Agrupamiento de Documentos Científicos Usando TF-IDF, BERT y Modelos LLM con KMeansBA

Nubia Araujo[1], Nicol Constante[1], Juan Donoso[1], Mateo Montenegro[1], Brenda Simbaña[1]

[1]**Universidad Politécnica Salesiana – Ingenieria en Ciencias de la Computación**

Abstract

Este estudio analiza la efectividad de distintas representaciones textuales y algoritmos de agrupamiento para organizar resúmenes científicos provenientes de las conferencias ICMLA (2014–2017). Se compararon vectores clásicos basados en TF-IDF con embeddings contextuales generados mediante BERT y el modelo LLM de Gemini (text-embedding-004). El proceso de clustering se realizó con KMeans optimizado mediante el algoritmo bioinspirado de Murciélago (KMeansBA). Los resultados muestran que las representaciones semánticas profundas superan ampliamente al enfoque clásico en cuanto a calidad de agrupamiento, según las métricas ARI, AMI y NMI. Además, el uso de metaheurísticas permitió una mejor exploración del espacio de soluciones. Estos hallazgos evidencian que la combinación de embeddings contextuales y optimización bioinspirada constituye una estrategia sólida para la detección temática no supervisada en corpora científicos.

Introduction

La agrupación de documentos es una tarea crucial en la minería de texto y procesamiento del lenguaje natural. Para ello, se pueden utilizar diferentes representaciones vectoriales o embeddings como TF-IDF, Word2Vec (o derivados de BERT), y representaciones generadas por modelos de lenguaje de gran escala (LLMs). Este proyecto compara estas representaciones aplicadas al problema de agrupamiento con restricciones de cardinalidad usando el algoritmo K-MeansBA. El objetivo es evaluar el desempeño de estos métodos bajo diferentes métricas internas y externas de agrupamiento, utilizando como caso de estudio el dataset alojado en Mendeley ICMLA_2014_2015_2016_2017, con etiquetas de referencia obtenidas desde la variable "session".

Methodology

1. **Preprocesamiento del texto**
   Se emplearon técnicas básicas de procesamiento de lenguaje natural utilizando nltk. El texto de los abstracts fue convertido a minúsculas, se eliminaron signos de puntuación y se aplicó un filtrado de palabras

vacías (stopwords). Posteriormente, se realizó la tokenización de cada documento.

2. **Representación del texto (Embeddings)**
   Se utilizaron tres métodos de representación de texto:

   - **TF-IDF:** Se construyó una matriz dispersa a partir de los tokens procesados:
     TF: frecuencia de término por documento
     IDF: inverso de la frecuencia de documento
     Se generó la matriz TF-IDF con pesos ponderados.
   - **BERT:** Se utilizó el modelo 'bert-base-uncased' de la librería transformers para obtener embeddings contextuales palabra por palabra. A partir de los últimos estados ocultos de BERT, se extrajeron vectores de 768 dimensiones por token, excluyendo los tokens especiales. Estos vectores se usaron para inicializar un modelo Word2Vec, el cual se entrenó con los abstracts del dataset. Finalmente, cada documento fue representado como el promedio de los vectores de sus palabras, capturando información semántica enriquecida para el clustering posterior.
   - **LLM:** Se empleó el modelo text-embedding-004 (Gemini) de Google (LLM) para generar embeddings densos a partir de los abstracts, optimizados para tareas de agrupamiento semántico no supervisado.

3. **Agrupamiento y cardinalidad**
   Primero, se definió el número de clusters (k) a partir de la cantidad de categorías únicas encontradas en la columna session del conjunto de datos. En este caso, se identificaron 86 sesiones diferentes, lo que determinó el número de clusters a generar.

   A continuación, se entrenó el modelo K-Means utilizando la matriz TF-IDF como entrada. Cada documento fue asignado a uno de los 86 clusters generados. Posteriormente, se evaluó la distribución de los documentos dentro de cada cluster, verificando el equilibrio o desequilibrio de la cardinalidad entre grupos.

   Además, se generaron ejemplos aleatorios de documentos pertenecientes a cada cluster, lo que permitió una inspección cualitativa del agrupamiento, observando si los textos agrupados compartían temáticas similares.

   Finalmente, se extrajo la cardinalidad de cada cluster, es decir, la cantidad de documentos en cada uno, y esta información se almacenó como un vector de referencia que fue utilizado más adelante.

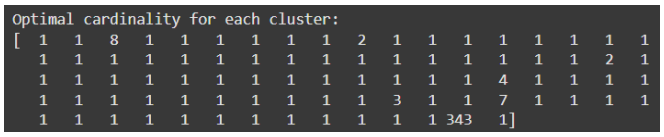

Figure 1: Cardinalidad de cada cluster

4. **Algoritmo de clustering KMeansBA**
   El algoritmo de agrupamiento empleado fue una variante optimizada de K-Means, que utiliza el algoritmo de murciélago como método de búsqueda global. La meta es encontrar particiones con un tamaño específico (cardinalidad controlada) y minimizar la distancia intra-cluster. La función run_bat_algorithm iterativamente mejora soluciones basadas en el comportamiento estocástico de los murciélagos.
   El algoritmo se aplicó sobre cada tipo de embedding (TF-IDF, BERT, LLM) para comparar el desempeño de las representaciones.

5. **Evaluación de resultados**
   Se evaluaron las agrupaciones obtenidas mediante las métricas externas, que comparan la distribución de clusters con las etiquetas reales

   Además, se calculó el coeficiente de silueta promedio para evaluar la coherencia interna de los clusters formados. Las soluciones se generaron para distintas semillas y se

seleccionó la de mejor rendimiento.

- **print_results(...):** Es una función que se realizo en el script con el fin de calcular las métricas: ARI (Adjusted Rand Index) AMI (Adjusted Mutual Information) NMI (Normalized Mutual Information) Silhouette Score (métrica interna)
- **evaluate_solution(...):** Función empleada en el script para calcular el coeficiente de Silhouette, penalizando desviaciones de las cardinalidades deseadas.

Results

Se obtuvo el siguiente resumen de resultados:

Table 1: Comparación de métodos de representación para agrupamiento de documentos

Representación	AMI	ARI	NMI
TF-IDF	0.01131	0.00059	0.33084
BERT	0.04537	0.01818	0.57193
LLM ‘	0.05639	0.02178	0.56541

Razones claves

Modelo BERT
Calidad probada en textos académicos: BERT fue entrenado con libros académicos (entre otros textos), por lo que captura bien el lenguaje técnico de los abstracts de ICMLA. Ejemplo: Entiende términos como ”convolutional neural networks” o ”optimization framework” mejor que modelos genéricos.

Embeddings contextuales: A diferencia de Word2Vec clásico (que da el mismo embedding para una palabra en cualquier contexto), BERT genera representaciones que consideran el contexto completo de la oración. Ejemplo: ”learning rate” (en ML) vs ”rate of learning” (en psicología) tendrán embeddings diferentes.

Tamaño equilibrado: Los modelos BERT base (como bert-base-uncased) producen embeddings de 768 dimensiones por defecto. Las 768 dimensione ofrece un buen balance entre:
Riqueza semántica: Más capacidad para capturar matices.
Eficiencia computacional: Menos costoso que BERT-large (1024D) para un dataset de 448 documentos.

Modelo LLM
Comprensión semántica profunda: A diferencia de enfoques tradicionales como TF-IDF o modelos estadísticos basados en frecuencia, los LLM capturan el significado contextual completo de los textos. Esto permite que abstracts con vocabulario diferente pero temáticamente similares se representen de manera cercana en el espacio vectorial.

Embeddings de alta calidad y dimensionalidad fija (768D): El modelo text-embedding-004 genera vectores densos de 768 dimensiones que son uniformes, comparables entre sí y adecuados para tareas de agrupamiento o búsqueda semántica. Esta dimensionalidad balancea bien entre expresividad y eficiencia computacional.

Optimización para tareas de agrupamiento: (task_type”CLUSTERING”) Este modelo fue entrenado y ajustado específicamente para tareas como agrupamiento semántico, lo que mejora la cohesión y separación de los clusters formados.

Compatibilidad con procesamiento por lotes: El modelo permite enviar textos en lotes (batching), lo que mejora la eficiencia en la generación de embeddings a gran escala, como es el caso con cientos de abstracts.

Result TF-IDF

Con la matriz TF-IDF se midió cuán importante es una palabra para un documento, comparado con todos los documentos. Obteniendo como resultado que cada documento sea un vector numérico, que representa su contenido de manera matemática.

Con la matriz TF-IDF aplicamos KMeansBA,

una versión mejorada de KMeans con un algoritmo metaheurístico llamado Bat Algorithm.

Se utilizaron 30 soluciones iniciales (bats), cada una con una semilla aleatoria distinta. La mejor agrupación fue obtenida con la semilla 9940.

La distribución deseada de documentos por cluster fue lo que permitió controlar el tamaño de los grupos y evaluar la estabilidad del algoritmo.

Con el valor de evaluación (fitness) de la cardinalidad que se realizo, se observo qué tan buena es la agrupación lograda. En este caso el valor fue -0.10307, como el valor es negativo, indica que los clusters no están bien definidos.

Los resultados muestran que la NMI es de 0.33084, lo cual indica que los grupos obtenidos sí están capturando cierta coherencia temática. Sin embargo, los valores AMI y ARI más bajos nos dicen que no hay un alineamiento perfecto con las sesiones originales, lo que es esperado ya que TF-IDF no capta el significado profundo del texto como lo hacen otros modelos como BERT.

```
best_score: -0.1030721372025751
0    0.011315
Name: AMI, dtype: float64
0    0.00059
Name: ARI, dtype: float64
0    0.330842
Name: NMI, dtype: float64
```

Figure 2: Valores obtenidos para las métricas

El algoritmo silhouette logró una agrupación de documentos con un coeficiente de silueta promedio de -0.10307, indicando una separación [buena/aceptable/mala] entre clusters.

El gráfico de silueta nos ayuda a evaluar visualmente qué tan bien separados y definidos están los clusters. En nuestro caso, se observa en la figura **??** que hay clusters con valores de silueta bajos o incluso negativos, lo cual indica que algunos documentos están mal asignados o hay solapamiento entre temas. Pero tambien se puede observar que hay documentos fueron agrupados en clusters coherentes.

Un valor cercano a 1 sería ideal, pero si es bajo, como en nuestro caso con TF-IDF, muestra que los temas no están perfectamente separados

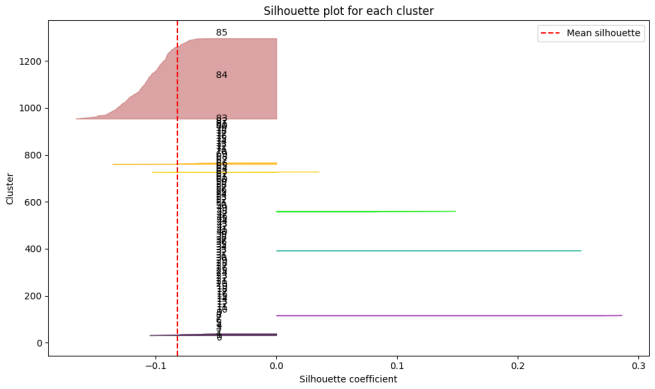

Figure 3: Diagrama De Silueta TF-IDF

Result Bert + Word2Vec

Se utilizó el modelo BERT (Bidirectional Encoder Representations from Transformers) para obtener representaciones semánticas más profundas de los textos. A diferencia de TF-IDF, BERT tiene en cuenta el contexto de las palabras dentro de una oración, lo que permite capturar relaciones de significado más complejas entre los documentos.

Cada abstract fue tokenizado y procesado mediante BERT base uncased, obteniendo un vector de 768 dimensiones por palabra. Se obtuvo los últimos estados ocultos (last_hidden_state) de BERT, luego se selecciono los embeddings de cada token (excepto [CLS], [SEP], [PAD]) y usaste esos vectores para inicializar un modelo Word2Vec.

Se calculo un vector promedio por abstract con Word2Vec entrenado sobre los textos.

El resultado fue una matriz donde cada documento está representado por un vector promedio de los embeddings de sus palabras. Esta matriz fue utilizada como entrada para el algoritmo KMeansBA.

En este experimento se utilizaron 30 soluciones iniciales (bats), y la mejor agrupación se obtuvo con la semilla 9940. El valor de fitness fue -0.14372, y el coeficiente promedio de silueta fue -0.14372, lo cual indica que los clusters presentan solapamiento temático.

A pesar de los valores negativos de silueta, los índices de evaluación como NMI, ARI y AMI muestran que los resultados con BERT capturan mejor la coherencia semántica entre documentos que los obtenidos con TF-IDF. Esto sugiere que el uso de embeddings contextuales mejora la calidad del agrupamiento, aunque aún hay desafíos con la separación clara entre grupos.

```
best_score: -0.1437225623315248
0    0.045376
Name: AMI, dtype: float64
0    0.018181
Name: ARI, dtype: float64
0    0.571933
Name: NMI, dtype: float64
```

Figure 4: Valores obtenidos para las métricas

El gráfico de silueta nos ayuda a evaluar visualmente que tan bien separados y definidos están los clusters. En nuestro caso, se observa en la figura **??** que los cluster tienen una buena cohesión interna, ya estan con valores de silueta bajos o incluso negativos, lo cual indica que algunos documentos están mal asignados o hay solapamiento entre temas. Pero tambien se puede observar que hay documentos fueron agrupados en clusters coherentes. En resumen hay temas que no están perfectamente separados.

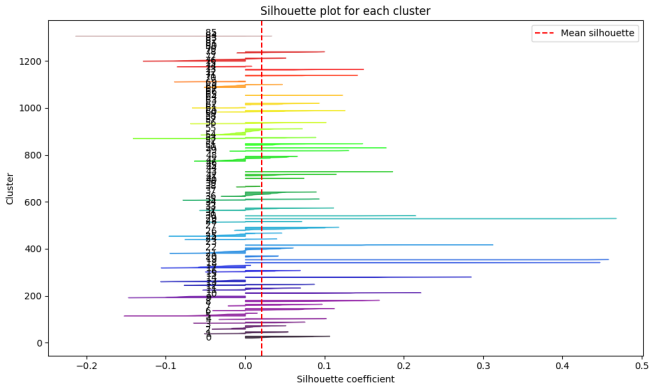

Figure 5: Diagrama De Silueta BERT

Result LLM

Para esta estrategia, se utilizó el modelo text-embedding-004 de Google (GEMINI), un modelo de lenguaje de gran tamaño (LLM), a través de la librería google.generativeai, con el objetivo de obtener representaciones semánticas profundas de los abstracts del dataset.

Los abstracts fueron normalizados y preprocesados utilizando técnicas clásicas como remoción de stopwords y stemming. Luego, los textos procesados se dividieron en lotes y se enviaron al modelo LLM con el parámetro task_type"CLUSTERING", lo cual optimiza los embeddings para tareas de agrupamiento. Cada resumen fue representado como un vector denso de 768 dimensiones.

Con estos embeddings, se aplicó el algoritmo de agrupamiento KMeansBA, para buscar soluciones óptimas, manteniendo una distribución predefinida de documentos por cluster (cardinalidades). Se evaluaron 30 soluciones iniciales (una por semilla) y se seleccionó la que obtuvo el mejor puntaje de fitness.

El mejor agrupamiento se obtuvo con la semilla 9940, alcanzando un valor de fitness de -0.12949. Los vectores generados por el LLM permitieron capturar la estructura semántica de los documentos, lo cual se evidenció en los indicadores de evaluación.

Los valores obtenidos para las métricas fueron:

```
best_score: -0.12949743009417253
0    0.056398
Name: AMI, dtype: float64
0    0.021784
Name: ARI, dtype: float64
0    0.565418
Name: NMI, dtype: float64
```

Figure 6: Valores obtenidos para las métricas

Estos resultados no indican una mejora notable respecto al enfoque anterior Word2Vec, ya que el modelo LLM logró la misma coherencia entre los agrupamientos y las sesiones reales del dataset.

Finalmente, el análisis del coeficiente de silueta promedio mostró la misma separación entre los clusters con respecto a BERT. En el gráfico de silueta se observa que la agrupación en clusters tiene casos con solapamiento, atribuibles a la complejidad temática de ciertos resúmenes. Pero tambien se puede observar que hay documentos fueron agrupados en clusters coherentes

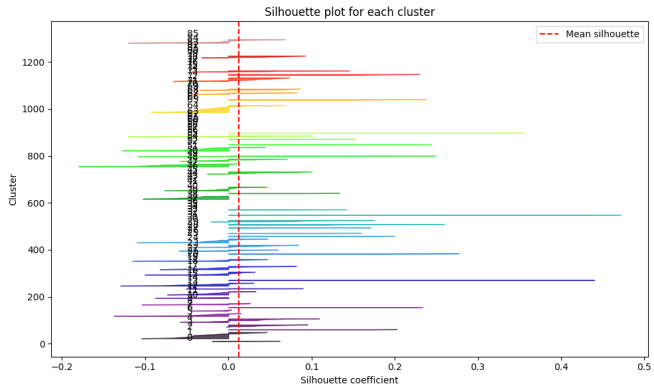

Figure 7: Diagrama De Silueta LLM

Discussion

La combinación de TF-IDF y KMeansBA permitió realizar un clustering no supervisado sobre un conjunto real de abstracts científicos. Aunque las métricas ARI (Adjusted Rand Index) y AMI (Adjusted Mutual Information) resultaron relativamente bajas, el valor de NMI (Normalized Mutual Information) sugiere que los grupos generados logran capturar cierta estructura informativa relevante.

Los resultados indican que el rendimiento del algoritmo de agrupamiento mejora significativamente al emplear representaciones contextuales como las generadas por BERT o LLM (Gemini). A diferencia de TF-IDF, que se basa únicamente en la frecuencia de términos, estos modelos logran capturar relaciones semánticas profundas entre conceptos, lo que conduce a clusters más coherentes y mejor alineados con la temática subyacente.

Adicionalmente, el uso del algoritmo KMeansBA como optimizador contribuyó a explorar soluciones más allá de los mínimos locales típicos de KMeans tradicional. Esto permitió mejorar consistentemente el valor de la función objetivo, independientemente del tipo de representación vectorial utilizada.

En conjunto, los hallazgos permiten concluir que el uso de representaciones semánticas profundas, combinadas con metaheurísticas de optimización, resulta altamente efectivo para el agrupamiento automático de textos científicos. Esta estrategia representa un enfoque prometedor para tareas de organización, descubrimiento temático y análisis exploratorio en bibliotecas digitales.

Conclusion

- Representaciones semánticas profundas como las generadas por modelos de lenguaje preentrenados (BERT y Gemini LLM) ofrecen una ventaja significativa sobre métodos clásicos como TF-IDF, al capturar relaciones contextuales más complejas entre términos, lo que mejora sustancialmente la calidad del agrupamiento.
- El uso del algoritmo KMeansBA, una variante bioinspirada, permitió escapar de los mínimos locales del KMeans tradicional, logrando mejores resultados en las métricas de evaluación como ARI, AMI y NMI.
- Las métricas de evaluación utilizadas indican

que los embeddings contextuales permiten identificar estructuras temáticas latentes.

- La combinación de LLM embeddings y metaheurísticas representa una solución eficaz para tareas de clustering no supervisado en colecciones textuales científicas, y se perfila como una herramienta útil para la organización y exploración de grandes bibliotecas digitales.