Minería de Medios Sociales - 2024-2025

Universidad de Granada

Práctica de Minería de Texto - KNIME

Miguel García López

Índice

Índice de figuras

Índice de cuadros

1. Introducción

Este documento trata sobre la **minería de texto** y alguna de las técnicas empleadas en todo el flujo completo del proceso. Este flujo de trabajo o *workflow* será realizado con el *software KNIME*, el cual viene integrado con múltiples herramientas específicas para ello, además de contar con *plugins* extra si fuese necesario. *KNIME* tiene además la ventaja de ser una herramienta de construcción de *ETL's* visual, de forma que es muy intuitivo ver el flujo de interacción con los datos.

Para este trabajo se requiere crear dicho flujo utilizando como conjunto de datos una serie de documentos de texto. Se deben realizar al menos:

1. Tres nodos de pre-procesado de los datos.
2. Una técnica de minería con su visualización.
3. Una técnica de visualización de los documentos.

2. Conjunto de datos

Los datos han sido seleccionados del baúl de *Obsidian* del estudiante. Este *software* permite la creación de notas "minimalistas" escritas en *markdown* y la visualización de las mismas. *Obsidian* tiene la característica principal de permitir enlazar notas con otras, creando un grafo dirigido en el proceso.

El estudiante ha estado utilizado el *software* durante años, comenzando su uso en mitad de la carrera de ingeniería informática, por lo que tiene una cantidad de notas suficiente, en total unas 350 notas. En la figura 1 se puede visualizar algunas de las notas del estudiante y en la figura 2 se observa la estructura de grafo creada, la cual enlaza notas entre ellas según una mezcla de toma de notas estilo **MOC** (*Map Of Concepto*) y *Zettelkasten*, que es un estilo de organización que se basa en crear notas individuales (*zettels*) que contienen ideas concretas y autocontenidas.

3. Workflow en KNIME

En la figura 3 se observa el flujo completo, con todos los nodos y sus conexiones. Inicialmente los datos se leen como texto plano de la carpeta de Conceptos del baúl del estudiante. Esta tabla puede observarse en la figura 4.

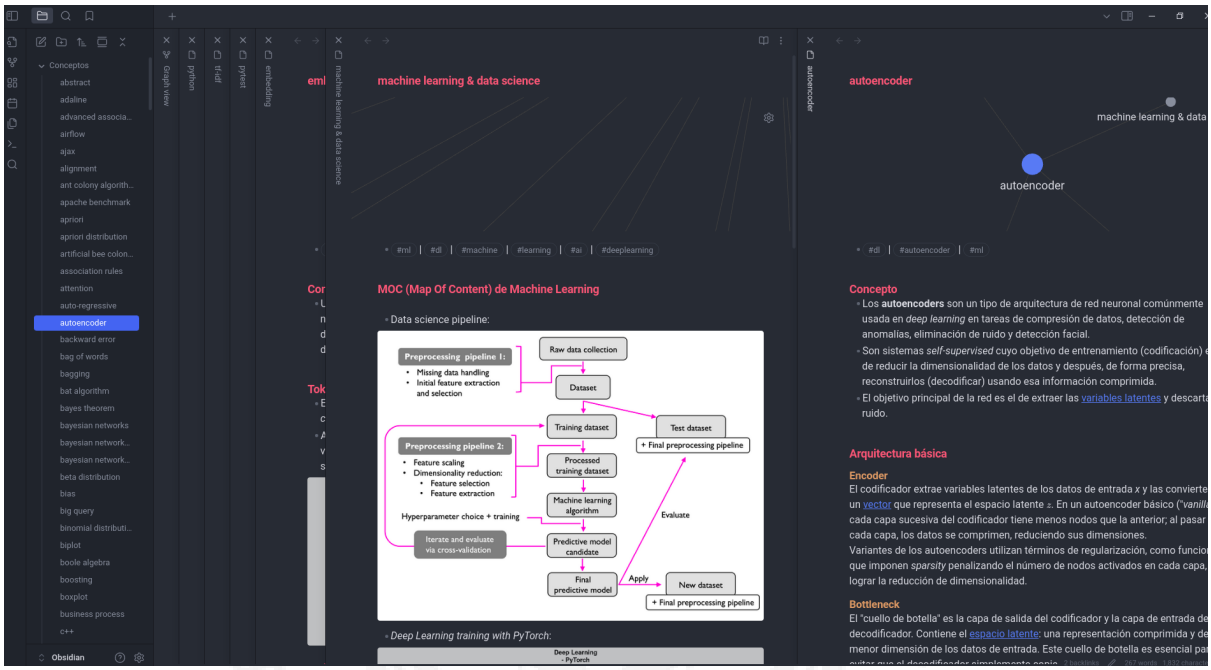

Figura 1: Notas tomadas en el editor *Obsidian*

3.1. Procesamiento del texto

Para el procesamiento previo del texto se usan una serie de nodos en serie que se van a proceder a explicar con detalle. Primero de todo, es necesario convertir la columna principal, *Document Body Text*, de tipo de dato *string* a *document*. Esto se debe hacer ya que ciertos nodos no aceptan como entrada el tipo de datos *string*.

Seguido a ello se utiliza un nodo *Case Converter* para pasar todo el texto a minúscula. Después, al estar tratando con texto en *markdown*, es necesario filtrar ciertos caracteres especiales que "ensucian" la información que se pretende extraer.

Markdown utiliza signos de almohadilla (#) para los encabezados, como títulos o secciones y sub-secciones. Además también utiliza símbolos especiales para las palabras escritas en *itálica* o **negrita**. Todo ello se pre-procesa con las siguientes expresiones de *Regex* utilizando algunos nodos de *String Manipulation*.

- **Caracteres especiales (exclamaciones, comillas, paréntesis angulares, asteriscos, etc)**:

```
regexReplace($Text$, "[>()\#.!?¡\"'*|,‘-]", "")
```

- **Enlaces entre notas**:

```
regexReplace($Text$, "\[\[.*?\]\]", "")
```

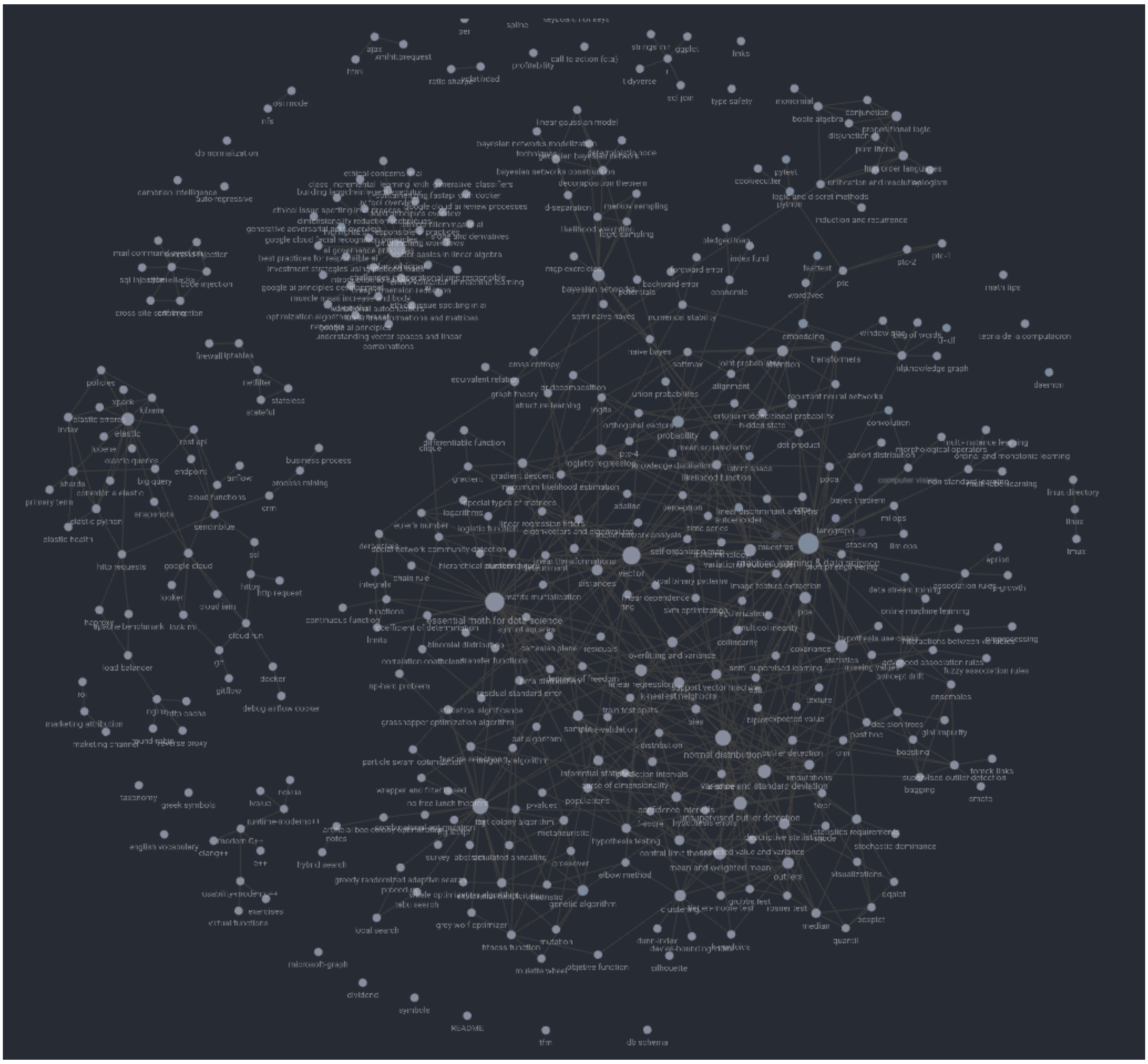

Figura 2: Grafo que captura la relación de las notas tras años de toma en *Obsidian*

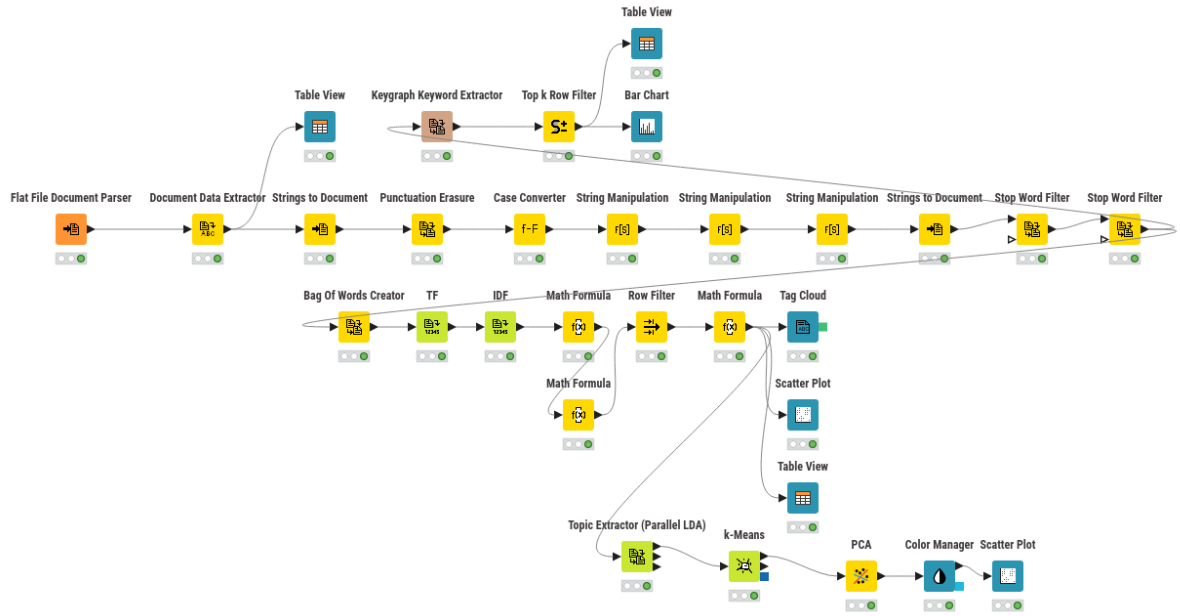

Figura 3: *Workflow* de procesamiento o *ETL* en *KNIME*

Rows: 350 | Columns: 3 Table | Statistics

	#	RowID	Document *Text document*	Title *String*	Document body text *String*
☐	1	Row0	"/home/migue8gl/Obsidian/Conceptos/metaheuristic.md"	/home/migue8gl/Obsidian/Conceptos/metaheuristic.md	- #metaheuristic | #optimization | #exploration | #exploitation - Una metah
☐	2	Row1	"/home/migue8gl/Obsidian/Conceptos/teoria de la computacion.md"	/home/migue8gl/Obsidian/Conceptos/teoria de la computacion.md	\section{Teoría de la computación} La teoría de la computación es un cam
☐	3	Row2	"/home/migue8gl/Obsidian/Conceptos/hypothesis testing.md"	/home/migue8gl/Obsidian/Conceptos/hypothesis testing.md	- #hypothesistesting - Cuando hacemos test sobre hipótesis, tenemos dos
☐	4	Row3	"/home/migue8gl/Obsidian/Conceptos/k-nearest neighbors.md"	/home/migue8gl/Obsidian/Conceptos/k-nearest neighbors.md	- #knn | #ml ## Introducción - El algoritmo de k vecinos más cercanos (**k
☐	5	Row4	"/home/migue8gl/Obsidian/Conceptos/distances.md"	/home/migue8gl/Obsidian/Conceptos/distances.md	- #distances | #clustering | #ml # Distancia Euclidea - La distancia euclide
☐	6	Row5	"/home/migue8gl/Obsidian/Conceptos/reverse proxy.md"	/home/migue8gl/Obsidian/Conceptos/reverse proxy.md	- #reverseproxy | #proxyinverso - Es una aplicación que se encuentra delan
☐	7	Row6	"/home/migue8gl/Obsidian/Conceptos/whale optimization algorithm.md"	/home/migue8gl/Obsidian/Conceptos/whale optimization algorithm.md	- #mh | #wao | #whaleoptimization ## Inspiración - Simula el comportamie
☐	8	Row7	"/home/migue8gl/Obsidian/Conceptos/ppca.md"	/home/migue8gl/Obsidian/Conceptos/ppca.md	- #pca | #probabilistic | #ml | #ppca # Concepto - La [[pca|PCA]] probabilis
☐	9	Row8	"/home/migue8gl/Obsidian/Conceptos/python.md"	/home/migue8gl/Obsidian/Conceptos/python.md	- #python - [[ptc]] - Asignatura de programación técnica y científica. - [[pyte
☐	10	Row9	"/home/migue8gl/Obsidian/Conceptos/number theory.md"	/home/migue8gl/Obsidian/Conceptos/number theory.md	- #numbertheory - La teoría de números es un área de las matemáticas do
☐	11	Row10	"/home/migue8gl/Obsidian/Conceptos/marketing attribution.md"	/home/migue8gl/Obsidian/Conceptos/marketing attribution.md	- #marketing | #business | #attribution # Concepto - La atribución es funda
☐	12	Row11	"/home/migue8gl/Obsidian/Conceptos/rvalue.md"	/home/migue8gl/Obsidian/Conceptos/rvalue.md	- #rvalue - En C++, un rvalue es una expresión que se evalúa en un valor, pe
☐	13	Row12	"/home/migue8gl/Obsidian/Conceptos/f-score.md"	/home/migue8gl/Obsidian/Conceptos/f-score.md	- #fscore # Concepto - El **F-score** (también conocido como **F1-score

Figura 4: Tabla inicial de los datos textuales en crudo

Todo ello vuelve a convertirse a tipo de dato *document* con un nodo concreto. Después se utilizan dos nodos de filtrado de *stop words*, que son palabras carentes de potencial significado semántico, tanto en español como en inglés, ya que los apuntes contienen notas en ambos idiomas.

3.2. Extracción de palabras clave

Posterior a la limpieza del texto y paralelo a otros procesos, se utiliza un nodo extractor de palabras clave. Este nodo funciona usando un método basado en grafos que, en primera instancia, selecciona un conjunto de términos de alta frecuencia y los añade al grafo como nodos, para después calcular la fuerza de asociación entre términos.

Se filtran los k términos más importantes dados este algoritmo de minería de texto y se visualizan. En las figuras 5 y 6 se pueden visualizar las 15 palabras claves más relevantes extraídas. Cabe destacar que todas pertenecen al mismo sub-conjunto de apuntes pertenecientes a lógica y métodos discretos. El término más relevante es "conjunto". Este término es muy utilizado y referenciado entre documentos, tiene sentido que sea el más relevante, así como tiene también cierto sentido que lógica y métodos discretos sea un sub-conjunto de documentos que contiene los términos más relevantes, ya que en estos textos se relatan y explican detalladamente conceptos relacionados con los grafos (las *keywords* detallan explícitamente conceptos de grafos, como por ejemplo los vértices), los cuales son muy referenciados y usados en otros documentos, incluso en aquellos con temáticas totalmente opuestas, pues se hace referencia y se explican métodos que hacen uso de esta estructura.

3.3. TF-IDF y descubrimiento de grupos

Partiendo del nodo final del flujo de limpieza y pre-procesado, se continua con el flujo de técnicas de descubrimiento de grupos. Se coloca un nodo cuya finalidad es convertir texto en una representación numérica llamada "Bolsa de Palabras"(*Bag of Words*). Esta técnica transforma documentos de texto en vectores numéricos contando la frecuencia de cada palabra en el documento. No preserva el orden de las palabras, sino que crea una "bolsa"que contiene recuentos de palabras.

Después se calcula la frecuencia de términos, midiendo cuántas veces aparece cada palabra en un documento. La frecuencia de términos es un componente importante para determinar la relevancia de las palabras en un documento. Cuanto más frecuente sea una palabra en un documento específico, más importante podría ser para ese documento. Sin embargo, esta medida por sí sola puede sobrevalorar términos comunes. Por ello se utiliza esta métrica en conjunto con la frecuencia inversa de documentos.

El siguiente nodo calcula la frecuencia inversa de documentos, una medida que otorga mayor peso a las palabras que aparecen en pocos documentos. El *IDF* reduce la impor-

Rows: 15 | Columns: 3

RowID	Keyword *Term*	Score *Number (double)*	Document *Text document*
3418	\forall[]	383	"firstorderlanguages lenguajesdeprimerord
3417	forma[]	388	"firstorderlanguages lenguajesdeprimerord
1812	vértices[]	398	"graphtheory teoríadegrafos lmd generalida
3416	fórmula[]	435	"firstorderlanguages lenguajesdeprimerord
1811	1[]	449	"graphtheory teoríadegrafos lmd generalida
1810	grafos[]	449	"graphtheory teoríadegrafos lmd generalida
3415	estructura[]	493	"firstorderlanguages lenguajesdeprimerord
1809	grafo[]	515	"graphtheory teoríadegrafos lmd generalida
1808	sucesión[]	519	"graphtheory teoríadegrafos lmd generalida
3414	\exists[]	526	"firstorderlanguages lenguajesdeprimerord
1807	lados[]	535	"graphtheory teoríadegrafos lmd generalida
3413	conjunto[]	536	"firstorderlanguages lenguajesdeprimerord
1806	conjunto[]	537	"graphtheory teoríadegrafos lmd generalida
3412	\alpha[]	539	"firstorderlanguages lenguajesdeprimerord
3411	cuantificadores[]	545	"firstorderlanguages lenguajesdeprimerord

Figura 5: Tabla con las 15 palabras clave más importantes

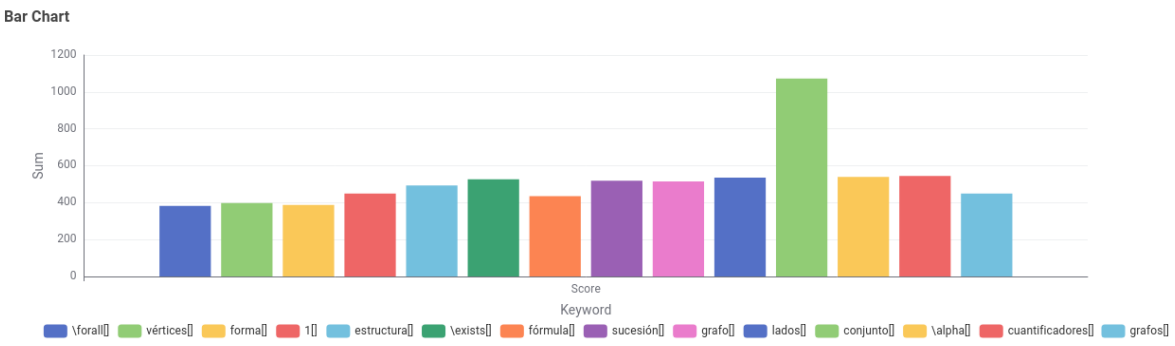

Figura 6: Gráfico de barras con los 15 términos más importantes

tancia de términos que son demasiado comunes en la colección de documentos. Multiplica una pequeña constante para términos que aparecen en muchos documentos, mientras que asigna un peso mayor a términos que aparecen en pocos documentos, destacando así las palabras que son más distintivas.

Dadas ambas métricas, se calcula una nueva conocida como **TF-IDF**, un estadístico que refleja la importancia de una palabra en un documento dentro de una colección. Esta nueva métrica combina las bondades de los dos anteriormente explicadas.

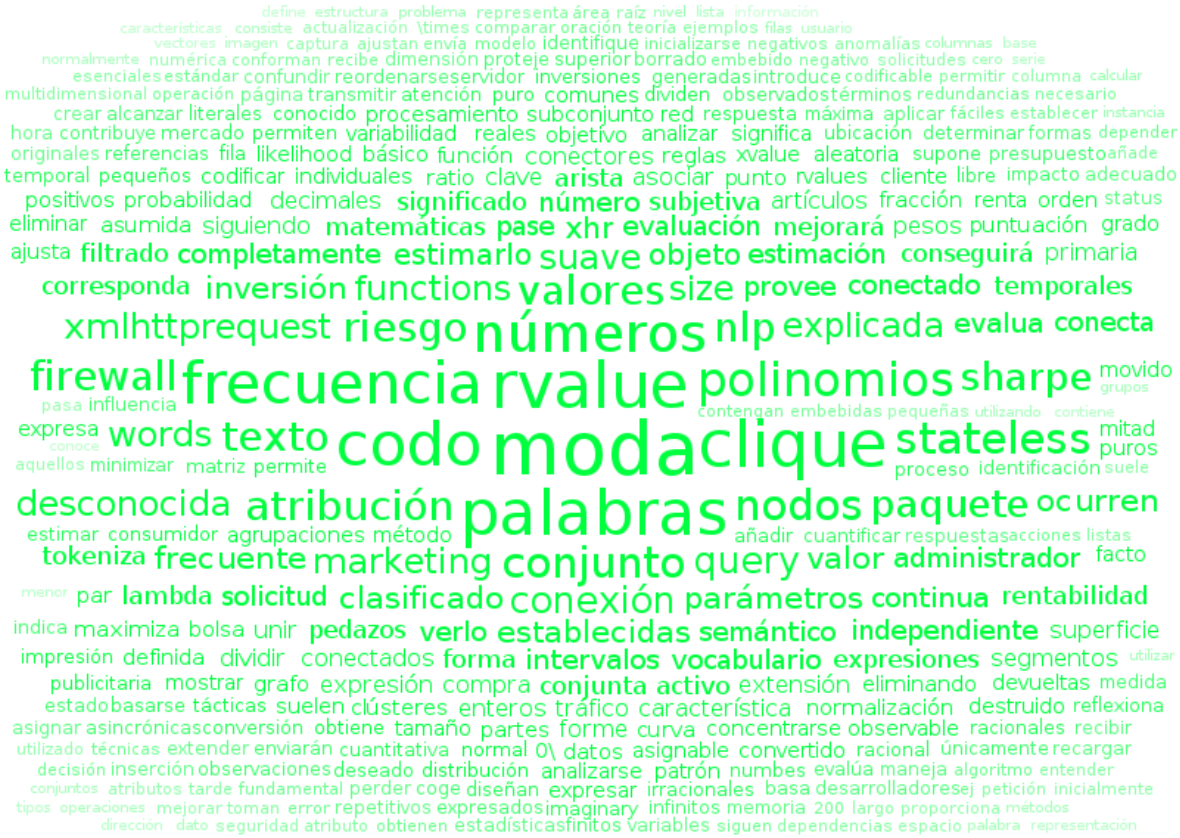

Figura 7: Núbe de palabras por **TF-IDF**

Posterior a este cálculo, se filtran aquellos términos cuyo calculo haya resultado en valores no numéricos. En la visualización de la figura 7 se observa la nube de términos generada, donde el tamaño de cada palabra representa su frecuencia o importancia. Se puede observar también la tabla con los datos correspondientes en la figura 8.

Los términos más grandes son algunos muy genéricos como *palabras*, *números*, *conjunto*. En cambio, también se resaltan otros más relevantes como *moda* (estadístico), *rvalue* (concepto de `c++`), *polinomios*, *clique*, etc. Todos estos términos orientados a un ámbito *STEM* (*Science, technology, engineering and mathematics*).

Paralelamente se aplica el nodo extractor de *topics*, que aplica *Latent Dirichlet Allocation* (**LDA**), un modelo probabilístico que identifica temas ocultos en colecciones de

RowID	NewText *Text document*	Term *Term*	TF rel *Number (double)*	IDF *Number (double)*	TF-IDF *Number (double)*
Row20	"terminar apuntes"	apuntes[]	0.5	2.071	1.035
Row20	"terminar apuntes"	terminar[]	0.5	1.947	0.973
Row40	"sql join sql joinexamplepng"	join[]	0.25	2.246	0.561
Row38	"links pinecone dochttpsdocspineconeioguid	apuntes[]	0.25	2.071	0.518
Row22	"iptables iptables herramienta cortafuegos es	iptables[]	0.181	2.545	0.46
Row27	"lvalue lvalue expresión refiere objeto ubicaci	lvalue[]	0.176	2.545	0.449
Row30	"cartesianplane xyplane coordinateplane plan	eje[]	0.286	1.516	0.433
Row27	"taxonomy taxonomía ciencia práctica clasifi	taxonomía[]	0.167	2.545	0.424
Row27	"primaryterm término primario propiedad asi	primario[]	0.217	1.851	0.402
Row12	"cyberattacks inyecciones vulnerabilidades in	injection[]	0.167	2.246	0.374
Row60	"mode moda definición moda valor frecuente	moda[]	0.176	2.071	0.365
Row26	"cplusplus librerías eigenhttpseigentuxfamily	moderno[]	0.143	2.545	0.364
Row37	"firewall cortafuegos firewall cortafuegos ese	cortafuegos[]	0.15	2.246	0.337
Row33	"pureliteral literalpuro literal puro conjunto clá	literal[]	0.182	1.851	0.337
Row27	"taxonomy taxonomía ciencia práctica clasifi	clasificación[]	0.25	1.288	0.322

Figura 8: Tabla de datos tras calcular **TF-IDF**

documentos. **LDA** asume que cada documento contiene una mezcla de temas, y cada tema es una distribución de probabilidad sobre palabras. El procesamiento paralelo permite aplicar este algoritmo a grandes volúmenes de datos de manera eficiente.

Este se utiliza junto a la métrica **TF-IDF** para identificar grupos mediante el algoritmo *k-means*. Tras un pequeño post-procesado (*PCA*, colores por *cluster*, prueba de valores para k, etc.) se obtienen los siguientes grupos (figura 9).

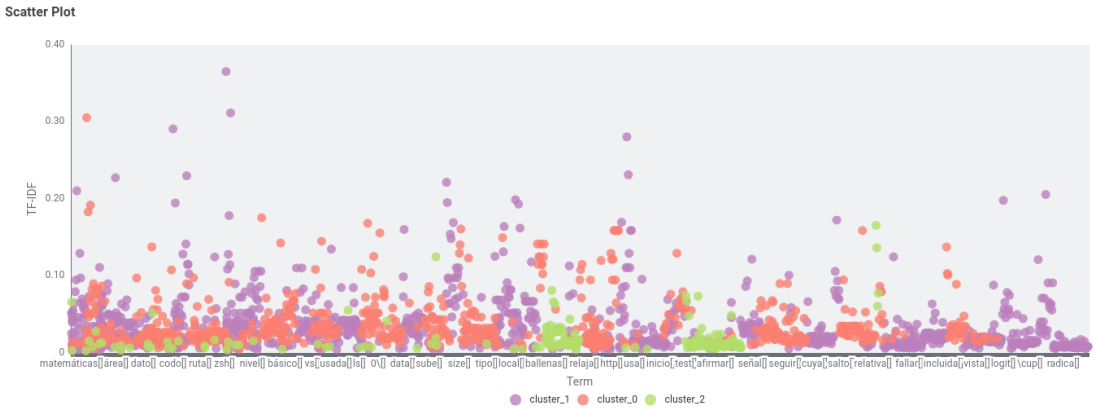

Figura 9: Cluster obtenido de los términos

- **cluster_0**: Grafos, redes y relacionado. También se incluyen algunos conceptos de inversión.
- **cluster_1**: Conceptos de programación avanzada (principalmente C++), matemática aplicada y estructuras de datos.

- **cluster_2**: Metaheurísticas, algoritmos de optimización.

El *cluster* número 2 es el más pequeño, se ha especializado mucho en algoritmos metaheheurísticos, de los cuales el estudiante tiene una gran cantidad de apuntes en relación a la temática de su *TFG*.

Según el análisis anterior, los términos más relevantes tenían que ver con teoría de grafos, lo que se ve reflejado en el grupo 0, que es además el que más elementos tiene y en relación a esa temática.

Por último, parece que el grupo 1 grupa todos los conceptos más relacionados con matemáticas, *machine learning* y programación.