante del análisis de clusters porque permite extraer conclusiones y analizar los datos gráficamente. Para visualizar los datos se utilizan nubes de palabras y grafos generados con una herramienta llamada Gephi. Lográndose resultados como el de la figura 2.10 que resulta de ejecutar K-Means. Se puede apreciar algunos de los clusters con mayor cantidad de elementos, entre ellos, "brazil", "falcao" y "fifa".

Figura 2.10: Falcao/Spanish/stadium tópicos de K-Means.

Semantic Expansion of Hashtags for Enhanced Event Detection in Twitter [Ozdikis, 2012]

Este trabajo tiene como objetivo la detección de eventos a través de los hashtags. La estrategia de detección de eventos se basa en realizar clustering sobre los tweets considerando como criterio los hashtags, y así lograr identificar patrones y agrupaciones.

En este caso particular, el proceso de recolección de tweets se restringió a Turquía. Se definieron zonas que en su conjunto cubrieran casi la totalidad del país para así recolectar casi 400 mil tweets.

Para la representación de los tweets se utilizó una representación vectorial generada mediante TF-IDF. Una vez obtenidos los vectores se los extendió semánticamente. Este proceso consistió en agregar palabras relacionadas a cada palabra del tweet si esta extensión ya no está presente. Por ejemplo, si un tweet posee la palabra “alexsandra”, entonces al mismo se le agrega “alexandra” que es la palabra semánticamente más similar en el corpus.

Para determinar las relaciones semánticas dentro del corpus se cuentan las ocurrencias de a pares de todas las palabras. Por ejemplo, en el momento de la extracción de los tweets se disputaba un partido entre “Galatasaray” y “Fenerbahce”, esta relación presente en los datos permitió detectar ese evento agrupando muchos de los tweets que mencionan una de las dos palabras o ambas.

Al finalizar la generación de vectores se utiliza un método de clustering aglomerativo utilizando como representación de los tweets los vectores normalizados, con la adición de una dimensión más, el tiempo. Para determinar la similaridad entre vectores y su pertenencia a un cluster se utiliza la similitud del coseno.

Una vez obtenidos los clusters se procede a evaluar el resultado. Para esto, se consideró validación externa clusterizando manualmente aproximadamente 6300 tweets, que en su mayoría pertenecían a dos eventos que ocurrieron durante los tres días que duró la recolección de tweets.

Los resultados obtenidos son muy buenos para los casos en que se usó la “expansión

semántica", obteniéndose valores aproximados de Precisión de 0.99, Recall de 0.80 y Medida F de 0.88.

Otros trabajos

También se analizaron otros trabajos como [Tang, 2014] y [Antenucci, 2011] en los cuales se menciona el concepto de expansión de tweets y también se intenta realizar clustering a través de los hashtags.

Capítulo 3

Corpus

El presente capítulo describe el proceso de obtención de tweets que forman parte del corpus de este trabajo. Se describe cómo fue evolucionando el proceso al igual que las tomas de decisiones para conformar el corpus final. Se dedica, además, una breve sección para describir la mecánica de anotación que se utilizó para comenzar con el armado del corpus. Este capítulo también presenta en detalle temas claves como la integración con Twitter. Se discuten los dos mecanismos básicos de integración que provee la plataforma presentando sus características y las razones para finalmente considerar tan solo uno de ellos. Por último se muestran estadísticas del corpus generado, lo cual permite conocer en mayor detalle su conformación.

Antes de comenzar con la obtención del corpus, es importante resaltar que la construcción se realizó de manera progresiva, realizando leves modificaciones a demanda. Es decir, se fue ajustando de manera tal de ir supliendo los objetivos y desafíos que se querían resolver en cada momento.

El problema original se centraba en "saber de quién y de qué se habla" en un tweet. Para esto, fue de interés obtener información de la mayor cantidad de fuentes distintas, siendo cada cuenta de Twitter una posible candidata. Esta característica permite no sesgar la opinión pública sólo a aquellas opiniones de las fuentes seleccionadas. Por ejemplo, obteniendo tweets solamente de un portal de noticias, se podría saber si el portal tiene cierta tendencia a hablar más de un tema o de una persona, pero esto no necesariamente refleja la opinión de las masas. Por este motivo se descartan métodos de obtención de tweets como los propuestos en [Cubero, 2015], donde la información se extrae únicamente de cuentas puntuales y previamente seleccionadas.

La inmensidad de cuentas existentes en Twitter presenta rápidamente un problema por lo cual surge la necesidad de buscar nuevas estrategias para su explotación. Por este motivo, se establecen ciertos criterios de búsqueda que reducen el universo de fuentes a un número más manejable y razonable, éstos se definen en la sección de obtención del corpus.

Luego de la obtención de los tweets y obtención de un primer corpus estable, se procede a realizar un proceso de anotación el cual se presenta a continuación.

3.1. Anotación

Esta sección sólo fue válida para lograr el primer objetivo del proyecto. Por este motivo no se profundiza en detalle, sin embargo, resultó de gran relevancia para lograr descubrir un patrón que forzó al equipo a cambiar de objetivo. Esto desencadenó en el tema de clustering de tweets.

El objetivo inicial del proyecto era identificar de quién hablaba un tweet, para esto fue necesario contar con un corpus etiquetado. En otras palabras, asociar a cada tweet una o varias personas de la lista de candidatas, o "desconocido" en caso que no coincidiera con ninguno. Para llevar a cabo esta tarea se construyó una aplicación web que permitiese a los etiquetadores marcar cada tweet según si se habla o no de una o varias personas. El resultado, un corpus donde cada uno de los tweets estuviese etiquetado con una o más entidades o en su defecto con la etiqueta de "entidad desconocida".

Proceso de anotación

La estrategia llevada adelante para satisfacer la etapa de etiquetado consistió en el uso de la aplicación por cuatro personas designadas; nuestros dos tutores y nosotros. La tarea consistió en dado un tweet mostrado en pantalla seleccionar el o los posibles candidatos.

La aplicación consistió en la presentación de un tweet por vez, cada tweet era mostrado secuencialmente según su fecha de registro en el sistema para así evitar repeticiones. Al mismo tiempo que el tweet se hacía visible en pantalla, se sugería una etiqueta autogenerada. La cual se correspondía con la o las entidades para la cual ese tweet estaba presuntamente asociado. Esta etiqueta autogenerada no es más que la asignación realizada producto de la búsqueda de los tweets a través de la REST API de Twitter. Una peculiaridad de la etiqueta es que podía corresponderse a múltiples personas o ninguna. Esto se calculó en base a coincidencias de palabras clave lo cual se detalla en la sección de 3.2.

Dado el tweet y la etiqueta sugerida, el etiquetador debía confirmar la etiqueta autogenerada o modificarla. Para ello se elegían o borraban las selecciones para las entidades que se creían estar vinculadas al tweet. Además, en caso de dudas acerca de su etiqueta, existía la posibilidad de establecer la etiqueta de "no convincente" para plasmar que la selección no es lo suficientemente predecible. A modo de ejemplo, se muestra la figura 3.1.

Criterios de anotación

Una vez que un tweet es etiquetado por las cuatro personas, ese tweet se encontraba apto para recibir una etiqueta final que determinara de quién se hablaba (o de quién no se hablaba) en caso que los etiquetadores estuvieran de acuerdo.

Se considera que un tweet habla de una persona si se puede responder alguna de las siguiente preguntas de forma afirmativa:

Figura 3.1: Instancia de etiquetado para tweet de Luis Suárez.

1) *¿Se dice algo de <x-persona>?*

2) *¿Me permite saber algo de <x-persona>?*

Este criterio fue revisado y aprobado por todos los involucrados en el proceso de anotación y fue el único criterio válido utilizado para decidir si un tweet habla o no de una persona.

Resultado del proceso de anotación

Luego de cuatro semanas y habiendo etiquetado alrededor de 1400 tweets y con algunos pendientes de ser etiquetados aún, existían percepciones sobre la existencia de un patrón en los tweets etiquetados.

Partiendo de la base de que los tweets son textos muy cortos (140 caracteres como máximo) quien escribe un tweet, en caso de querer hablar de alguien en particular debe mencionarlo o hacer referencia explícita a través de una mención a su cuenta. De esta forma se hace necesario utilizar su nombre completo o un apodo que identifique a la persona. Estas condiciones permiten que el mensaje sea entendido sin precisar de mayor contexto, sin embargo se vuelve una condición suficiente para identificar de quién se habla en la amplia mayoría de los tweets.

Esto último no es como fue pensado el uso de menciones, según se definen en las guías de la plataforma, el propósito es que sea usado para satisfacer la necesidad de dirigir un mensaje a alguien en particular. Esto se usa erróneamente y en la amplia mayoría de los

casos, es usado para decir de quién se habla. De esta manera se establece una línea base muy alta y que casi resuelve automáticamente el problema que se quería resolver.

Este patrón se confirma analizando las etiquetas autogeneradas por el proceso de recolección y comparando con las etiquetas asignadas por los etiquetadores. Este análisis muestra que el 72 % de las etiquetas autogeneradas que fueron asignadas a una entidad son correctas (según el criterio definido por los anotadores), y en algunos casos se llega incluso al 100 %. Además aproximadamente el 25 % restante se corresponde a casos donde la etiqueta autogenerada no se corresponde con ninguna entidad. De esta forma se observa un alto porcentaje de acierto por el proceso de extracción de tweets y se cumple el objetivo rápidamente. Esto genera la necesidad de determinar un nuevo objetivo.

El nuevo problema que se quiere resolver parte de la base de que se tienen tweets correctamente identificados con la persona a la que hacen referencia. Y lo que se quiere lograr es agrupar los tweets en base a los temas que se tratan en ellos. Esto es conocido como clustering, en este caso, de tweets.

Dado que existe un nuevo objetivo por alcanzar, se vuelve necesario realizar ajustes en la forma de recuperar los tweets. Algunas de las nuevas tareas involucran generalizar en mayor grado las búsquedas y mejorar los criterios para las nuevas necesidades. El resultado de esta nueva forma de obtener tweets termina siendo el nuevo corpus a ser utilizado para enfrentar el problema de clustering. A continuación se presenta como se construye el corpus.

3.2. Obtención del Corpus

Como primer paso de la obtención del corpus, se realiza una selección de personalidades uruguayas bajo cierto criterio de reconocimiento a nivel nacional e internacional. Particularmente se eligen personas del ámbito futbolístico y político. Esta decisión es influenciada por algunos portales como por ejemplo AdSocia, El Observador y Pantallazo que calcularon algunos rankings sobre las cuentas de Twitter uruguayas. En los resultados se destacan atributos como la cantidad de seguidores y la puntuación de Klout score. Estos dos atributos en combinación con nuestros criterios personales permitieron la toma de decisiones para conformar la lista de candidatos a estudiar. En las tablas 3.3 y 3.4 se encuentra la lista completa de personas que la conforman cinco deportistas y once políticos.

Una vez determindadas las entidades se considera una lista que contiene el conjunto de palabras específicas para cada entidad. Estas palabras fueron seleccionadas únicamente bajo nuestro criterio, permitiendo así identificar a las entidades elegidas de una forma bastante amplia. Es importante destacar que la lista de palabras claves incluye entre ellas la cuenta de Twitter (en caso de poseer una), hashtags relacionados a la persona y su nombre completo como se muestra en la tabla 3.1.

Utilizando esta estrategia se genera el corpus necesario. Los criterios de búsqueda establecidos para cada entidad fueron sufriendo modificaciones hasta lograr resultados convincentes desde nuestro punto de vista. El corpus resultante es el utilizado finalmente para llevar a cabo la resolución del problema y está compuesto mayoritariamente de

Persona	Palabras clave
Luis Suárez	Luis suárez, luissuarez9, pistolero, lucho suarez, luisito suarez, luis, suarez, barcelona
Tabaré Vázquez	Tabaré vázquez, presidente de la república, tabaré, vázquez, FA

Cuadro 3.1: Palabras claves para Luis Suárez y Tabaré Vázquez.

tweets que potencialmente mencionan a las entidades seleccionadas, logrando además el objetivo de abarcar múltiples fuentes de origen, y evitando así el posible sesgo que se menciona en la introducción de este capítulo. En la tabla 3.2 se presentan algunos ejemplos de tweets recolectados.

Persona	Tweet
Luis Suárez	@LuisSuarez9 vistiendo la alternativa de @Uruguay `https://t.co/CL7vSDRrqA`
Tabaré Vázquez	Vazquez habla de ética? Sus discursos burlescos y de exterminio al oligarca(q tiene su capital en Uruguay) y no como su hijo en Panamá OFFS
Edinson Cavani	@ECavaniOfficial se quiere venir al atleti y no sabe como decirlo ya

Cuadro 3.2: Ejemplos de tweets recolectedos para Luis Suaréz, Tabaré Vázquez y Edinson Cavani.

3.3. Extracción de Tweets

El proceso de extracción de tweets se basa en la utilización de la interfaz que provee Twitter conocida comúnmente como REST Application Programming Interface (API). Si bien es una interfaz de gran utilidad para el cometido del proyecto, posee varias limitaciones. Esto hace que la obtención de tweets sea más desafiante.

Algunas de estas situaciones tienen que ver con la capacidad para recuperar tweets con fechas de antigüedad mayor a siete días. Esta limitante hace necesaria la implementación de un sistema de recolección de tweets basado en polling. El sistema deber ser capaz de retomar la búsqueda cada cierta cantidad de tiempo y de esta forma cubrir períodos de tiempo extensos sin pérdida de información. La rutina de extracción basa su búsqueda en la lista de palabras claves creada específicamente para cada persona.

También posee otras limitantes como ser la interrupción del servicio por excesivos llamados a la API a través del mismo usuario (rate limiting), o como el no indexado y omisión de tweets. Esto último aumenta las posibilidades de pérdida de información valiosa y provoca que los conjuntos de tweets recuperados sean más reducidos. Una consecuencia directa que se desprende de esta limitante es la reducción de la velocidad con la que se recuperan tweets y esto disminuye la capacidad de crecimiento del corpus.

Además, Twitter no recuerda los tweets proporcionados por la REST API a un usuario dado, por lo que se torna de suma importancia tener un control adecuado de duplicados.

En cuanto a las particularidades de los tweets obtenidos, existen tweets cuyo contenido coincide con el de algún otro tweet de manera idéntica. Estos casos se los define como retweet. Dado que son re-publicaciones de un mismo tweet pero a través de otro usuario, se opta por contemplar estos casos y conservar estos tweets. De esta manera se evita omitir información valiosa al momento de la construcción del corpus, ya que se considera que la repetición de contenido por parte de usuarios diferentes puede ser un indicio para la identificación de temas muy populares o "trending topics".

Teniendo presente las consideraciones anteriores, es razonable pensar que un corpus de magnitudes como las necesarias conlleva mucho tiempo. Para lograr construir el corpus se creó una rutina dedicada a la recolección de tweets que se ejecuta con una periodicidad de 12 horas.

Si bien se podría estimar cuántos tweets recoge el proceso en un día, esto no representa la realidad dado que existen muchos factores que afectan de manera directamente proporcional esta estadística. Por ejemplo, en aquellos días donde se define un torneo mundial de fútbol o en períodos de elecciones presidenciales, el flujo de tweets relacionados con la política y el fútbol serán ampliamente superiores a días donde no exista ningún evento de estas características.

Debido a la limitada evolución del tamaño del corpus, se fueron realizando extensiones al mismo. En un principio el corpus estaba únicamente conformado por tweets de origen uruguayo. Luego de algunos meses de experimentaciones y nuevas formas de búsqueda se decide generar un corpus que abarque más regiones. Para ello se decide valorar todos aquellos tweets escritos en español y que respeten los criterios de búsqueda establecidos para las entidades. En esta ocasión se aprecian aumentos sustanciales en el volumen de tweets.

En la siguiente sección se muestran algunas estadísticas relacionadas al corpus construido.

3.4. Estadísticas

Luego de 10 meses de recuperación sostenida, se logra construir un corpus con 419.133 ejemplares de los cuales 9.329 son de origen uruguayo y los restantes 409.804 del resto del mundo. Las siguientes tablas (3.3 y 3.4) muestran la distribución de tweets por entidad y por origen.

Curiosamente parecen haber más tweets del resto del mundo que de Uruguay, esto llama la atención en el caso de los políticos. Analizando los resultados, se identifica que la detección de la ubicación se establece en base a la configuración geográfica del usuario y esta no es un requisito obligatorio en Twitter. Muchos de los tweets se identifican como originarios del "resto del mundo" aunque es altamente probablemente que pertenezcan a Uruguay.

Futbolistas	**Cant. tweets (#t)**	**#t origen Uruguay**	**#t resto mundo**
Luis Suárez	**201.701**	**1446**	**200.255**
Edinson Cavani	39.884	568	39.316
Diego Forlán	21.890	190	21.700
Sebastián Abreu	11.964	142	11.822
Diego Godín	184	146	38

Cuadro 3.3: Deportistas uruguayos que conforman el corpus.

Políticos	**Nro. tweets (#t)**	**#t origen Uruguay**	**#t resto mundo**
Luis Lacalle Pou	**58.100**	1363	**56.737**
Pedro Bordaberry	26.295	803	25.492
Pablo Mieres	27.554	688	26.866
Jorge Larrañaga	19.990	439	19.551
Raúl Sendic	6.395	306	6.089
Luis Alberto Heber	6.391	286	6.105
Tabaré Vázquez	2.993	295	6.105
José Mujica	2.003	**1983**	20
Julio María Sanguinetti	344	344	0
Danilo Astori	273	79	194
Jorge Batlle	226	172	54

Cuadro 3.4: Políticos uruguayos que conforman el corpus.

3.5. Clustering manual

El clustering manual es un requisito fundamental para poder llevar a cabo validaciones externas. Como se mencionó en el capítulo 2, en la sección de validación, para poder validar un resultado externamente es necesario compararlo contra un Gold Standard. Llevar a cabo el clustering manual requiere cantidades de tiempo considerables. Por este motivo, se decide utilizar un subconjunto de tweets acotado que permita lograr los resultados esperados en un tiempo razonable. Para ello se decide seleccionar tweets pertenecientes a tres personas en tres fechas diferentes. De esta forma se obtienen tres conjuntos de tweets como insumo para realizar la validación externa:

a) El primero está conformado por tweets correspondientes a Luis Lacalle Pou en el día 9 de Diciembre de 2016 con 123 tweets.

b) El segundo se compone de 239 tweets pertenecientes a Edinson Cavani en el día 13 de Febrero de 2017.

c) El tercero corresponde a Luis Suárez en el día 6 de Enero de 2017 y se conforma por 113 tweets.

Cabe destacar que se cuenta con dos versiones del corpus de Edinson Cavani. Una versión clusterizada manualmente por el Anotador 1 (A1) y otra versión clusterizada por el Anotador 2 (A2). El fin es poder evaluar qué tanto se diferencian los conjuntos resultantes y qué tanto difieren clusters hechos manualmente por personas diferentes.

A continuación se realiza una comparación cuantitativa de los clusters obtenidos por A1 y A2 mediante el uso del índice externo ARI.

Evaluación cuantitativa entre clusters manuales

Para realizar el análisis se asume arbitrariamente que uno de los resultados generados de forma manual es correcto (A1 en este caso) y se lo compara con el otro (A2) mediante el índice ARI. Cabe señalar que este índice es simétrico, lo cual permite suponer que cualquiera de los dos resultados de clusterizar es el correcto sin pérdida ni mal uso de la información.

En caso de que el nivel de acuerdo o concordancia entre las agrupaciones generadas por una persona y por la otra sea elevado se esperaría obtener un valor del índice muy cercano a 1. En este caso, el resultado obtenido es 0.556, un número más bajo de lo que se esperaría a priori.

A continuación se realiza una comparación cualitativa de los clusters obtenidos en A1 y A2.

Evaluación cualitativa entre clusters manuales

Analizando cualitativamente los clusters resultantes generados manualmente, se intenta encontrar características que permitan entender por qué el índice ARI no alcanza un valor más cercano al óptimo.

La siguiente tabla muestra algunas observaciones iniciales (tabla 3.5).

Clustering Manual	**A1**	**A2**
Tweets ruido	27	62
#C	27 + cluster ruido	28 + cluster ruido
#C idénticos	9	9

Cuadro 3.5: Comparativa de las dos agrupaciones manuales A1 y A2 sobre el corpus de Edinson Cavani.

Considerando los 27 tweets asignados como ruido en el clustering de A1, existe una asignación que es ruido y forma parte de un cluster según el clustering manual de A2. Por el contrario, en 35 casos ocurre que A2 designa como ruido y A1 lo asocia a algún cluster. Esto significa que 26 tweets fueron designados como ruido en ambos, el resto uno u otro no lo designó como ruido. Se procede a analizar algunos de estos casos.

Lo que ocurre en la mayoría es que se dan diferentes niveles de granularidad, por ejemplo ocurre que hay algunos tweets haciendo referencia a que Cavani está pensando en el enfrentamiento contra Suárez y existen algunos tweets referidos al partido que se

disputará entre Suárez y Cavani. Esto se puede ver de varias formas, se puede considerar un cluster genérico que agrupe el enfrentamiento Suárez-Cavani, se puede considerar el partido @PSG_inside - @FCBarcelona dado que todos estos tweets también mencionan los equipos y la liga, pero también se puede tomar el enfoque que considera A2 para determinar que uno hace referencia a un pensamiento y el otro a un partido y en este caso asignarlos a clusters diferentes.

Otro caso similar se da con tweets de aliento para un partido a realizarse "mañana". Existen muchos tweets de diferentes índoles, algunos deseando buena suerte, otros planteando su posición y favoritismo por un equipo sobre el otro e incluso cuestionando a los anfitriones. Existen clusters genéricos que agrupan varios temas pero también ocurre que al realizar clustering más fino existen más tweets que quedan en agrupaciones de menos de tres elementos considerándose entonces como ruido.

De este modo, ocurre que la agrupación que puede realizar una persona sea un tanto diferente a la realizada por otra persona. Basta con tener algún conocimiento extra sobre un tema tratado en los tweets, o por el simple hecho de agrupar a diferentes niveles de granularidad. La diferencia en los tweets ruido que se generan con las dos agrupaciones hechas a mano, se debe a que a mayor nivel de granularidad, menor es la cantidad de tweets por tópico que se tiene. Esto resulta en la mayoría de los casos en la conformación de clusters de dos o un tweet. Previamente se decidió que estos conjuntos no son clusters por no alcanzar la cantidad mínima de elementos, resultando de este modo en tweets ruido.

Otra dificultad que se presenta a la hora de clusterizar manualmente es que se tienen muchos tweets de muchos temas diferentes. Lograr consolidar solo aquellos que hablan de un mismo tópico en un único cluster resulta un desafío interesante. Lograrlo requiere de mucho tiempo, concentración y una buena estrategia de agrupación. Claro está que cada uno empleó la técnica que creyó más conveniente y sin acuerdos previos de cómo llevar adelante la agrupación, con el único objetivo de agrupar tweets de temas iguales en un mismo cluster.

Estos conjuntos clusterizados de forma manual se utilizan como Gold Standard para llevar a cabo las validaciones externas.

Capítulo 4

Clustering aplicado a tweets

En este capítulo se presentan todas las etapas del proceso de clustering junto con los detalles necesarios para comprender cada una de ellas y así poder evaluar el desempeño de dos algoritmos, K-Means y DBSCAN. Además se expone una sección con los experimentos que se realizaron y cuyo análisis de resultados y evaluación se describe en el siguiente capítulo.

Para poder llevar a cabo todo el proceso surgen ciertas necesidades, siendo un ejemplo fundamental la creación de una línea base. A través de ella es que se logra comparar los resultados obtenidos en cada experimento realizado.

Una vez determinada la línea base se describe la solución propuesta. Se incluye una etapa de preprocesamiento con diferentes alternativas que intentan reducir los malos resultados generados por aquellas características que se identifican como ruido. También se evalúan algunas posibles formas de representar los tweets al igual que formas para expandir el significado de un tweet.

También existe una etapa de ajuste de parámetros de los algoritmos que permite mejorar la inicialización y ejecución de los algoritmos. Finalmente, para obtener aún mejores resultados se torna necesaria la introducción de una etapa de postprocesamiento de los resultados. Se describen sobre el final del capítulo los experimentos realizados con el fin de conocer cómo afectan los filtros, parámetros, expansiones y ajustes de los algoritmos al clustering de tweets.

En resumen, este capítulo detalla el procedimiento realizado para lograr clusterizar conjuntos de tweets así como los experimentos contemplados para conocer los resultados.

La figura 4.1 muestra las etapas a alto nivel del proceso que se lleva a cabo para lograr el clustering. Se comienza por el corpus generado, se pasa por las etapas de preprocesamiento y expansión que se aplica a cada tweet de manera individual. Luego se representan estos "nuevos tweets", se calculan los parámetros de inicialización de los algoritmos y se clusteriza. Finalmente se postprocesa cada tweet y se obtienen los clusters finales.

Para validar se utilizan los corpus clusterizados manualmente. Estos conjuntos tienen diferentes propósitos. Por un lado, interesa conocer los valores de los índices para el corpus de "entrenamiento" (Luis Lacalle) y de este modo conocer el grado de mejora que

tienen los algoritmos en todo momento. Por otro lado, interesa conocer el valor de los índices para el corpus de Luis Suárez dado que es el utilizado para evaluar los resultados finales. También interesa conocer el comportamiento para el clustering de Edinson Cavani ya que es la instancia existente para poder comparar empíricamente clusters obtenidos manualmente por personas diferentes de manera independiente.

El término "entrenar" en este documento se refiere al proceso de mejora del algoritmo de clustering y no al concepto de entrenar asociado al aprendizaje supervisado. Esto significa que a modo de ir mejorando el algoritmo que realiza el clustering, se usa un conjunto de datos (Luis Lacalle) con el cual poder ir realizando validaciones tanto internas como externas y así comparar con la línea base su progreso.

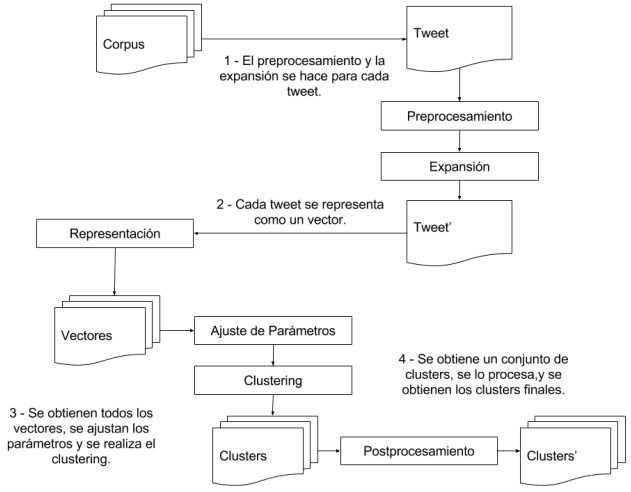


Figura 4.1: Proceso por el que pasa un corpus para ser clusterizado.

4.1. Línea base

Dado que se busca evaluar los algoritmos K-Means y DBSCAN, se necesita una línea base contra la cual comparar.

Twitter propone dentro de sus diferentes funcionalidades un concepto muy simple para permitir establecer el tema de un tweet. Tal como se define en la guía de ayuda de Twitter la forma de indexar o establecer el tópico de un tweet es anteponiendo el símbolo de numeral (#) a una palabra [Twitter, 2017]. Se establece como motivación para su uso la posibilidad de permitir a los usuarios seguir fácilmente sus temas de interés a través de esta etiqueta.

De esta forma se puede concebir el "#" como un identificador de los tópicos y por esta razón se vuelve razonable utilizar dicha convención para identificar el tema y así lograr agrupar tweets que hablen de lo mismo.

La principal característica de las etiquetas es que son palabras (o conjuntos de palabras concatenadas) que comienzan con el símbolo numeral y pueden ser utilizadas en

cualquier parte de un tweet y en reiteradas ocasiones. También es posible utilizar múltiples etiquetas por tweet, así como ninguna.

Por las razones anteriores, se decide construir una línea base (LB) donde los atributos considerados sean las etiquetas del tweet. Donde cada tweet se modela como un vector de largo la cantidad de hashtags en el corpus, y cada posición representa la presencia o no de una etiqueta en dicho tweet. Se considera a continuación un ejemplo.

Ejemplo 3. *Sea un corpus conformado solamente por los siguientes dos tweets.*

1. *"RT @fpscremini: ¡Felicitaciones al padrino de los #**valientes!** Segunda #**BotaDeOro** @LuisSuarez9 un campeón!"*
2. *"RT @Uruguay: #**CHIURU** | Con el encuentro de hoy, ante @LaRoja, @LuisSuarez9 llegó a 90 partidos con la selección mayor de @Uruguay. https://..."*

Para los tweets presentes en el corpus del ejemplo 3 las representaciones son $[1, 1, 0]$ y $[0, 0, 1]$ respectivamente, y donde además se cumple que el vector de atributos es de la forma [#valientes , #BotaDeOro, #CHIURU].

Con la representación anterior se espera que se generen dos clusters. Uno que identifique la disputa del partido entre Uruguay y Chile donde Suárez alcanza sus 90 partidos con la selección y por otro lado el logro de una bota de oro por parte de Luis Suárez.

Siguiendo esta lógica con los tweets de los corpus de Luis Lacalle, Edinson Cavani y Luis Suárez se procede a generar los clusters correspondientes. Posteriormente se analizan los resultados con el índice de validación interno Silhouette (IS) y el índice de validación externo Adjusted Rand Index (ARI) utilizando la distancia coseno. La tabla 4.1, presenta los resultados de estos índices y la cantidad de clusters (#C) para distintos corpus y algoritmos, donde los valores indicados en negrita señalan los mejores resultados para cada índice en cada uno de los corpus. Se computan los resultados para cada uno de los corpus, sin embargo, el único resultado que nos interesa en esta etapa es el referido a Luis Suárez ya que es el corpus designado para la evaluación. El resto de los corpus se usan en diferentes instancias. Como ya se comentaba, el de Edinson Cavani es usado para clusterizar manualmente y el de Luis Lacalle para ir mejorando el algoritmo.

De los resultados anteriores (tabla 4.1) se desprenden las siguientes observaciones:

1. Para el caso de K-Means conocer K de antemano es sencillo ya que por la representación elegida es la cantidad de combinaciones distintas de hashtags presentes en el corpus.
2. La cantidad de clusters detectados por DBSCAN es muy inferior a la estimada para K-Means. Esto se debe al hecho de que DBSCAN detecta a muchos tweets como ruido.
3. Índice ARI bajo e incluso negativo, esto muestra que los clusters generados en comparación con los clusters obtenidos de forma manual son extremadamente distintos. Un índice de valor 0.0 es comparable con realizar clustering aleatoriamente.

	K-Means			DBSCAN		
Corpus	**#C**	**IS**	**ARI**	**#C**	**IS**	**ARI**
Luis Suárez	14	**0.124**	**0.078**	2	0.068	0.074
Edinson Cavani (con validación de A1)	30	**0.314**	**0.108**	4	0.148	0.104
Edinson Cavani (con validación de A2)	30	**0.314**	0.074	4	0.148	**0.085**
Luis Lacalle	4	**0.000**	-0.017	1	**0.000**	**-0.016**

Cuadro 4.1: Validación de clusters generados con la línea base de hashtags.

4. El índice de Silhouette presenta distintos valores. Sin embargo esta medida está estrechamente relacionada a la representación de un tweet y cómo se mide la distancia. En este caso esa relación genera falsos positivos ya que como todos los tweets se representan con un vector que indica la presencia de un hashtag o no, y la distancia se mide comparando el ángulo entre estos vectores, dos tweets con distintos temas y mismos hashtags son agrupados en el mismo cluster y su ángulo es cero.

5. Se aprecia una variación en el índice ARI para el conjunto de Edinson Cavani entre un conjunto de validación generado por una persona (A1) y el generado por la otra (A2). Esto refleja como personas distintas pueden agrupar los mismos elementos de diferentes maneras.

6. Al realizar un análisis cualitativo de los clusters generados, rápidamente se llega a la conclusión que son muy malos, nada parece coincidir.

Para ejemplificar y explicar por qué ocurren los resultados anteriores se presenta el siguiente ejemplo.

Ejemplo 4. *Corpus conformado por dos tweets.*

1. *"@ECavaniOfficial feliz cumpleaños #PSGFCB"*

2. *"🎙 @ECavaniOfficial : "Jugamos 4 torneos en el año. Estamos concentrados en cada objetivo". #PSGFCB"*

Siguiendo la lógica anterior, al realizar el clustering, ambos tweets terminarían en un mismo cluster debido a que ambos poseen solamente una etiqueta y resulta ser la misma.

Si ahora, además, se analizan los contenidos de los tweets es visible que ambos tienen temas centrales muy diferentes, por lo tanto el cluster que se obtiene no es representativo. Este problema se debe principalmente al "mal" uso que se le da a las etiquetas o como en este caso un uso demasiado genérico.

Esta no es la única situación en la cual el clustering por hashtags tiene carencias, lo mismo sucede cuando los tweets no poseen hashtags, en ese caso, dado que el tweet no tiene etiquetas, el vector representante de dichos tweets es el vector con atributos vacíos ([0 , 0 , ... , 0]). Sin más, todos estos tweets son parte de un mismo cluster donde erróneamente intentan definir un tema.

Por estas razones, es necesario establecer una nueva línea base que sea más representativa del conjunto de tweets.

El objetivo de esta nueva línea base es intentar eludir los problemas presentes en la línea base anterior y construir un punto de comparación más preciso. Para ello se propone un nuevo enfoque del estilo Bag of Words (BOW) en donde los atributos no sean los hashtags sino las palabras que componen al tweet. Además, existen tweets que poseen links, considerando el trabajo realizado en [Dela Rosa, 2011], se decide quitarlos y reducir el posible ruido que estos generan.

Otras palabras que se omiten en la generación de atributos son las stopwords. Al igual que ocurre en otros proyectos que trabajan con lenguaje natural, se opta por remover aquellas palabras que no tienen contenido semántico. En este caso, se utiliza el conjunto de palabras neutras de NLTK en combinación con stopwords específicas a nuestro corpus que fueron obtenidas analizando los tweets y las frecuencias de las palabras presentes. Por detalles de estas palabras, consultar el apéndice D.

Por último, existe una tercer mejora en la generación de la línea base donde se convierten las palabras a su equivalente en minúsculas. Esto permite medir distancias entre palabras de forma consistente sin considerar mayúsculas y minúsculas.

Con estos cambios, se vuelve a computar los resultados para los índices y de este modo efectivamente utilizarlos como referencia para lo que vendrá (véase tabla 4.2).

En este caso la cantidad de clusters óptima no es conocida de antemano y por lo tanto en el caso de K-Means se utiliza una técnica de maximización del índice de Silhouette que se introduce con más detalle en la sección de ajuste de parámetros.

	K-Means			**DBSCAN**		
Corpus	**#C**	**IS**	**ARI**	**#C**	**IS**	**ARI**
Luis Suárez	12	**0.386**	**0.494**	3	0.270	0.361
Edinson Cavani (con validación de A1)	16	**0.390**	**0.556**	6	0.287	0.107
Edinson Cavani (con validación de A2)	16	**0.390**	**0.525**	5	0.269	0.192
Luis Lacalle	9	**0.401**	**0.598**	4	0.309	0.362

Cuadro 4.2: Validación de clusters generados con la línea base BOW.

De los resultados obtenidos (tabla 4.2) se extraen las siguientes conclusiones:

1. El número de clusters para ambos algoritmos es diferente a los valores en la línea base anterior, en el caso de K-Means porque fue optimizado y esto mejora el clustering en general. Con DBSCAN la diferencia se debe solamente a la nueva forma de representar los tweets que permite detectan por parte del algoritmo otras agrupaciones.

2. Para el caso de K-Means el índice ARI aumenta significativamente, esto se atribuye a que la solución generada se aproxima mejor a la realizada de forma manual y por lo tanto los clusters son más similares. Con DBSCAN se logra una mejora menos notoria.

3. Si se analizan los clusters obtenidos, los resultados presentan una gran mejoría en comparación con la línea base anterior. De todos modos, existen casos que dan lugar a mejoras.

Obtenida la línea base, la cual se utilizará como punto de comparación, se prosigue a presentar la solución desarrollada. Se intentará a partir de la combinación de diferentes técnicas lograr clusterizar tweets de tal manera que se reconozca con mayor precisión el sentido de pertenencia de un tweet a un cluster.

4.2. Preprocesamiento

Se considera necesaria una fase de preprocesamiento de los tweets. Estos poseen información muy relevante que debe ser catalogada y explotada previo a la etapa de clustering. Esta etapa se construye en base a un pipeline de filtros aplicables a los tweets del corpus. Vale mencionar que se decide reutilizar la transformación a minúsculas y la eliminación de URLs que se mencionaron en la construcción de la línea base. Esta etapa conforma un paso esencial para agregar valor, contexto y eliminar ambigüedad existente en la naturaleza de los tweets. También se puede encontrar un complemento a esta sección en el apéndice A donde se presentan más detalles.

Tokenizado

El paso previo al filtrado de tweets, consiste en convertir el texto del tweet a su forma en tokens. Para ello se utiliza la librería de NLTK TweetTokenizer que provee la capacidad de trabajar adecuadamente con el dominio de tweets. Esto lo que genera es tokenizaciones que contemplan casos particulares y típicos en el contexto de Twitter.

A continuación se presenta un ejemplo (figura 4.2) de los tokens que se obtienen para un caso particular:

Extracción de menciones y etiquetas

Dado un conjunto de tweets a ser clusterizados se eliminan las menciones y etiquetas en cada uno de ellos y además se almacena esta información de forma resumida en una matriz de ocurrencias para su posterior uso.

```
>>> from nltk.tokenize import TweetTokenizer
>>> tknzr = TweetTokenizer()
>>> s0 = "This is a cooool #dummysmiley: :-) :-P <3 and some arrows < > -> <--"
>>> tknzr.tokenize(s0)
['This', 'is', 'a', 'cooool', '#dummysmiley', ':', ':-)', ':-P', '<3', 'and', 'some', 'arrows', '<', '>', '->', '<--']
```

Figura 4.2: Ejemplo recuperado de `http://www.nltk.org/api/nltk.tokenize.html` el día 30 de abril de 2017.

Eliminación de Abreviaciones

Por la naturaleza de los mensajes en Twitter (solo 140 caracteres) es habitual observar mensajes con abreviaciones de muchas palabras. En su mayoría suelen ser palabras que no aportan valor al objetivo del clustering. Por ese motivo se diseña un filtro capaz de eliminar abreviaturas del estilo de "xq" (porque), "d" (de), "q" (qué), "hrs" (horas), entre otras.

Eliminación de Emojis

En la web se suelen utilizar muchos emojis (emoticones o íconos) para expresar sentimientos o incluso frases. Se decidió eliminar estos elementos ya que el análisis de sus significados en una frase desde el punto de vista emocional o de sus significados excede el alcance de este trabajo. Por este motivo, se eliminan tanto emoticones simples ";)", ":)" como emoticones UNICODE del estilo: "🙂😊".

El proceso de eliminar los emojis fue realizado en varias etapas debido a la dificultad que presenta esta tarea y a que una nueva versión de emojis fuera lanzada durante el desarrollo. Actualmente se tiene en cuenta la quinta versión de los Emojis UNICODE que incluye nuevos emoticones y nuevos códigos.

Eliminación de risa

Otro patrón notorio presente en los tweets es la utilización de expresiones de risa. En este contexto, son expresiones que no agregan valor, sino ruido. Expresiones como por ejemplo "jajaja", "jijij", "jeje", etc. son también eliminadas.

Eliminación de números

Los números (escritos con dígitos arábigos, "1", "2", "23", etc) también se eliminan en el preprocesamiento ya que su aparición puede generar agrupaciones que no necesariamente refieren a lo mismo.

Eliminación de palabras cortadas

Se descubre que Twitter en aquellos tweets que son re-tweeteados agrega el prefijo "RT " superando en algunos casos los 140 caracteres. Por este motivo, Twitter recorta los sobrantes caracteres agregando "..." al final. Esto genera palabras incompletas que

terminan resultando en ruido para un tweet. Por este motivo se decide eliminar las palabras que fueron recortadas por Twitter.

Eliminación de puntuación

También son eliminados en el proceso de limpieza todos los símbolos no pertenecientes al abecedario, como por ejemplo “ %”, “$”, “!”, etc.

Eliminación de RTs

En trabajos relacionados se vio como la eliminación de re-tweets puede afectar de manera positiva el análisis de datos. Por este motivo se considera la posibilidad de retirarlos y así comparar el efecto que esto produce. Dado que nuestro objetivo es clusterizar por tema, un re-tweet parece ser de gran valor para determinar la importancia de un tópico en relación a otros temas. De este modo, evaluar la eliminación o no de re-tweets es un factor a tener en cuenta.

Stopwords específicas del corpus

El conjunto de stopwords originalmente se conformaba por el set provisto por NLTK para el idioma español. Dado que el contexto de Twitter es un tanto especial por el hecho de que se encuentran muchas faltas ortográficas y NLTK construye su lista con términos ortográficamente correctos, se decidió construir una lista de stopwords personalizada. Como base para las decisiones de las palabras a incluir, se utilizó un artículo de la editorial Rubio [Rubio, 2015] que lista los errores ortográficos más frecuentes. Con esta base, sumada a un análisis del contexto de nuestro corpus se construyó una lista de stopwords que contiene tanto la lista de NLTK así como aquellas palabras de mayor frecuencia en los tweets del corpus. Consultar apéndice D.

Colapsar vocales repetidas

En el corpus es notoria la cantidad de palabras escritas con muchas vocales repetidas para intentar acentuar alguna emoción, o en el caso de los tweets sobre futbolistas para para indicar un gol. Algunos ejemplos son “goooooool” y “vamoooo”. Por este motivo se hace necesaria la aplicación de una función que colapse las repeticiones de vocales a una única ocurrencia.

4.3. Expansión de tweets

La expansión de tweets hace referencia al proceso de agregar o reemplazar atributos o características a un tweet dado. Esta técnica intenta generalizar los tweets de alguna manera volviéndolos mas fácil de agrupar. Es importante descatar que los tweets son expandidos previo al proceso de la construcción de la representación, es decir los tweets siguen siendo tweets pero con las palabras modificadas o con más palabras.

Visualmente la expansión se puede ver como un aumento del "área" que representa un tweet, haciendo que dos tweets no tan cercanos puedan solaparse.

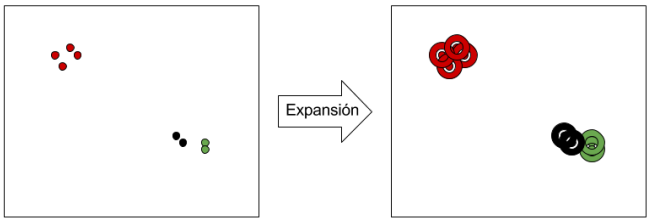


Figura 4.3: En la imagen, previo a la expansión se tienen tres clusters, pero luego de la expansión se podría decir que existen solo dos.

A continuación se listan algunos de los métodos creados y utilizados para realizar la expansión con el fin de mejorar la representación de los tweets.

4.3.1. Expansión con lemas

El método se compone de dos pasos bien definidos. El primero consiste en determinar si cada una de las palabras que componen el tweet existe en la base de léxicos de WordNet. El objetivo de esto es comprobar si se tratan de palabras para las cuales WordNet tiene identificado definiciones y sinónimos, o palabras malformadas o con errores tipográficos. En caso de aquellas palabras donde no se la reconoce, se prosigue a una segunda etapa de análisis. Este paso consiste en analizar morfosintácticamente la palabra con Freeling, luego de este análisis existen dos posibles resultados. El primero, en caso de resultar en un análisis exitoso, la expansión de la palabra consiste en sustituir el término original por el lema resultante, esto asegura que todos los tweets que contengan palabras con el mismo lema tengan el mismo atributo a la hora de clusterizar. Por otro lado, en caso de que el análisis no resulte exitoso, se conserva la palabra original dado que puede ser un tema o palabra clave dentro del corpus y no se desea perder esa información.

A modo de ejemplo se presentan tweets en su versión original y expandida (ejemplo 5)

Ejemplo 5. *Expansión por lemas*

Original: *"les faltó la pulga y @luissuarez9 , @neymarjr no pudo **dar** en ancho para el partido 😐😐⚽⚽ https://t.co/gr9bb5q2hu"*

Expandido: *"les faltar la pulga y @luissuarez9 , @neymarjr no poder **dar** en ancho para el partido 😐😐⚽⚽ https://t.co/gr9bb5q2hu"*

Original: *"el día que leo messi le **dio** una asistencia de **chilena** a @luissuarez9... #golazosportags. https://t.co/rpxwk0hpwt"*

Expandido: *"el día que leo messi le* ***dar*** *una asistencia de* ***chileno*** *a @luissuarez9 ... #golazosportags . https://t.co/rpxwk0hpwt"*

Original: *"aquel día que leo messi le* ***dió*** *una asistencia de* ***chilena*** *a @luissuarez9... https://t.co/rpxwk0hpwt"*

Expandido: *"aquel día en que leo messi le* ***dar*** *una asistencia de* ***chileno*** *a @luissuarez9 ... https://t.co/rpxwk0hpwt"*

En estos casos particulares, además de notarse las palabras llevadas a su forma de lema, se destacan dos situaciones típicas de esta transformación. Por un lado la palabra "dio" / "dió" es transformada a su lema "dar" la cual pasa a ser un atributo común a los tres tweets. Si quitamos dicha expansión los tres tweets poseerían tres atributos diferentes los cuales se tratarían como diferentes en el proceso de clustering.

Por otro lado, si ahora se considera la palabra "chilena" presente en los últimos dos tweets. Ocurre algo inesperado, y es que la transformación lleva de una palabra que posee un significado a otra completamente distinta. Si consideramos un cuarto tweet que haga mención a alguna persona de nacionalidad "chileno" se podría estar entrelazando temas que tal vez no correspondan en un mismo cluster.

Por esta razón el uso de este método puede llegar a no ser lo suficientemente ventajoso dependiendo del contexto y del corpus.

4.3.2. Expansión con representaciones vectoriales de las palabras

Debido al poco contenido de un tweet, muchas veces resulta difícil determinar si dos tweets tratan un mismo tema. Para intentar resolver este problema se intenta aplicar un método de expansión diferente al anterior el cual no se agregan palabras, sino que generalizan las existentes. En este caso se busca agregar palabras similares al contenido de un tweet para obtener su versión expandida.

Para lograr este objetivo se entrenó un modelo vectorial de palabras sobre un corpus en español de 1 billon de palabras [Cardellino, 2016] utilizando el algoritmo Word2Vec. La ejecución del entrenamiento se realizó en el Cluster FING, la plataforma de Computación Científica de Alto Desempeño de la Universidad de la República, Uruguay [Nesmachnow, 2010].

El proceso de expansión consiste en agregar al tweet la palabra más cercana según el modelo para cada palabra del tweet.

Ejemplo 6. *Expansión por Word2vec*

Original: *"Mesa política del FA respalda a Maduro. Gobierno para un lado, fuerza política para otro. Así estamos! @SenJavierGarcia @LuisLacallePou"*

Expandido: *"Mesa política del FA respalda a Maduro. Gobierno para un lado, fuerza política para otro. Así estamos! @SenJavierGarcia @LuisLacallePou* ***tribuna estrategia apoya simpático mandatario extremo unidad"***

Para el ejemplo anterior se tiene que las relaciones de proximidad entre palabras son las presentes en la tabla 4.3. Al momento de realizar la expansión solo se considera la palabra más cercana obteniendo así la versión expandida.

Palabra	**Palabras más cercanas**
mesa	tribuna, terna, comisión, tarima, junta, sala
política	estrategia, modernizadora, ideológica, institucionalidad, doctrinaria, democracia
respalda	apoya, respaldará, apoyamos, respaldó, respaldamos, suscribe
maduro	simpático, hiperactivo, comedido, carismático, cauteloso, comprensivo
gobierno	mandatario, gobierno, parlamento, gobernador, govern, gabinete
lado	extremo, borde, flanco, pasillo, contrafuerte, margen
fuerza	unidad, resistencia, potencia, milicia, tropa, autoridad

Cuadro 4.3: Relaciones de proximidad.

En este caso ocurre que la palabra "Maduro" es expandida con la palabra "simpático" lo cual agrega una clase distinta de palabra al tweet pudiendo potencialmente agrupar este tweet con otro de distinto tema. Un análisis morfosintáctico del tweet podría reducir la cantidad de ocurrencias de estos casos.

También es importante destacar que debido a que la representación de los tweets no tiene en cuenta el orden de las palabras, agregar los nuevos atributos al final de la representación es una decisión arbitraria que no posee efectos secundarios indeseados.

4.3.3. Expansión con raíces

La expansión por raíces consiste en el procesamiento de cada una de las palabras del tweet llevándola a su forma raíz. Esto permitiría entre otras cosas detectar palabras fuertemente relacionadas por su significado. El motivo parece prometedor, sin embargo esto no es siempre correcto. Consideremos los siguientes ejemplos.

Ejemplo 7. *Expansión por raices 1*

Original: *"@luissuarez9 sos un **grande** aqui en uruguay te **queremos** **mucho** no **cambien** nunca tu **humilda besos**"*

Expandido: *"@luissuarez9 sos un **grand** aqui en uruguay te **quer much** no **cambi nunc** tu **humild bes**"*

En este caso particular, palabras que suelen ocurrir con frecuencia como "queremos", "querido", "quería", etc. resultan en una misma raíz: "quer". Esto es algo positivo para el proyecto dado que dichas palabras implican en cierto sentido un mismo sentimiento. Existen otras situaciones donde la sustitución genera que un tweet difiera más a otro. Consideremos el siguiente ejemplo:

Ejemplo 8. *Expansión por raices 2*

*Original: "@luissuarez9 ya q el es un gran **fanatico** la **tematica** de su **cumple** se **trata** de ti y **seria** un **sueño cumplido para** el si **pudieras saludarlo**"*

*Expandido: "@luissuarez9 ya q el es un gran **fanat** la **temat** de su **cumpl** se **trat** de ti y **seri** un **sueñ cumpl par** el si **pud salud**"*

Observando el caso expandido del tweet, se puede ver como la palabra "saludarlo" se encuentra reducida a la palabra "salud". Lo mismo ocurriría si en vez de "saludarlo" tuviéramos "saludar", "saludamos", "saludé" y algunas otras. Esto se transforma en un efecto indeseado de la transformación raíz, ya que si ahora tuviéramos un caso de un tweet que hable sobre la salud de alguien, por ejemplo supongamos que Suárez sufrió alguna lesión en un encuentro de fútbol previo a su cumpleaños y sus seguidores hablaran de que "son fanáticos y que quisieran que recupere su salud porque es un gran jugador", el contexto previo se pierde y con un gran grado de certeza ambos tweets formarían parte del mismo cluster de manera errónea.

4.4. Representación de los Tweets

Para representar los tweets en un modo amigable para el procesamiento computacional se utiliza un vector de atributos. Cada tweet se compone de una lista de atributos ordenados que representan la entidad original. Esta lista de atributos está compuesta por valores ordenados en forma secuencial que oscilan en el intervalo $[0, 1]$. A continuación se presentan las tres variaciones consideradas. La lista de atributos conformada solamente por palabras, la lista conformada por la unión de palabras, etiquetas y menciones de un tweet y un último caso donde se considera la versión anterior asignando pesos a cada uno de los atributos.

Atributos de palabras

Para generar esta representación una vez hecho el preprocesamiento se procede a analizar la frecuencia de cada palabra en cada tweet obteniendo así una medida numérica que expresa cuán relevante es una palabra para un tweet en el corpus. Esta medida se conoce por sus siglas en inglés como TF-IDF (Term frequency – Inverse document frequency). De esta forma se logra representar a cada tweet como un vector de atributos donde cada coordenada representa la importancia de una palabra y no tiene en cuenta su posición relativa en el documento, así se obtiene para cada tweet un vector de la forma $\omega = (\omega_0, ..., \omega_n)$ que lo representa.

Atributos de palabras, etiquetas y menciones

Una segunda forma utilizada para representar los tweets se basa en palabras y etiquetas de los tweets. Esta representación a diferencia de la anterior intenta explotar algunos datos recolectados en la etapa de preprocesamiento que no se utilizan en la representación anterior como por ejemplo las ocurrencias de menciones en cada tweet. Para tener en cuenta estos datos se aplica la misma lógica de conteo aplicado a las palabras. Se obtiene así un vector $\theta = (\theta_0, ..., \theta_m)$ que representa las menciones en un tweet y otro vector similar $\eta = (\eta_0, ..., \eta_k)$ que representa los hashtags.

Estos dos vectores θ y η se utilizan de forma tal de que se genera un tercer vector t de la forma $t = [(\omega_0, ..., \omega_n), (\theta_0, ..., \theta_m), (\eta_0, ..., \eta_k)]$ que se compone de la concatenación del vector de atributos de palabras con el vector de atributos de menciones y de etiquetas. Esta nueva forma de representar agrega información extra que la representación anterior no consideraba.

Atributos de palabras, etiquetas y menciones con pesos

La tercera representación se basa fuertemente en la representación basada en atributos de palabras, etiquetas y menciones con el agregado de pesos a las variables. La gran diferencia se debe al agregado de pesos α, β y γ a cada tipo de coordenada (mención, etiqueta o palabra) de forma tal que aporten diferente importancia en la representación global del tweet. El vector resultante queda de la forma:

$$t = (\alpha * \omega_0, ..., \alpha * \omega_n, (\beta * \theta_0, ..., \beta * \theta_m), (\gamma * \eta_0, ..., \gamma * \eta_k))$$

.

4.5. Ajustes de parámetros

Los algoritmos seleccionados requieren la configuración de parámetros iniciales que deben ser determinados previo al proceso de clustering en sí, y por este motivo se dedica la presente sección a su ajuste. Existen varias estrategias para su selección. Considerando que el objetivo final es lograr un mejor resultado, la selección aleatoria se vuelve una decisión poco segura. Por este motivo se opta por la implementación de algunos algoritmos que faciliten este cálculo.

Antes de entrar en detalle, es importante destacar que estos cálculos no aplican para las líneas bases. En ellas se consideraron los valores por defecto para ambos algoritmos. Además, una decisión importante tomada en esta etapa del proyecto es que una agrupación de tweets se convierte en cluster sí y sólo sí posee tres o más elementos.

K-Means

En el caso de K-Means el único parámetro a determinar es K, es decir cuántos clusters debe generar el algoritmo para lograr el resultado óptimo. Dado que el corpus en general varía y no se conocen sus elementos en detalle, se debe encontrar un método

capaz de determinar K bajo condiciones a priori desconocidas. Para esto una posible técnica consiste en maximizar el índice de Silhouette que mide la calidad interna de los clusters. Esto presenta un problema visible y es que las métricas de validación internas suelen mejorar a medida que K aumenta. Porque a mayor cantidad de clusters la correspondencia cluster-tweet se vuelve más directa. Una consecuencia de esto es la reducción de los tamaños de los clusters. Para resolver este problema se debe buscar un k tal que $k+1$ no mejora la métrica m de forma significativa, es decir $m(k) < m(k+1) + \epsilon$. Un método conocido es el "Método del Codo" (Elbow method) que intenta determinar el punto donde se cumple que las mejoras no aumentan significativamente.

A continuación se muestra un ejemplo de cómo se lleva a cabo el cálculo de K para un corpus de 165 tweets. La gráfica representa el método del codo, y donde se aprecia que el óptimo está dado cuando la cantidad de clusters es de seis. (véase figura 4.4)

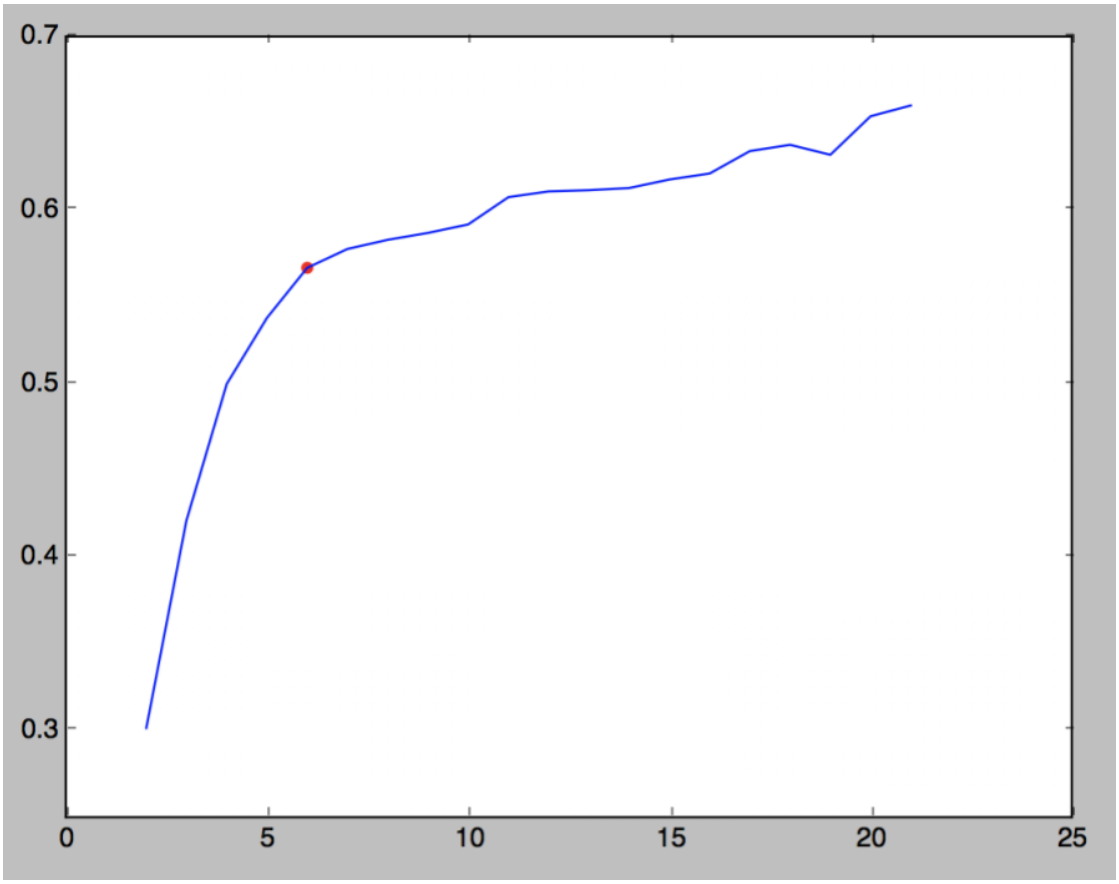


Figura 4.4: Ejemplo de gráfica representativa al método del codo para 165 tweets

DBSCAN

En el caso de DBSCAN los parámetros a ajustar son ϵ, la distancia máxima entre dos elementos para que se consideren en el mismo vecindario, y la mínima cantidad de elementos para considerar un elemento como centroide, $min\_samples$. Para la estimación de estos parámetros los autores de DBSCAN presentan junto al algoritmo una heurística para poder aproximarlos y que permite obtener la partición más "fina" de los datos. Es decir el conjunto de agrupaciones más chicas posibles. En caso que se quieran obtener agrupaciones con otras características se deben ajustar los parámetros según las necesidades.

Dado k se define una distancia *k-dist* que representa la distancia desde un punto p al *k-ésimo* vecino más cercano. Si se ordenan las distancias en orden descendente se obtiene una gráfica que representa de forma cercana la densidad presente en los datos. Si se elige un tweet arbitrario t y se fija $\epsilon = k\text{-}dist(t)$ y $min\_samples = k$ entonces todos los puntos con menor distancia que ϵ serán centros. Dado que en este caso se estableció $k = 3$, si tomamos el punto codo de la función *3-dist* se obtienen parámetros que permiten obtener los clusters más "finos" donde todos cumplen que cada uno tiene al menos tres tweets.

En la gráfica 4.5 se muestra la representación de la función *3-dist* para un conjunto de 513 tweets y donde el valor estimado de ϵ se encuentra en 0.022.

4.6. Post-procesamiento

Esta etapa surge como una necesidad para realizar validaciones más precisas y equitativas. El objetivo es aplicar esta fase sobre los resultados obtenidos para el algoritmo K-Means, no así sobre DBSCAN.

Por un lado, en la sección de ajuste de parámetros (4.5.1), se toma la decisión de considerar como cluster válido a aquellas agrupaciones de al menos tres elementos. Esto, si se considera el funcionamiento de DBSCAN, es logrado de manera automática al ajustar el parámetro que infiere la cantidad mínima de elementos en un cluster. Sin embargo, no sucede lo mismo para K-Means el cual asigna todo elemento a algún cluster incluso generando clusters de un solo elemento.

Por otro lado, en la validación externa, los clusters de menos de tres elementos se consideran ruido. Al validar el resultado de DBSCAN, la comparación es consistente, sin embargo no lo es para K-Means. Para contrastar este resultado indeseado, se agrega una etapa final de post-procesamiento al clustering con K-Means que consta de una heurística conformada por dos actividades diferentes.

Primero, se procesan los clusters resultantes y aquellos que no alcancen al menos tres tweets, se los categoriza como ruido, añadiéndolos al subgrupo del cluster ruido. El segundo paso, consiste en reajustar los clusters resultantes. Se reacomodan los clusters sobrevivientes enumerándolos de manera secuencial nuevamente.

De esta manera, el criterio de decisión de que un tweet es ruido es uniforme tanto para el clustering manual como para los resultados obtenidos de ejecutar K-Means y DBSCAN.

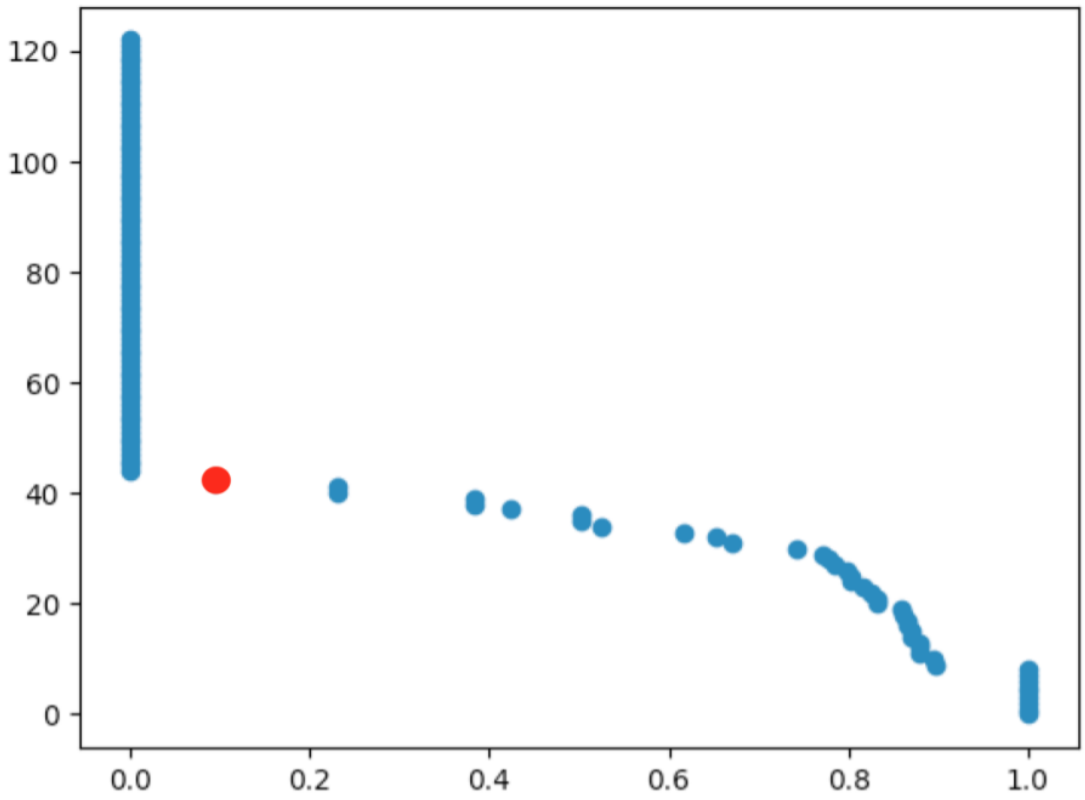


Figura 4.5: Captura de *3-dist* para 513 tweets

4.7. Experimentos

En este momento se tiene todo lo necesario para llevar a cabo el objetivo final de clusterizar tweets. Se generó una línea base interesante con la cual comparar los resultados, se presentó la etapa de preprocesamiento de los tweets, se introdujo el concepto de expansión de los tweets y la forma de representarlos, se dedicó una sub sección a detallar como se ajustaron los parámetos iniciales de los algoritmos y finalmente se describió el proceso de postprocesamiento realizado. En este punto lo que resta es llevar todo lo anterior a la práctica, para ello se decide presentar tres grandes experimentos que engloban las diferentes combinaciones de filtros, expansiones y preprocesamientos en los tweets. Como experimento extra se opta por realizar un cuarto experimento que combina las técnicas de los mejores resultados para los tres experimentos anteriores.

El objetivo de esta sección es presentar los diferentes experimentos sin entrar en detalles sobre los resultados obtenidos. Los resultados se analizan en el siguiente capítulo de análisis de resultados.

Es importante destacar que todos los experimentos se llevan a cabo para el corpus de Luis Suárez. Este corpus hasta el momento se encontraba reservado, es decir, nunca

fue utilizado previamente para "entrenar" los algoritmos.

4.7.1. Experimento 1: Características ruidosas

Este primer experimento es la base de los siguientes experimentos y supone la aplicación/quitado de los filtros presentados en la sección de preprocesamiento. Se considera un aspecto relevante poder conocer aquellos filtros que aportan mejoras así como aquellos que afectan negativamente al resultado del clustering. El objetivo de este experimento es combinar diferentes filtros así como también aplicarlos aisladamente. En la línea base se usan tres filtros básicos, la transformación de las palabras a su forma en minúsculas, la eliminación de stopwords y el quitado de URLs. Sin embargo se construyeron ocho filtros completamente independientes. Probar todas las combinaciones supone clusterizar 2^8 combinaciones, es decir 256 opciones por cada algoritmo, lo cual resulta en la totalidad de 512. Resulta lógico computar un número reducido de combinaciones ya que existen elementos de los tweets que son indeseados, tal es el caso de números, palabras cortadas, entre otros.

4.7.2. Experimento 2: Ponderación de atributos

El segundo experimento toma como punto de partida las condiciones que logran el mejor resultado para los dos algoritmos en simultáneo en el experimento anterior. A partir de allí, se decide ponderar de diferentes maneras los diferentes atributos que se introducen en la sección de representación de tweets de este capítulo. Es decir, se ponderan en diferentes grados las palabras, menciones y los hashtags que componen los tweets. Se prueban solo algunas combinaciones de pesos debido a que el universo de posibilidades es muy amplio. Entre estas combinaciones se mide darle solo importancia a las palabras, dividir la importancia entre palabras, hashtags y menciones de distintas maneras y no darle importancia a las palabras. Los valores posibles de los pesos son valores entre [0, 1] sin embargo se utilizan valores exactos por simplicidad. Posibles valores son 0.0, 0.10, 0.20, 0.25, 0.40, 0.50, 0.70 y 1.0.

4.7.3. Experimento 3: Expansiones

Al igual que en el experimento dos, para este experimento se toma como punto de partida el mejor resultado del primer experimento. Para esta evaluación se expanden los tweets con las palabras más cercanas a las palabras que conforman el tweet según algún criterio. El objetivo de este experimento es evaluar como el agregar significado al tweet mediante las expansiones definidas afecta los resultados. La expansión permite generalizar el contenido de un tweet y de esta manera facilitar la agrupación de tweets que por naturaleza utilizan palabras diferentes para hablar de los mismos temas.

4.7.4. Experimento 4: Combinación de experimentos previos

Los experimentos 2 y 3 toman como punto de partida el experimento 1. Esto implica que los resultados de los experimentos 2 y 3 son completamente independientes entre

sí. Es de interés conocer el comportamiento cuando se realiza clustering considerando el mejor resultado de los tres experimentos. Por este motivo surge la idea de considerar un cuarto experimento que una todo lo aprendido con los anteriores experimentos.

Capítulo 5

Análisis de resultados

Esta sección presenta dos tipos de análisis con diferentes enfoques. El primero de ellos tiene por cometido presentar los resultados obtenidos para los diferentes experimentos realizados. El segundo enfoque se centra en mostrar visualmente los resultados del clustering. Esto último, a pesar de ser un complemento para visualizar las agrupaciones logradas, agrega el valor de facilitar la separación entre clusters y notar imperfecciones o casos particulares no resueltos por el clustering final.

Se comienza presentando los resultados obtenidos para los experimentos anteriores para posteriormente finalizar el capítulo con los resultados gráficos logrados.

5.1. Evaluación de experimentos

Para evaluar los diferentes experimentos de manera empírica se recurre, al igual que como se realizó con la línea base, a validar interna y externamente cada resultado del clustering sobre el corpus reservado para la evaluación, el de Luis Suárez del día 6 de enero de 2017. Para ello se utilizan una vez más los índices de Silhouette y ARI con el objetivo de determinar cómo afecta cada etapa del proceso al clustering. Recordemos que el índice de Silhouette es una medida interna que sirve para conocer que tan correctamente los elementos de los clusters respetan la definición de clustering. Es decir, que tan similares son instancias de un mismo cluster, y que tan diferentes son instancias entre clusters. Por otro lado, el índice ARI, es una medida externa para indicar que tan bien agrupados están los elementos de un cluster en comparación con un Gold Standard calculado manualmente.

5.1.1. Experimento 1: Características ruidosas

La tabla 5.1 muestra los diferentes resultados obtenidos tras aplicar los diferentes filtros. La misma presenta el número de clusters (#C) calculado automáticamente, el índice de Silhouette (IS) y el Adjusted Rand Index (ARI) para cada algoritmo utilizado y para cada filtro aplicado.

	K-Means			DBSCAN		
Característica de ruido	#C	**IS**	**ARI**	#C	**IS**	**ARI**
Línea base	12	**0.386**	**0.494**	3	0.270	0.361
Abreviaciones	7	**0.364**	**0.459**	5	0.295	0.433
Emojis	8	**0.425**	**0.505**	5	0.295	0.433
Risa	9	**0.389**	**0.630**	5	0.295	0.433
Números	8	**0.436**	**0.482**	5	0.293	0.433
Palabras cortadas	8	**0.403**	**0.524**	5	0.292	0.433
Puntuación	7	**0.380**	**0.454**	5	0.293	0.433
URLs	9	**0.384**	**0.566**	5	0.282	0.413
Retweets	1	-	**0.084**	0	-	-
Colapsar vocales repetidas	8	**0.408**	**0.501**	5	0.293	0.433
Todas combinadas excepto retweets	9	**0.433**	**0.534**	5	0.316	0.465

Cuadro 5.1: Validación de clusters obtenidos para el algoritmo generado.

Se puede apreciar como la cantidad de clusters para K-Means es mayor a la obtenida en comparación con la línea base. También se puede observar como ambas evaluaciones, tanto la interna como la externa, presentan leves mejoras debido a la gran reducción de ruido presente en los tweets. En el caso de DBSCAN a pesar de observarse algunas mejoras, estas no son tan significativas como en el caso de K-Means. Se puede ver como K-Means en todas las pruebas realizadas los índices son más cercanos a 1. Esto teóricamente indica que los clusters son más precisos que los generados por DBSCAN.

Es importante destacar que en el caso que se eliminan los retweets para ambos algoritmos los resultados resultan ser muy malos. En ambos casos no se pueden calcular los índices de Silhouette dado que para K-Means la cantidad de clusters resultantes es 1 (más el cluster ruido) y para DBSCAN no hay cluster resultante, todo es ruido y el índice al menos debe contar con la existencia de dos clusters diferentes para comparar resultados. En cuanto al índice ARI para el cluster obtenido con K-Means, el resultado es extremadamente malo como se esperaba, dado que todos los tweets que no fueron considerados ruido están en un cluster de muchos temas. Dado que la eliminación de retweets reduce significativamente la calidad de los resultados, se decide no utilizarlo en el resto de las pruebas.

La prueba final que engloba aplicar todos los filtros excepto el de eliminación de retweets logra un resultado muy interesante. Analizando primero el caso de K-Means, si bien tanto IS como ARI no arrojan el resultado más cercano a 1 en relación a todas las pruebas, ambos son el segundo índice más alto y se encuentran sobre el valor promedio. En cuanto a DBSCAN, la prueba da notablemente el mejor resultado.

Como resultado general se tiene que aplicar todos los filtros existentes a excepción de eliminar re-tweets, logra el mejor resultado con ambos algoritmos. Por este motivo, para los siguientes experimentos se parte de la base que todos estos filtros son aplicados.

5.1.2. Experimento 2: Ponderación de atributos

Tomando como punto de partida el resultado del experimento previo, es decir la limpieza de características ruidosas con la salvedad de eliminar re-tweets, se realiza el segundo experimento de ponderar las palabras, los hashtags y las menciones de los tweets. De esta forma se asignan ciertos pesos α a las palabras, β a los hashtags y γ a las menciones para así poder determinar la importancia que tienen éstos en la representación de un tweet.

En la tabla 5.2 se muestran el número de clusters (#C) calculado automáticamente, el índice de Silhouette (IS) y el Adjusted Rand Index (ARI) para cada distribución de pesos asignada y cada algoritmo. En esta ocasión solo se resaltan los mejores resultados y no los mejores resultados relativos a cada experimento. Esto principalmente por la razón de ser el mismo experimento con la salvedad de los pesos asignados, donde conocer la mejor combinación de pesos es lo que importa.

	K-Means			**DBSCAN**		
Distribución de pesos (α, β, γ)	**#C**	**IS**	**ARI**	**#C**	**IS**	**ARI**
Línea base	12	0.386	0.494	3	0.270	0.361
(1.00, 0.00, 0.00)	7	0.375	0.499	5	0.316	**0.456**
(0.50, 0.25, 0.25)	8	0.396	**0.588**	5	0.281	0.413
(0.70, 0.20, 0.10)	8	0.374	0.557	5	0.281	0.413
(0.00, 0.50, 0.50)	8	**0.641**	0.486	7	**0.617**	0.385
(0.20, 0.40, 0.40)	9	0.431	0.539	5	0.279	0.413

Cuadro 5.2: Validación de clusters obtenidos para el algoritmo generado con ponderación de atributos. Donde $(\alpha, \beta, \gamma) = (palabras, hashtags, menciones)$.

Se prueban solo algunas combinaciones de ponderaciones debido a que el universo de posibilidades es muy amplio.

Observando la cuarta combinación en la cual no se le da importancia a las palabras, curiosamente se obtiene el valor más alto para el índice de Silhouette. Así mismo, la cantidad de clusters detectados coincide con la cantidad de clusters lograda en el conjunto de validación calculado de forma manual. Sin embargo el índice ARI se ve reducido en comparación al valor de referencia que se tenía del primer experimento. Al analizar los clusters de forma cualitativa, rápidamente se puede ver que los clusters son malos y no aglomeran temas fuertemente relacionados; por ese motivo el índice ARI resulta bajo.

5.1.3. Experimento 3: Expansiones

El experimento de expandir los tweets con nuevas palabras o modificaciones de las actuales resulta un experimento interesante. La tabla 5.3 presenta el mismo formato que las tablas anteriores y muestra resultados para los dos índices en el caso de las expansiones por lema, raíz, word2vec y una combinación de word2vec-lema.

	K-Means			**DBSCAN**		
Expansión	**#C**	**IS**	**ARI**	**#C**	**IS**	**ARI**
Línea base	12	0.386	0.494	3	0.270	0.361
Lema	20	0.420	0.537	5	**0.316**	**0.456**
Raíz	21	0.404	0.502	5	**0.316**	**0.456**
Word2Vec	10	0.350	**0.639**	5	**0.316**	**0.456**
Word2Vec & Lema	25	**0.427**	0.352	5	**0.316**	**0.456**

Cuadro 5.3: Resultados obtenidos para las diferentes expansiones de tweets.

Los resultados no parecen ser muy alentadores. Las diferentes expansiones obtienen resultados iguales o peores a los anteriores en ambos algoritmos para casi todos los casos. En el caso de la expansión con word2vec con el algoritmo K-Means parece haber una mejora interesante medida con el índice ARI al mismo tiempo que un bajo IS. Se presume que los problemas encontrados en la sección 4.3 son los causantes de estos resultados y por esta razón no es suficiente con expandir.

A pesar de los resultados, la estrategia de expansión es prometedora. Existen algunos trabajos previos, como los presentados en la sección de trabajos previos del marco teórico, que utilizados en combinación y considerando más variables generen mejores resultados.

5.1.4. Experimento 4: Combinación de experimentos previos

Analizando los resultados para los experimentos 2 y 3, se decide analizar el resultado de clusterizar el corpus de Luis Suárez considerando distribución por pesos y representación con expansión por simultáneo. Para ello se utiliza la distribución [0.50, 0.25, 0.25] dado que supone un muy buen resultado en el experimento 2 y también se considera la expansión por Word2Vec.

La tabla 5.4 muestra los resultados obtenidos para este experimento.

Para el caso de DBSCAN, los resultados son inferiores a cualquier otro obtenido previamente, lo cual indica que esta forma de representar y realizar clustering resulta desfavorable para este algoritmo. Por otro lado, cuando se trata de K-Means si bien no son los mejores resultados, tampoco son tan malos. Se tiene que el índice de Silhouette es mejor en la combinación de ambas estrategias que en cada una por separadas. Esto indica que los clusters cumplen mejor con la definición de cluster. Por otro lado la comparación con el clustering manual no es tan buena. Este resultado si bien no es excelente tampoco descarta que el uso de Word2Vec y dicha distribución por pesos al ejecutar K-Means sea una mala decisión.

	K-Means			DBSCAN		
	#C	**IS**	**ARI**	**#C**	**IS**	**ARI**
Línea base	12	0.386	0.494	3	0.270	0.361
Distribución (0.5, 0.25, 0.25)	8	0.396	0.588	5	0.281	0.413
Expansión Word2Vec	10	0.350	**0.639**	5	**0.316**	**0.456**
Word2Vec & distribución	8	**0.402**	0.523	5	0.275	0.413

Cuadro 5.4: Resultados obtenidos para las diferentes expansiones de tweets.

5.2. Visualización de los clusters

Poder observar los resultados obtenidos con la etapa anterior es un aspecto importante en el contexto de este proyecto. Si bien hasta el momento se realizó la evaluación de los clusters, se consideró necesario también una instancia para representarlos gráficamente. La misma se basa fuertemente en las técnicas utilizadas en el trabajo previo de [Godfrey, 2014] donde se logran representaciones en forma de nubes de palabras.

A continuación se plantean algunos de los aspectos considerados en la construcción, así como los resultados obtenidos.

Las imágenes que se ven a continuación son resultado de clusterizar el corpus de Luis Lacalle con K-Means. Se lo ejecuta para los resultados obtenidos en la línea base así como para la versión final lograda. La representación del resultado consiste en realizar un conteo de palabras considerando finalmente solo aquellas más significativas, es decir, las que se repiten en más ocasiones dentro de un cluster en particular. Todas las palabras tenidas en cuenta para la generación de la nube son parte de las palabras claves de cada cluster. Aquellas que se repiten en mayor medida son las que finalmente representan a los clusters obtenidos.

Línea base

Para la línea base, se realizaron dos análisis gráficos de los resultados. Primero, se muestran los resultados obtenidos al realizar el clustering con la línea base inicial. Dichos clusters se pueden ver en la figura 5.1.

Analizando el resultado, se ve como existe una cantidad considerable de ruido en cada uno de los clusters. Esto se debe principalmente a la falta de aplicación de filtros para conformar los atributos de un tweet. Proceso que se realiza posterior a la línea base. En la imagen se puede ver como el corpus termina agrupándose en tan solo cuatro clusters. Esto se debe a que se encuentra conformado por tan solo cuatro combinaciones de hashtags diferentes y en esta etapa, los atributos son únicamente los hashtags.

Como se menciona en el capítulo 4, cuando se habla de la generación de la línea base, se realiza una leve mejora a esta línea base inicial. Se aplicó un enfoque diferente, que no solo considera como atributos a los hashtags, sino que a todas las palabras que

Figura 5.1: Cuatro clusters en base a Etiquetas (hashtags). Correspondiente a la línea base en sus orígenes.

conforman el tweet. Vimos como la evaluación de esta versión mejorada de la línea base daba mejores resultados. Lo mismo ocurre con la visualización de los clusters como es de esperar.

En la figura 5.2 se aprecia como el número de clusters solamente difiere del conjunto de evaluación en una unidad. Sin embargo, el resultado aún no es claro. Si bien la cantidad de clusters casi coinciden, el contenido de los mismos es muy diferente. Se puede apreciar nuevamente una gran presencia de ruido que en muchos clusters no permite determinar el tema principal.

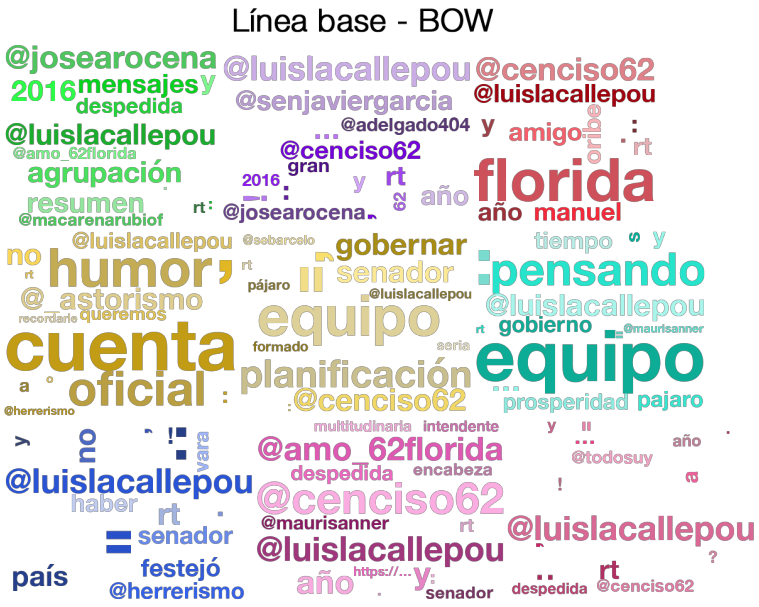


Figura 5.2: Nueve clusters resultantes para la línea base mejorada, utilizando K-Means.

Instancia de "Mejor resultado obtenido"

En el mejor caso obtenido (figura 5.3) la cantidad de clusters coincide casi exactamente con el número de clusters identificados al agrupar manualmente el conjunto de evaluación. Véase figura 5.4. Además una mejoría notoria es la posibilidad de determinar a grandes rasgos y de manera directa con solo mirar, el tema tratado en cada uno de los clusters.

Se puede ver como para el mismo corpus y algoritmos los resultados mejoran significativamente en comparación con la línea base. En la figura 5.1, los clusters logrados no permiten conocer el tópico central de cada cluster. De hecho, tres clusters son representados por un hashtag dado, y el restante son los 120 tweets restantes del corpus. En la figura 5.2 se nota como el algoritmo sin considerar stopwords y links permite lograr un mejor resultado. Se aprecian algunas palabras que dan un poco más de noción acerca del tópico que representan pero aún se nota como la línea base puede ser mejorada. Esta perspectiva cambia en la figura 5.3. En ella se pueden diferenciar más claramente los temas relevantes del corpus. Las nubes de palabras logradas no contienen palabras sin sentido y a la vez dan una noción mucho mayor de los temas hablados.

Para conocer en mayor detalle el resultado obtenido, se tiene a continuación una muestra de los tweets que conforman dos de los clusters obtenidos en esta última representación. Consideremos primero el cluster compuesto por las palabras: *[blancos, grandes, viva, gran, familia, nacional, partido, honran]*

- RT @JRadiccioni: @PabloLarraz10 @PNACIONAL @LuisLacallePou @jorgewla-

Figura 5.3: Muestra obtenida para el algoritmo mejorado. Diez clusters resultantes más el cluster ruido.

rranaga @nimcallonimvoy @Lista404ch VIVA!!!!

- @PabloLarraz10 @PNACIONAL @LuisLacallePou @jorgewlarranaga @nimcallonimvoy @Lista404ch VIVA!!!!
- @LuisLacallePou @CEnciso62 Dos grandes que honran la gran familia del partido nacional !! Viva los blancos ...

Los anteriores son los únicos tres tweets que hacen referencia a alentar o festejar ser parte del partido blanco en el corpus. En el mismo hay dos tweets que son su versión original y un re-tweet lo cual hace razonable que estos tweets estén cercanos y terminen siendo parte del mismo cluster. El tercer tweet si bien parece un poco diferente, no lo es tanto dado que existe una palabra en común y es la única palabra presente en los primeros 2 tweets. Esta es “viva”. Recordando que los tweets a este nivel son preprocesados, las menciones no son tenidas en cuenta para esta ejecución del algoritmo de clustering, al igual que otras palabras como stopwords y símbolos de puntuación. En resumen los tweets se reducen a:

- viva
- viva

Figura 5.4: Clusters pertenecientes al corpus clusterizado manualmente. Nueve clusters más el cluster ruido.

- grandes honran gran familia partido nacional viva blancos

Consideremos el segundo cluster representado por [*encabeza, intendente, despedida, senador, multitudinaria, año*] que se conforma parcialmente por los tweets presentados a continuación. La lista completa de tweets se puede consultar en el apéndice E.

1 RT @maurisanner: Despedida de año de la @amo_62florida con @CEnciso62 y @LuisLacallePou https://t.co/g9eXj0RBaK

5 RT @SenJavierGarcia: Gran despedida del año Agrup Manuel Oribe con intendente @CEnciso62 dip@JoseArocena @adelgado404 y @LuisLacallePou ht...

5 RT @SenJavierGarcia: Gran despedida del año Agrup Manuel Oribe con intendente @CEnciso62 dip@JoseArocena @adelgado404 y @LuisLacallePou ht...

6 RT @CachitoMarrero: Aquí está La Manuel Oribe Festejando la Fiesta Despedida del Año @CEnciso62 @LuisLacallePou @PNACIONAL https://t.co/orI...

6 RT @CachitoMarrero: Aquí está La Manuel Oribe Festejando la Fiesta Despedida del Año @CEnciso62 @LuisLacallePou @PNACIONAL https://t.co/orI...

1 RT @maurisanner: Despedida de año de la @amo_62florida con @CEnciso62 y @LuisLacallePou https://t.co/g9eXj0RBaK

11 RT @glaborde78: Gran convocatoria de @JRadiccioni el sábado pasado en San Ramon @CPereyraSM @fernandowvillar @LuisLacallePou https://t.co/v...

13 RT @JRadiccioni: Despedida del año de la Lista 400 Video completo en https://t.co/efkZHGvg49 @PNACIONAL @bace @Herrerismo @LuisLacallePou @...

15 RT @TodosUy: Llega @LuisLacallePou a San Ramón para reencontrarse con amigos y participar de la despedida del año de lista 400 https://t.co...

16 @LuisLacallePou mañana a las 18 hrs en @MunicipioGMvd se coloca cápsula del tiempo que se abrirá en 2066. Te esperamos

19 Habria que evaluar. En que categoria de IMBECIL se situa @LuisLacallePou y si los que lo aplauden no lo superan lar... https://t.co/hsDsOAqW2x

Para este cluster se tiene un número superior de tweets. Existen más matices y por este motivo el cluster representado está formado por más palabras.

Una vez más, la presencia de retweets se hace visible en este cluster. La cantidad de "duplicados" no es elevada, sin embargo se da una situación particular. Se considera como ejemplo los tweets 1, 5 y 6. Si se analizan en detalle estos tres tweets, se puede ver que no son el mismo tweet. El contenido cambia ligeramente, sin embargo todos ellos contienen la palabra "despedida". Esto ocurre con 11 de 19 tweets lo cual permite al algoritmo de clustering armar cierta semejanza en los tweets.

En aquellos tweets que no participa la palabra "despedida", se aprecia que otras palabras de dichos tweets sí aparecen en tweets que contienen la palabra "despedida". De este modo una vez más, la distancia entre estos tweets se ve reducida dado que se comparten atributos en común. Un ejemplo de este caso son los tweets 11, 13 y 15.

Como contraparte se observan casos donde hay tweets que no están relacionados al tópico del cluster y pertenecen a él de todas formas. Tal es el caso de los tweets 16 y 19. En estas situaciones es más difícil inferir la razón de vinculación por parte del algoritmo a un mismo cluster. Aún así, no podemos descuidar el hecho de que los vectores de atributos que representan un tweet son de dimensiones considerables y esto implica valorar muchas magnitudes y no solo las palabras claves. Otro factor influyente es que hay ocasiones en que al ojo humano dos tweets no le parecen similares, pero los vectores que lo representan tienen muchas dimensiones y esto en ciertas ocasiones determina que tweets no "tan similares" terminan siendo "más similares" que el resto de tweets del corpus. De esta forma estos tweets o son identificados como ruido o forman parte de un cluster que tal vez no los identifica demasiado.

Capítulo 6

Conclusiones

En este trabajo se construye un sistema de clustering de tweets que involucra múltiples pasos incluyendo la recolección, segmentación, limpieza, clustering y visualización de tweets. Para ello, se construye una aplicación para recolectar tweets que refieren a múltiples personas, a partir de la cual se crea un corpus de 419 mil tweets. Este corpus se encuentra dividido por personas y por fechas durante un período de siete meses. Se logra construir un sistema de clustering que logra superar la línea base en la gran mayoría de los experimentos realizados para los dos algoritmos de clustering estudiados. También se construyen visualizaciones de generación automática para representar el resultado del clustering gráficamente. Esto permite observar de forma sencilla los temas más relevantes para el corpus.

Con este trabajo se genera un precedente relacionado al clustering en el contexto de proyectos de grado ya que hasta la fecha el tema no había sido tratado en ese marco. Se logra introducir el tema de clustering con un nivel de profundidad extensivo. Además, se genera material que puede ser de utilidad para futuros trabajos dentro del mismo contexto.

A pesar de ser el clustering un tema con varios años de estudio, su uso aplicado a Twitter es algo relativamente reciente y los trabajos disponibles no son abundantes como en otras áreas. Luego de intensas búsquedas se logró recolectar algunos trabajos que en su globalidad buscan alcanzar el mismo objetivo, el clustering de documentos cortos o tweets.

A lo largo de la elaboración de este proyecto, se rescatan muchas lecciones aprendidas, que se detallan en los párrafos siguientes.

En cuanto al clustering mediante hashtags se puede afirmar que solo es útil si los tweets del corpus poseen en su mayoría hashtags y si el nivel de granularidad provisto por los hashtags presentes es suficiente como para discriminar los distintos temas. Para esto se tiene que dar una condición y es que el uso de las etiquetas sea el debido, en caso contrario es similar a realizar clustering de forma aleatoria, y esto es lo que termina sucediendo en nuestra realidad.

En cuanto a los experimentos realizados, se tienen algunas observaciones. Como se ve en el primer experimento, la limpieza de los tweets es fundamental. Un tweet puede

poseer muchas características que solo generan ruido como ser emoticones, abreviaturas, expresiones de risa, símbolos incompletos, entre otros. De esta manera, resulta vital que los tweets sean preprocesados previo a comenzar con la tarea de clustering si se quieren lograr resultados interesantes.

En el segundo experimento se observó como las distintas características en las representaciones de los elementos a clusterizar son extremadamente importantes. La elección para determinar cómo un elemento se representará, así como los atributos que lo identifican cumplen un rol fundamental. También lo es la noción de distancia utilizada, ya que afecta directamente a los resultados.

Debido al poco contexto que posee un tweet es fundamental poder de alguna manera expandir este contexto o lograr comprenderlo con el mayor nivel de detalle posible, ya sea con información externa al tweet o buscando relaciones entre tweets. Agregar estos datos a la representación del tweet es una forma válida y demostró ser muy útil en algunos trabajos previos, pero como se observa en el tercer experimento su aplicación indiscriminada no necesariamente genera mejores resultados. En caso de realizar algún tipo de expansión es importante no hacerlo aleatoreamente y sin tener en cuenta los posibles problemas que se pueden introducir, como algunos de los presentados en la sección 4.3.

Entender cómo afecta al algoritmo y determinar los parámetros a utilizar en los métodos utilizados es de alta importancia ya que puede ayudar a mejorar enormemente los resultados. La realización de esta tarea de forma automática no es fácil y no siempre resulta en valores exitosos.

La evaluación del clustering es muy compleja y tiene muchos falsos positivos, la evaluación interna es muy útil si se tiene certeza absoluta sobre la representación de los datos y la noción de distancia. En caso de existir mayor incertidumbre en estos aspectos la evaluación externa es más adecuada. Una desventaja que posee la evaluación externa es la dificultad para llevarla a cabo ya que por lo general se requiere intervención humana y esto puede llevar cantidades considerables de tiempo y trabajo.

La visualización en forma de nube de los clusters generados es una herramienta muy útil. Permite ver con facilidad los resultados, pero, como vimos, si el proceso de clustering no alcanza un nivel determinado se hace compleja su interpretación por la cantidad de ruido presente. Asimismo, observar los datos de una manera resumida puede hacer la tarea de detectar problemas más sencilla.

6.1. Trabajo a futuro

El proyecto fue sufriendo ajustes que llevaron a resolver ciertos problemas y posponer otros. En cuanto al clustering en términos generales, se utilizaron dos algoritmos de clustering diferentes, sin embargo, existen otros acercamientos y técnicas que podrían utilizarse. Se podría probar la implementación de nuevas formas de representar los tweets que contemplen los puntos débiles que se fueron detallando a lo largo del documento.

Dado que el contexto es escaso por naturaleza en un tweet, cuanto más información precisa se agregue, mejor los resultados a obtener. Otra formas de agregar contexto pue-

den mejorar el resultado del clustering, sin embargo, siempre hay que tener en cuenta el tipo de información a anexar. Como vimos, el hecho de agregar información no es suficiente, debe tener un nivel de especificidad suficiente para no generar mayor interferencia en los tweets.

Una observación respecto a la naturaleza de los tweets y la forma en que se utilizan por la gente sugiere la generación y estudio de conexiones entre tweets. En el marco de este proyecto, los tweets fueron tratados como objetos individuales, sin embargo, se notaron patrones de relacionamiento que resultaban en conversaciones y/o información sobre una misma actividad desde otra perspectiva. Como trabajo a futuro se podrían asociar acontecimientos a los diferentes eventos. Por ejemplo, en el caso de entrevistas, lograr agruparlas con lo que se dice en dicha entrevista. También se sugiere el estudio de conversaciones entre tweets.

Otro posible aspecto a extender es la noción de sentimientos de los clusters. Una vez construído un cluster, conocer la positividad del tópico puede ser un aspecto interesante para conocer el sentimiento y/o reputación de/hacia una persona. Buscar la integración de este proyecto con otros proyectos de grado puede ser un paso inicial para conocer aquellas características a mejorar.

Por último, la realización de una línea de tiempo que muestre los tópicos para una persona en un período de tiempo dado no se muestra en este trabajo. Este trabajo deja todo lo necesario para realizarla, del mismo modo que se presentan los clusters resultantes para Luis Lacalle en la sección de 5.2 del capítulo 5, es posible realizar su análogo para una secuencia de días determinada.

Glosario

AdSocia Primer plataforma de publicidad en Twitter para Latinoamérica que une famosos y anunciantes. Brinda una funcionalidad de ranking que permite listar y filtrar bajo ciertos criterios de búsqueda. `http://adsocia.com/index.php?m=website&a=ranking&place=Uruguay`. 32, 69

algoritmo En el ámbito matemático se aprecian definiciones como: conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas. 2, 3, 6, 69

API Application Programming Interface. 33, 69

Application Programming Interface Interfaz que es utilizada para acceder o utilizar una aplicación o servicio. 33, 69, 71

Bag of Words Forma de representar las instancias, en donde cada una es llevada a una forma vectorial de conteo de las apariciones de sus palabras, tomándose por lo tanto como características a la cantidad de veces que aparece cada palabra. 43, 69, 71

BOW Bag of Words. 43, 69

característica característica (o atributo, o en inglés feature o attribute) es una propiedad medible de una instancia que se observa. 2, 69

corpus Conjunto habitualmente muy amplio y estructurado de textos, que generalmente representan ejemplos reales de uso de una lengua dentro un contexto. 2, 29, 41, 69

El Observador Portal multimedia de noticias actualizadas las 24 horas, sobre Uruguay y el mundo. Publicó en su edición del día 17 de Julio de 2012 a la hora 15:21 un resumen con aquellas cuentas más influyentes en Uruguay. `http://www.elobservador.com.uy/las-cuentas-twitter-uruguayas-mas-influyentes-n228321`. 32, 69

Freeling Es una librería de análisis morfosintáctico, detección de entidades, POS-tagging, parsing, desambiguación y etiquetado semántico. 47, 69

instancia elemento perteneciente al dominio definido de un tópico que es objeto de estudio. 6, 8, 69

Klout score Es un medidor de la influencia social utilizado por la plataforma Klout para puntuar sus usuarios en base a esta posible influencia. Es un valor que oscila entre 1 y 100, donde 100 indica un entidad de máxima influencia.`https://klout.com`. 32, 69

lema Unidad autónoma constituyente del léxico de un idioma. Es una serie de caracteres que forman una unidad semántica. 47, 69

léxico Conjunto de palabras que conforman un determinado lecto. 47, 69, 72

mención Esta es otra forma de hacer de tus tweets algo más parecido a una conversación. Permite mencionar a una cuenta de una persona y así entablar "mini diálogos". Vea el link `https://support.twitter.com/articles/14023` recuperado el 17 de Noviembre de 2016 por más información. 31, 69

Pantallazo Portal web sobre noticias políticas, económicas, deportivas, culturales, tecnológicas y de interés general de Uruguay y el mundo. También publica aquellos uruguayos con mayor cantidad de seguidores en Twitter el día 21 de Abril de 2016 a la hora 18:48. `http://www.pantallazo.com.uy/contenido/Twitter-cumple-10-anos--Cuales-son-los-uruguayos-con-mas-seguidores--302964`. 32, 69

PLN Procesamiento del Lenguaje Natural. 69

Procesamiento del Lenguaje Natural Campo que combina tecnologías de la ciencia computacional con la lingüística aplicada, con el objetivo de hacer posible la comprensión y el procesamiento asistidos por computadora de información expresada en lenguaje humano para determinadas tareas. 3, 69, 72

rate limiting La API de Twitter limita la cantidad de pedidos que se pueden realizar a cada uno de sus endpoints. Estos límites son en base a las invocaciones que se realizan a través de un usuario autenticado. Se puede ver más detalle en el sitio oficial `https://dev.twitter.com/rest/public/rate-limiting` así como también las ventanas de tiempo y peticiones de los diferentes endpoints en la tabla presente en `https://dev.twitter.com/rest/public/rate-limits` . Ambos links fueron consultados por última vez el día 28 de Abril de 2017. 33, 69

retweet Un Retweet es publicar nuevamente un Tweet. La característica retwittear de Twitter ayuda a los usuarios a compartir ese tweet con todos sus seguidores. Es posible retwittear tus propios tweets y los tweets de los demás. A veces la gente escribe RT al inicio de un tweet para indicar que están publicando nuevamente el contenido de alguien más. Eso no es una característica o comando oficial de Twitter, pero quiere decir que están citando el tweet de otro usuario." (Twitter, 2017). Recuperado de `https://support.twitter.com/articles/230754` el día 15 Noviembre de 2016. 34, 58, 69, 73

RT Retweet. 69

Statista Portal de estadísticas de más de 18.000 recursos diferentes. Establece en su reporte extraído de `https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/` el día 5 mayo de 2017 la cantidad de usuarios activos estimados al mes desde el comienzo del año 2010 hasta Mayo de 2017. 69

Term frequency – Inverse document frequency Es una medida que expresa cuán relevante es una palabra para un documento en un corpus. 50, 69, 73

TFIDF Term frequency – Inverse document frequency. 69

token Símbolo utilizado como unidad mínima de trabajo en el análisis de cierto texto. En este trabajo un token equivale a una palabra en un tweet. 44, 69

tokenización Proceso de separar un texto en tokens. 44, 69

twitter Red social en donde solamente se pueden publicar y compartir tweets (mensajes cortos de 140 caracteres). 69

Word2Vec Esta herramienta proporciona una implementación eficiente de los modelos de bow-of-words y skip-gram para computar representaciones vectoriales de palabras. Estas representaciones se pueden utilizar posteriormente en muchas aplicaciones de procesamiento de lenguaje natural y para investigación adicional. Por detalles de la implementación existe su fuente original en `https://code.google.com/archive/p/word2vec/` recuperada el 3 de Enero de 2017. 24, 69

WordNet Base de datos léxica que agrupa palabras en conjuntos de sinónimos llamados Synsets, proporciona definiciones cortas y generales y almacena las relaciones semánticas entre los conjuntos de sinónimos. 47, 69

Apéndice A

Filtros

La presente sección tiene por objetivo detallar los filtros mencionados en la fase de preprocesamiento de los tweets. Se presentan en el orden que se detallaron originalmente en el documento y se listan al mismo tiempo varias referencias que complementan en detalle cada uno de los filtros.

Eliminación de abreviaciones

Este filtro es una lista customizada de abreviaciones que aplican en el contexto del idioma español y que se utiliza con frecuencia en muchos ámbitos de comunicación. La lista es la siguiente:

[‘rt’, ‘xq’, ‘hrs’, ‘d’, ‘c’, ‘q’, ‘x’, ‘desp’]

Eliminación de emojis

Los emojis como se mencionó se conforman por dos tipos de filtros distintos:

i. Emojis sencillos

   [‘:)’, ‘:D’, ‘;)’, ‘;D’, ‘:/’, ‘:s’, ‘:P’, ‘:-P’, ‘xD’]

ii. Emojis UNICODE

   La lista de códigos UNICODE para los emoticones utilizada es la correspondiente a la quinta versión de emoticones y se lista en su completitud en el sitio: `http://unicode.org/emoji/charts/full-emoji-list.html` Extraída el día 20 Abril 2017.

Además, esta lista debió ser complementada con símbolos de otros idiomas. La base de estos caracteres fue recolectada de `http://jrgraphix.net/research/unicode_blocks.php` el día 27 abril de 2017.

Eliminación de risa

Para eliminar la risa se utiliza la siguiente expresión regular base:

$$\backslash b(j|a) + \backslash b$$

La misma concatenada con cada una de las variantes para las diferentes vocales conforma la expresión regular:

$$\backslash b((j|a)+)\backslash b|\backslash b((j|e)+)\backslash b|\backslash b((j|i)+)\backslash b|\backslash b((j|o)+)\backslash b|\backslash b((j|u)+)\backslash b$$

Esta última encuentra patrones de la forma ja, jj, jaja, jajja, je, ej, jjeje, jej, jeje... y así para cada una de las vocales.

Eliminación de números

El filtro de numeración se realiza con una expresión regular capaz de reconocer cualquier número entero entre $[-min\_int, max\_int]$.

$$(\hat{}-?[0-9] + \$)$$

Eliminación de puntuación

Para eliminar puntuación se utiliza como filtro una lista de símbolos de puntuación utilizado en el idioma español.

[‘¡’, ‘!’, ‘"’, ‘#’, ‘$’, ‘%’, ‘&’, ‘(’, ‘)’, ‘*’, ‘+’, ‘,’, ‘-’, ‘.’, ‘/’, ‘:’, ‘;’, ‘<’, ‘=’, ‘>’, ‘¿’, ‘?’, ‘@’, ‘[’, ‘]’, ‘^’, ‘_’, ‘`’, ‘{’, ‘|’, ‘}’, ‘~’, ‘¸’, ‘..’, ‘...’, ‘. . . ’]

Eliminación de URLs

En la eliminación de URLs, se utilizan dos expresiones regulares de distintos tipos. Una sobre tokens y otra sobre texto plano. Esto se debe básicamente a la diferencia que se tiene entre la línea base y el desarrollo. Para la línea base se utiliza el texto de los tweets como una cadena de caracteres y por ende se utiliza la siguiente expresión regular:

https[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*(),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+)

Cuando refiere al desarrollo de la solución al problema se utiliza la segunda forma de remover URLs:

^http[s]?

Esto es posible dado que el proceso de remover links en este caso se realiza sobre el contexto de un token, y por ende si un token comienza de tal forma, se está en presencia de un link.

Eliminación de RTs

Para filtrar retweets se tienen en cuenta algunos aspectos importantes. Por un lado, la forma de reconocer un retweet es a través de consultar el inicio de un tweet de la forma 'rt '. En Twitter es obligatorio que un retweet comience de la forma RT dado que es un token que agrega la plataforma a aquellos tweets que además son retweets de algún otro. La comparación se ejecuta con el comparativo en minúsculas dado que el tweet es previamente llevado a esta forma.

Por otro lado, un aspecto fundamental al momento de remover re-tweets yace sobre el hecho de remover solo aquellos tweets en el cual el tweet re-tweeteado fue recuperado y es parte del corpus. Esto en otros términos implica el hecho de no perder tweets por el simple hecho de ser un re-tweet. En aquellos casos donde Twitter no proveyó del tweet original al menos se perdura una copia re-tweeteada.

Apéndice B

Arquitectura del sistema

A continuación se muestra un diagrama de la arquitectura del sistema (B.1). La misma se encuentra modularizada por componentes.

Etapas

1. Etapa de recuperación de tweets y construcción de corpus. Por detalles, consultar el capítulo 3.
2. Separación de tweets por fechas.
3. Preprocesamiento, filtrado y expansión de tweets. El capítulo 4, secciones 2 y 3 detallan esta etapa.
4. Existe una cuarta etapa que es la de asignación y obtención de atributos. La misma se define en la sección 4 del capítulo 4.
5. El quinto componente es el encargado de llevar a cabo el clustering en sí. Para conocer al detalle esta etapa se recomienda consultar la sección 5 del capítulo 4, así como también el capítulo 5 que analiza los resultados para los diferentes experimentos realizados. En esta etapa existe también un postprocesamiento de los clusters resultantes, que se puede consultar en la sección 6 del capítulo 4 por más detalles.
6. El último componente es el encargado de llevar los resultados del proceso de clustering a su forma gráfica en nubes de palabras, por mas detalles puede consultarse la sección 2 del capítulo 5.

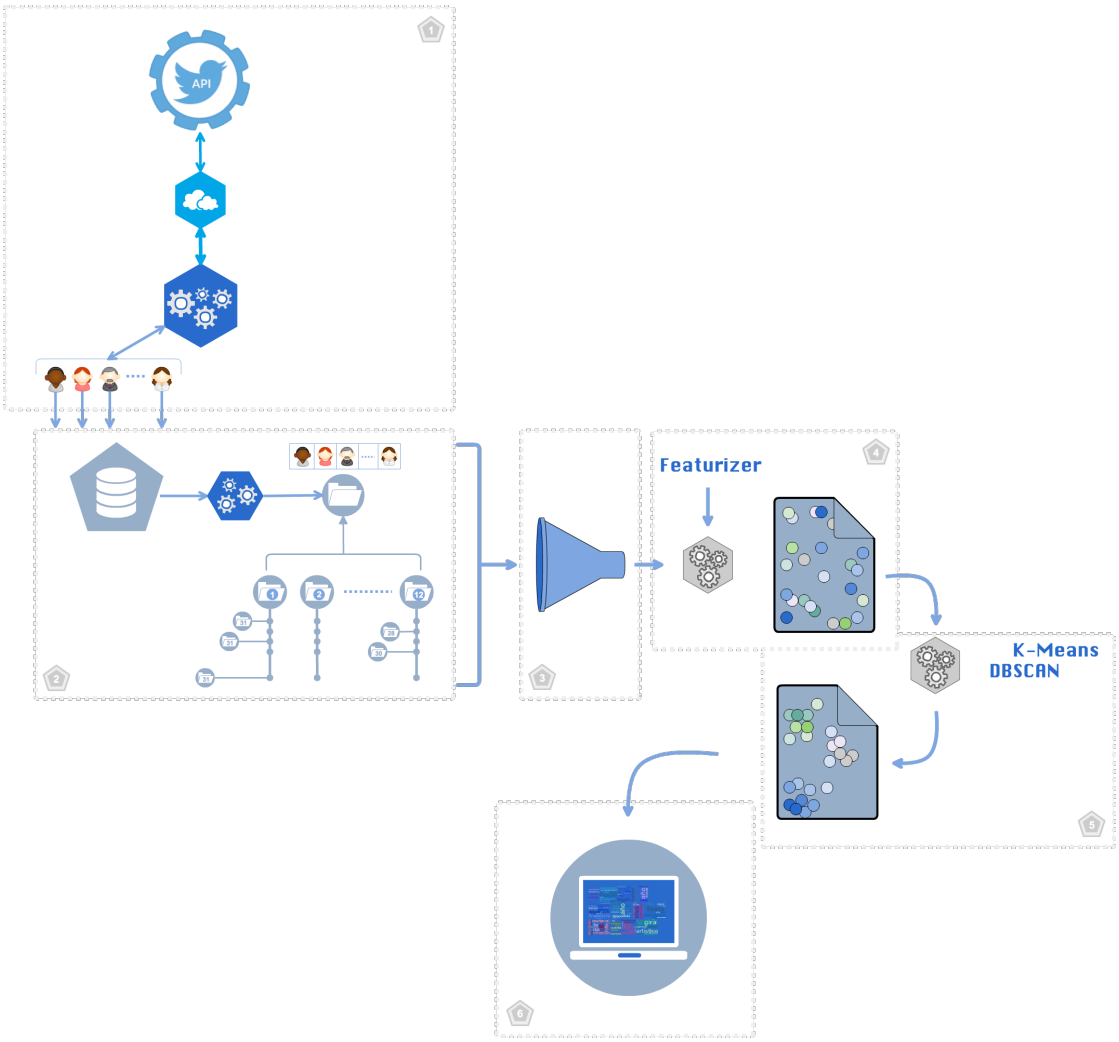


Figura B.1: Arquitectura del sistema

Apéndice C

Uso del programa de clustering

En el presente apéndice se muestra como utilizar el programa para hacer clustering. Al ejecutar el programa con la bandera "help", se muestran diferentes opciones configurables que pueden ser tenidas en cuenta al momento de ejecución.

```
Usage:
        python3 -m lib.python.clustering-algorithms.bow_custom [options]

Options:

-h, --help
        Show this help message and exit.

--al=ALGORITHM, --algorithm=ALGORITHM
        Set the algorithm to use for clustering. Options are: kmeans
        (default), dbscan.

-c CORPUS_FILE, --corpus-file=CORPUS_FILE
        File path to the set of tweets that will be clusterized, this
        option is mandatory.

 --extend-features
        Extend feature vector with @, # and words.

--freeling
        Activate freeling to replace those unknown words by
        synset(wordnet).

 --hso, --hide-std-output
        Enable/Disable showing clusters on standard output.
```

```
--ido, --id-only
        If set the cluster output will show only the id of the tweet,
        otherwise the full JSON will be saved.

--it, --id-and-text
        If set the cluster output will show the id and the text of the
        tweet, otherwise the full JSON will be saved.

-k, --find-k
        If present kmeans will calculate best kmeans based on silhouette
        score.

-m METRIC, --metric=METRIC
        Set the metric for validation. Options are: cosine (default),
        euclidean.

--max-df=MAX_DF
        When building the vocabulary ignore terms that have a document
        frequency strictly higher than the given threshold
        (corpus-specific stop words). If float, the parameter represents
        a proportion of documents, integer absolute counts. This
        parameter is ignored if vocabulary is not None.

--min-df=MIN_DF
        When building the vocabulary ignore terms that have a document
        frequency strictly lower than the given threshold. This value is
        also called cut-off in the literature. If float, the parameter
        represents a proportion of documents, integer absolute counts.
        This parameter is ignored if vocabulary is not None.

--minibatch
        Use ordinary k-means algorithm (in batch mode).

--n-features=N_FEATURES
        Maximum number of features (dimensions) to extract from text.

--no-idf
        Disable Inverse Document Frequency feature weighting.

-o OUTPUT_FILE, --output-file=OUTPUT_FILE
        File path where the clustering results will be saved. If this is
        not set the results will be printed to the standard output.
```

```
-p, --plots
        If present it will plot at different stages of the process.

--remove-rt
        Removes re-tweets if enabled.

--sort-output
        Sort output by clusters.

--stemming
        Activate stemming to replace by the stem of the words.

--to, --text-only
        If set the cluster output will show only the text of the tweet,
        otherwise the full JSON will be saved.

--verbose
        Print progress reports inside k-means algorithm.

--with-e-v=EXTERNAL_VALIDATION
        Use the input file as target to external validate the clusters
        generated.

--word2vec=WORD2VEC
        Expands the tokens from a tweet with the most similar word.
```

A continuación se presenta una posible ejecución para el corpus de Luis Lacalle Pou del 9 de Diciembre de 2016 con validación externa y algoritmo K-Means.

```
python3 -m lib.python.clustering-algorithms.bow_custom  --corpus-file
data/person/luis_lacalle_pou/2016/december/9.txt --text-only --with-e-v
data/outputs/llp_2016_dec_9_cluster_manual_for_validation.txt
--algorithm=kmeans
```

Apéndice D

Stopwords

El presente apéndice lista las stopwords utilizadas durante el clustering. La lista está conformada por stopwords para el idioma español provistas por NLTK y otras agregadas que son específicas al corpus. Las palabras agregadas fueron obtenidas mediante un análisis de frecuencias de las palabras en nuestro corpus.

Lista de stopwords

aca, acá, ah, ahi, ahí, al, algo, algunas, algunos, ante, antes, asi, así, aun, aún, bien, cada, casi, como, con, contra, cual, cuando, cómo, da, dan, de, del, desde, di, dia, donde, dos, durante, dí, día, el, él, ella, ellas, ellos, en, entre, era, erais, éramos, eran, eras, eres, es, esa, esas, ese, eso, esos, esta, estaba, estabais, estaban, estabas, estad, estada, estadas, estado, estados, estamos, estando, estar, estaremos, estará, estarán, estarás, estaré, estaréis, estaría, estaríais, estaríamos, estarían, estarías, estas, este, estemos, esto, estos, estoy, estuve, estuviera, estuvierais, estuvieran, estuvieras, estuvieron, estuviese, estuvieseis, estuviesen, estuvieses, estuvimos, estuviste, estuvisteis, estuviéramos, estuviésemos, estuvo, está, estábamos, estáis, están, estás, esté, estéis, estén, estés, fue, fuera, fuerais, fueran, fueras, fueron, fuese, fueseis, fuesen, fueses, fui, fuimos, fuiste, fuisteis, fuéramos, fuésemos, ha, habida, habidas, habido, habidos, habiendo, habremos, habrá, habrán, habrás, habré, habréis, habría, habríais, habríamos, habrían, habrías, habéis, había, habíais, habíamos, habían, habías, hace, han, has, hasta, hay, haya, hayamos, hayan, hayas, hayáis, he, hemos, hube, hubiera, hubierais, hubieran, hubieras, hubieron, hubiese, hubieseis, hubiesen, hubieses, hubimos, hubiste, hubisteis, hubiéramos, hubiésemos, hubo, iba, ir, la, las, le, les, llega, lo, los, mas, me, mejor, mi, mil, min, mio, mis, mismo, mucho, muchos, muy, más, mí, mía, mías, mío, míos, nada, ni, nos, nosotras, nosotros, nuestra, nuestras, nuestro, nuestros, os, otra, otras, otro, otros, pa, para, pero, poco, por, porque, que, quien, quienes, qué, retweet, retweeted, se, sea, seamos, sean, seas, sentid, sentida, sentidas, sentido, sentidos, ser, seremos, será, serán, serás, seré, seréis, sería, seríais, seríamos, serían, serías, seáis, siente, sigue, sin, sintiendo, sobre, sois, somos, son, sos, soy, su, sus, suya, suyas, suyo, suyos, sé, sí, tal, también, tan, tanto, te, tendremos, tendrá, tendrán, tendrás, tendré, tendréis, tendría, tendríais, tendríamos, tendrían, tendrías, tened, tenemos, tenga, tengamos, tengan, tengas, tengo, tengáis, tenida, tenidas, tenido,

tenidos, teniendo, tenéis, tenía, teníais, teníamos, tenían, tenías, ti, tiene, tienen, tienes, todo, todos, tu, tus, tuve, tuviera, tuvierais, tuvieran, tuvieras, tuvieron, tuviese, tuvieseis, tuviesen, tuvieses, tuvimos, tuviste, tuvisteis, tuviéramos, tuviésemos, tuvo, tuya, tuyas, tuyo, tuyos, tí, tú, un, una, uno, unos, va, ve, ver, vi, vos, vosostras, vosostros, vs, vuestra, vuestras, vuestro, vuestros, ví, ya, yo.

Apéndice E

Cluster de ejemplo

Tweets pertenecientes al cluster representado por [*encabeza, intendente, despedida, senador, multitudinaria, año*]

1 RT @maurisanner: Despedida de año de la @amo_62florida con @CEnciso62 y @LuisLacallePou https://t.co/g9eXj0RBaK

2 @LuisLacallePou Bromas mediante, se dirige a esta Agrupación, que junto a @CEnciso62 supo proclamarlo candidato en... https://t.co/tdj9VkKbP0

3 @LuisLacallePou en despedida de @amo_62florida: "Saben como le dicen al Pájaro: Corralón Municipal, puro carretilla... https://t.co/zRhZ3Whx4H

4 @LuisLacallePou: "El acto es dos veces más grande que el del año pasado y el pajaro mas chico" @tvflorida https://t.co/N5T6jDmbj5

5 RT @SenJavierGarcia: Gran despedida del año Agrup Manuel Oribe con intendente @CEnciso62 dip@JoseArocena @adelgado404 y @LuisLacallePou ht...

5 RT @SenJavierGarcia: Gran despedida del año Agrup Manuel Oribe con intendente @CEnciso62 dip@JoseArocena @adelgado404 y @LuisLacallePou ht...

6 RT @CachitoMarrero: Aquí está La Manuel Oribe Festejando la Fiesta Despedida del Año @CEnciso62 @LuisLacallePou @PNACIONAL https://t.co/orI...

7 @GramajoWilson @CEnciso62 @MarcosPerez62 @Nancymartinezc @LuisLacallePou @RafaelPereyra17 @evelin_olga Que bueno ver tanta gente buena junta

6 RT @CachitoMarrero: Aquí está La Manuel Oribe Festejando la Fiesta Despedida del Año @CEnciso62 @LuisLacallePou @PNACIONAL https://t.co/orI...

1 RT @maurisanner: Despedida de año de la @amo_62florida con @CEnciso62 y @LuisLacallePou https://t.co/g9eXj0RBaK

8 RT @TodosUy: .@LuisLacallePou obre el proyecto de los Centros socioeducativos de FOEB "me reconforta un sindicato que quiera lo mejor para...

9 RT @fperdomo400: Como todos los años la despedida del año de la Lista 400 @LuisLacallePou https://t.co/Nfj3AwUouZ

10 RT @CPereyraSM: Lista 400 en Canelones con @LuisLacallePou @JRadiccioni @Herrerismo @TodosUy @bacedaprensa @joselo400 https://t.co/2ejc4waC...

11 RT @glaborde78: Gran convocatoria de @JRadiccioni el sábado pasado en San Ramon @CPereyraSM @fernandowvillar @LuisLacallePou https://t.co/v...

12 RT @JRadiccioni: Con Los Kompadres y @LuisLacallePou festejando en San Ramón https://t.co/rmbWhVtDvN

13 RT @JRadiccioni: Despedida del año de la Lista 400 Video completo en https://t.co/efkZHGvg49 @PNACIONAL @bace @Herrerismo @LuisLacallePou @...

14 RT @JRadiccioni: Desde San Ramón la Lista 400 @LuisLacallePou @ANiffouri400 @PNACIONAL @TodosUy @Herrerismo @bacedaprensa @Blancos_UY @Cla...

15 RT @TodosUy: Llega @LuisLacallePou a San Ramón para reencontrarse con amigos y participar de la despedida del año de lista 400 https://t.co...

16 @LuisLacallePou mañana a las 18 hrs en @MunicipioGMvd se coloca cápsula del tiempo que se abrirá en 2066. Te esperamos

17 RT @TodosUy: .@LuisLacallePou en la despedida del año de la lista 400 de Canelones, agradece a los compañeros el trabajo realizado en este...

18 FIESTA COMPLEJO AMÉRICA CON SU PRESIDENTE JUNTO AL PAN @gloriasaravista @LuisLacallePou @PNACIONAL https://t.co/SQ340Ao4a1

19 Habria que evaluar. En que categoria de IMBECIL se situa @LuisLacallePou y si los que lo aplauden no lo superan lar... https://t.co/hsDsOAqW2x

Bibliografía

[Antenucci, 2011] Antenucci, D., H. G. M. A. T. M. (2011). *Classification of Tweets via clustering of hashtags.* Recuperado de http://www.twiki.di.uniroma1.it/pub/ApprAuto/WebHome/AntenucciHandyModiTinkerhess.pdf.

[Banfield, 1993] Banfield, J. D., R. A. E. (1993). *Model-based Gaussian and non-Gaussian clustering.* Internaltional Biometrics Society. Recuperado de https://www.stat.washington.edu/raftery/Research/PDF/banfield1993.pdf.

[Bezdek, 1998] Bezdek, J., P. N. (1998). *Some New Indexes of Cluster Validity.* IEEE Transactions on Systems, Man, and Cybernetics.

[Brun, 2005] Brun, M., S. C. H. J. L. J. C. B. S. E. D. E. (2005). *Model-based evaluation of clustering validation measures, Pattern Recognition.* Pattern Recognition. Recuperado de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.2871&rep=rep1&type=pdf.

[Cardellino, 2016] Cardellino, C. (2016). *Spanish Billion Words Corpus and Embeddings.* Recuperado de http://crscardellino.me/SBWCE/.

[Cimiano, 2004] Cimiano, P., H. A. S. S. (2004). *Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text.* Proceedings of the 16th European Conference on Artificial Intelligence.

[Cubero, 2015] Cubero, M., C. S. (2015). *Detección de humor en textos en español.* Facultad de Ingeniería: Uruguay. https://www.fing.edu.uy/inco/grupos/pln/prygrado/Informepghumor.pdf.

[Davies, 2000] Davies, D., B. D. (2000). *A cluster separation measure.* IEEE Trans. Pattern Anal. Machine Intell.

[Defays, 1977] Defays, D. (1977). *An Efficient Algorithm for a Complete Link Method.* Computer Journal.

[Dela Rosa, 2011] Dela Rosa, K., S. R. L. B. F. R. (2011). *Topical Clustering of Tweets.* Carnegie Mellon University. http://www.cs.cmu.edu/~kdelaros/sigir-swsm-2011.pdf.

[Desgraupes, 2013] Desgraupes, B. (2013). *Clustering Indices*. University Paris Ouest.

[Duda, 2001] Duda, R. O., H. P. E. S. D. G. (2001). *Pattern Classification*. Wiley.

[Dunn, 1974] Dunn, J. (1974). *Well separated clusters and optimal fuzzy partitions*. Journal of Cybernetics.

[Ester, 1996] Ester, M., K. H. P. S. S. X. X. (1996). *A Density-Based Algorithm for Discovering Clusters*. AAAI Press. Recuperado de `https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf`.

[Everitt, 2011] Everitt, B., L. S. L. M. S. D. (2011). *Cluster Analysis*. Wiley. Recuperado de `http://hbanaszak.mjr.uw.edu.pl/TempTxt/EverittEtAl_2011_ClusterAnalysis.pdf`.

[Gan, 2011] Gan, G. (2011). *Data Clustering in C++: An object oriented approach*. Chapman & Hall.

[Gath, 1989] Gath, I., G. A. (1989). *Unsupervised optimal fuzzy clustering*. Pattern Analysis and Machine Intelligence, IEEE Transactions.

[Godfrey, 2014] Godfrey, D. (2014). *A Case Study in Text Mining: Interpreting Twitter Data From World Cup Tweets*. Carnegie Mellon University. `https://arxiv.org/pdf/1408.5427.pdf`.

[González, 2010] González, D. (2010). *Algoritmos de Agrupamiento basados en densidad y Validación de clusters. Tesis Doctoral*. Recuperado de `http://www.cerpamid.co.cu/sitio/files/DamarisTesis.pdf`.

[Grabusts P., 2002] Grabusts P., B. A. (2002). *Using Grid-clustering methods in data classification*. IEEE.

[Halkidi, 2001a] Halkidi, M., B. Y. V. M. (2001a). *On Clustering Validation Techniques*. Journal of Intelligent Information Systems. Recuperado de `http://web.itu.edu.tr/sgunduz/courses/verimaden/paper/validity_survey.pdf`.

[Halkidi, 2001b] Halkidi, M., S. D. T. G. V. M. (2001b). *Data mining in Software Engineering*. Intelligent Data Analysis.

[hees, 1986] hees, E. (1986). *The effectiveness and efficiency of agglomerative hierarchic clustering in document retrieval*. Cornell University Ithaca.

[Hubert, 1985] Hubert, L., A. P. (1985). *Comparing partitions*. P. Journal of Classification.

[Introini, 2011] Introini, D., L. D. (2011). *Proyecto detección clusters*. Recuperado de `https://eva.fing.edu.uy/file.php/514/ARCHIVO/2011/TrabajosFinales2011/informe_final_introini_lena.pdf`.

[Jain, 1988] Jain, A., D. R. (1988). *Algorithms for Clustering Data.* Prentice Hall. Recuperado de `http://www.cse.msu.edu/~jain/Clustering_Jain_Dubes.pdf`.

[Jain, 1999] Jain, A., M. M. F. P. (1999). *Data Clustering: A Review.* ACM Computing Surveys.

[Kalyani, 2012] Kalyani, P. (2012). *Approaches to Partition Medical Data using Clustering.* International Journal of Computer Applications. Recuperado de `https://pdfs.semanticscholar.org/82e1/22314dcbe9170e3f8fe9863737cc6f2237ec.pdf`.

[Lieberman, 2013] Lieberman, M. (2013). *Why our brains are wired to connect.* Oxford University Press.

[Manning, 2009] Manning, C., R. P. S. H. (2009). *An Introduction to Information Retrieval.* Cambridge University Press. `https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf`.

[Manning C., 1993] Manning C., Raghavan P., S. H. (1993). *Model-based Gaussian and non-Gaussian clustering.* International Biometric Society. Recuperado de `https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf`.

[Mikolov, 2013] Mikolov, T., S. I. C. K. C. G. D. J. (2013). *Efficient Estimation of Word Representations in Vector Space.* Recuperado de `https://arxiv.org/pdf/1301.3781.pdf`.

[Nesmachnow, 2010] Nesmachnow, S. (2010). *Computación científica de alto desempeño en la Facultad de Ingeniería, Universidad de la República.* Revista de la Asociación de Ingenieros del Uruguay.

[Ozdikis, 2012] Ozdikis, O., S. P. O. H. (2012). *Semantic Expansion of Hashtags for Enhanced Event Detection in Twitter.* Technical University Turkey. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.455.5135&rep=rep1&type=pdf`.

[Pinker, 1997] Pinker, S. (1997). *How the mind works.* W. W. Norton & Company. Recuperado de `http://hampshirehigh.com/exchange2012/docs/Steven%20Pinker%20-%20How%20The%20Mind-Works.pdf`.

[Rokach, 2005] Rokach, L., M. O. (2005). *Clustering Methods.* Springer US. Recuperado de `https://www.cs.swarthmore.edu/~meeden/cs63/s16/reading/Clustering.pdf`.

[Rousseeuw, 1987] Rousseeuw, P. (1987). *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis.* Journal of Computational and Applied Mathematics.

[Rubio, 2015] Rubio, E. (5 de Diciembre de 2015). 20 faltas de ortografía que vemos frecuentemente en redes sociales. Recuperado de http://cuadernos.rubio.net/prensa/post/20-faltas-de-ortografia-que-vemos-frecuentemente-en-redes-sociales.

[Sahoo, 2006] Sahoo, N., C. J. K. R. (2006). *Incremental Hierarchical Clustering of Text Documents*. Recuperado de `http://www.cs.cmu.edu/~callan/Papers/cikm06-nsahoo.pdf`.

[Sibson, 1972] Sibson, R. (1972). *SLINK: An optimally efficient algorithm for the single-link custer method*. The Computer Journal. Recuperado de `http://www.cs.ucsb.edu/~veronika/MAE/SLINK_sibson.pdf`.

[Smalheiser, 1994] Smalheiser, N. R., S. D. R. (1994). *Assessing a gap in the biomedical literature: Magnesium deficiency and neurologic disease*. Neuroscience Research Communications.

[Smalheiser, 1997] Smalheiser, N. R., S. D. R. (1997). *An interactive system for finding complementary literatures: a stimulus to scientific discovery*.

[Statista, 2017] Statista (2017). Number of monthly active twitter users worldwide from 1st quarter 2010 to 1st quarter 2017 (in millions). Recuperado de `https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/` el 5 de mayo de 2017.

[Swanson, 1987] Swanson, D. R. (1987). *Two medical literatures that are logically but not bibliographically connected*. John Wiley & Sons, Inc.

[Tan, 2006] Tan, P., S. M. K. V. (2006). *Cluster Analsys: Basic concepts and algorithms. Introduction to Data Mining*. Recuperado de `http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf`.

[Tang, 2014] Tang, G., X. Y. W. W. L. R. Z. T. F. (2014). *Clustering tweets using Wikipedia concepts*. Recuperado de `http://www.lrec-conf.org/proceedings/lrec2014/pdf/83_Paper.pdf`.

[Theodoridis, 2003] Theodoridis, S., K. K. (2003). *Pattern Recognition*. Elsevier. Recuperado de `http://www.manalhelal.com/Books/F2014/Pattern%20Recognition_2003.pdf`.

[Tudor, 2013] Tudor, E., B. A. S. E. C. (2013). *Clustering Techniques in Financial Data Analysis Applications On The U.S. Financial Market*. Recuperado de `http://www.utgjiu.ro/revista/ec/pdf/2013-04/29_Serban,Bogeanu,Tudor.pdf`.

[Twitter, 2017] Twitter (2017). ¿qué son las etiquetas (símbolos)? Recuperado de `https://support.twitter.com/articles/247830`.

[Vesanto, 2000] Vesanto, J., A. E. (2000). *Clustering of the self-organizing map*. IEEE Transactions on Neural Networks.

[Voorhees, 1986] Voorhees, E. (1986). *Implementing agglomerative hierarchic clustering algorithms for use in document retrieval*. Information Processing & Management.

[Wagner, 2007] Wagner, S., W. D. (2007). *Comparing Clusterings - An Overview.* Recuperado de `http://i11www.iti.kit.edu/extra/publications/ww-cco-06.pdf`.

[Wikipedia, 2017] Wikipedia (2017). Conceptual clustering. Recuperado de `https://en.wikipedia.org/wiki/Conceptual_clustering`.

[Wikipedia., 2017] Wikipedia. (2017). *Rand Index.* Recuperado de `https://en.wikipedia.org/wiki/Rand_index`.

[Xie, 1991] Xie, X. L., B. G. (1991). *A validity measure for fuzzy clustering.* IEEE Transactions on Pattern Analysis and Machine Intelligence.

[Yeung, 2001] Yeung, K., R. W. (2001). *Details of the Adjusted Rand index and Clustering algorithms Supplement to the paper 'An empirical study on Principal Component Analysis for clustering gene expression data' (to appear in Bioinformatics).* Recuperado de `http://faculty.washington.edu/kayee/pca/supp.pdf`.

[Äyrämö, 2006] Äyrämö, S., K. T. (2006). *Introduction to partitioning-based clustering methods with a robust example.* Reports of the Department of Mathematical Information Technology. Recuperado de `http://users.jyu.fi/~samiayr/pdf/introtoclustering_report.pdf`.