Ranking de Dimensiones en Vectores Densos para Recuperación Eficiente

Tomás Delvechio [0009-0005-2589-3436][1,3], Esteban Rissola [0000-0001-7072-4096][1], and Gabriel Tolosa [0000-0001-8237-7554][1,2]

[1] Depto. de Ciencias Básicas, Universidad Nacional de Luján, Argentina
[2] CIDETIC, Universidad Nacional de Luján, Argentina
[3] Maestría en Expl. de Datos y Desc. del Conocimiento, FCEN, UBA. Argentina
{tdelvechio, earissola, tolosoft}@unlu.edu.ar

Resumen La Recuperación de Información sobre colecciones de millones de documentos es una tarea computacionalmente intensiva. La aparición de representaciones densas (*embeddings*) posibilita construir vectores de centenas de dimensiones, convirtiendo al problema de resolver una consulta en una búsqueda de vectores más cercanos. Como es razonable considerar que no todas las dimensiones de estos *embeddings* son igualmente importantes, se propone *rankear* la importancia de las mismas para facilitar su poda según un requerimiento de efectividad objetivo. En este trabajo se evalúan diversos métodos para la poda. A partir de modelos ampliamente utilizados para generar *embeddings* y una colección de documentos y consultas de referencia en la comunidad, los experimentos muestran que es posible reducir el tamaño de los vectores hasta un 50 % manteniendo hasta un 90 % de efectividad, mejorando así la eficiencia.
Palabras clave: vectores densos, neural IR, ranking de dimensiones, eficiencia.

Dense Vectors Dimensions Ranking for Efficient Retrieval

Resumen Information retrieval over collections of millions of documents is a computationally intensive task. The emergence of dense representations (embeddings) enables the construction of vectors with hundreds of dimensions shifting the retrieval task into a nearest-neighbour vector search problem. We hypothesize that not all the embedding dimensions are equally important for the retrieval task and, therefore some could be pruned. In this paper, we propose to rank embedding dimensions based on their importance and evaluate different pruning methods following an objective effectiveness requirement. Based on widely used models for generating embeddings and well-known document collections, our experiments show that it is possible to reduce the size of vectors by up to 50 % while maintaining the effectiveness of up to 90 %, thus improving efficiency.
Keywords: dense vectors, neural IR, dimensions ranking, efficiency.

1. Introducción

La recuperación de texto es una tarea fundamental en aplicaciones basadas en el lenguaje, como la recuperación de información (RI). Esta tarea involucra un escenario en el que los documentos y las consultas están expresadas en texto libre no estructurado (o en lenguaje natural), y comparten una representación común que permite establecer su similitud. Un ejemplo de aplicaciones de RI son los motores de búsqueda web, que procesan millones de documentos para responder a las consultas de los usuarios con estrictas restricciones de tiempo. Este requisito exige gestionar representaciones de texto que sean simples y eficientes, careciendo de *cierta* comprensión de la semántica del texto.

El uso de redes neuronales en RI (*Neural IR*) comenzó a reducir la brecha semántica e indujo mejoras positivas en la efectividad, pero con implicaciones en la eficiencia. Uno de los cambios es la representación lógica del texto. En lugar de calcular y almacenar estadísticas de palabras (por ejemplo, frecuencia), se propone entrenar una red neuronal que mapea cada palabra w dentro de un corpus C de texto en un espacio latente multidimensional (de dimensión d) (Bengio et al., 2003). Este proceso produce un modelo de lenguaje (LM), que apunta a modelar la probabilidad generativa de secuencias de palabras, para predecir las probabilidades de futuras apariciones. Para cada objetivo $w \in C$, el procedimiento de entrenamiento considera las palabras circundantes. Luego, la representación lógica de cada palabra se convierte en un vector $\vec{v}$ de dimensión d, que generalmente se llama *embedding* de w. Cada dimensión de este vector corresponde a un valor en $\mathbb{R}$ que pondera su importancia.

Hoy en día, el marco conceptual más común para organizar la recuperación basada en vectores densos es la arquitectura de *bi-encoders* en la cual tanto documentos como consultas se representan mediante vectores densos usando un modelo dado (Karpukhin et al., 2020). Luego, el problema se puede expresar como una búsqueda de los vecinos (documentos) más cercanos al vector de la consulta. Considerando millones de documentos y vectores densos de decenas de dimensiones, comparar documentos y consultas es una tarea computacionalmente intensiva, planteando desafíos para su implementación e inquietudes sobre su impacto ambiental y económico debido al alto consumo de energía.

Sin embargo, es razonable considerar que no todas las dimensiones de los *embeddings* son igualmente importantes para la tarea de recuperación. Algunos autores proponen que los datos del mundo real representados en un espacio de alta dimensión, d, concentran su masa de probabilidad en un subconjunto M de dimensionalidad mucho menor ($d_M \ll d$) (Bengio et al., 2014). Bajo este supuesto, es posible considerar el uso de un número reducido de dimensiones para podar los vectores y acelerar las comparaciones, mejorando así la eficiencia y la escalabilidad del sistema. Existen asimismo propuestas de reducir el espacio de dimensiones de forma dependiente de la consulta, lo que requiere estimar de antemano cuáles son *útiles* para una consulta dada (Faggioli et al., 2024).

En este contexto, se proponen y evaluan diferentes métodos para rankear las dimensiones en vectores densos de forma independiente de la consulta. Se estima

la contribución de cada dimensión para luego ordenarlas descendentemente y facilitar la poda según un requerimiento de efectividad objetivo. Se muestra que es posible reducir el tamaño de los vectores en el momento de la indexación hasta un 50 % sin una degradación significativa del rendimiento. Para la creación de los vectores se emplean modelos para *embeddings* ampliamente utilizados para recuperación y colecciones de pruebas usadas en la comunidad.

2. Ranking de Dimensiones

La selección de dimensiones como estrategia para reducir la complejidad se aborda comúnmente en problemas de aprendizaje automático (AA) (Obaid et al., 2019). Sin embargo, aquí se propone *estimar la importancia de la dimensión* con el objetivo de obtener un ranking de acuerdo a su utilidad para la recuperación (*retrieval*). Se enfatiza la idea de mantener las características originales (los métodos no transforman los datos), para usar un número adecuado a una necesidad. Se consideran los siguientes métodos:

Varianza (*SVAR*): Método clásico utilizado en el AA para elegir características/columnas de un conjunto de datos (Li et al., 2017). Se calcula la varianza de cada dimensión de los vectores de documentos y se las ordena descendentemente. Este criterio se fundamenta en la suposición de que una mayor varianza conlleva un mayor poder de discriminación, lo que puede ser útil para la correspondencia documentos y consultas.

Correlación (*SCOR*): Aquí se calcula la matriz de correlaciones (ρ) entre todas las columnas. Luego, se suman las columnas (correlación total de cada dimensión) y se ordenan ascendentemente. La idea es capturar primero las dimensiones que están menos correlacionadas con las demás y, a priori, tendrían mayor efecto para la recuperación de documentos.

Importancia de las variables (*RFFI*): Aquí se aplica un método de reducción de dimensiones usado en AA. Se contruye un modelo de clasificación basado en Random Forest (RF) utilizando los juicios de relevancia de la colección como clase (*relevante* o *no relevante*). Durante el entrenamiento, RF calcula un valor de importancia de la característica midiendo cuánto mejora la pureza de las hojas de cada árbol del bosque (impureza de la disminución media).

Permutaciones de la Importancia (*RFPI*): Algunos autores estudiaron los posibles sesgos al utilizar la impureza de la disminución media para establecer la importancia de las características (Altmann et al., 2010). Una alternativa propuesta normaliza las medidas de importancia de las características basándose en una prueba de permutación. En lugar de analizar la impureza, considera la caída en la precisión de la prueba cuando sus valores se permutan aleatoriamente.

3. Resultados

Luego de aplicados los métodos de ranking de dimensiones se ejecuta una tarea de recuperación sobre documentos. Para ello, se utiliza la colección MS-MARCO[4] ($\approx 8M$ de pasajes y $\approx 7K$ de consultas). Los vectores de embedding

[4] https://microsoft.github.io/msmarco/

se calculan utilizando 3 modelos: a) Topic Aware Sampling Balanced (TAS-B), modelo basado en un esquema de supervisión *teacher-student* (Hofstätter et al., 2021) b) ANCE, técnica basada en *contrastive learning* (Xiong et al., 2021) y c) BGE-B, familia de modelos multilinguales basados en el concepto de *flagship embedding* (Chen et al., 2023). Luego, se generan vectores densos ($d = 768$), se calcula la importancia (por los métodos propuestos) y se indexan usando FAISS[5] tomando las $n\,\%$ primeras dimensiones, con incrementos del $10\,\%$. Se ejecuta la recuperación y se calcula Mean Reciprocal Rank en el top-10 (MRR@10).

En la Figura 1 se observa que, en general, los métodos utilizados permiten reducir las dimensiones de los vectores de *embeddings* hasta un $50\,\%$ manteniendo una efectividad de $\approx 90\,\%$ para los modelos TAS-B y BGE-B. El método RFFI no alcanza tal performance para TAS-B. El comportamiento usando el modelo ANCE es completamente diferente. Para lograr un rendimiento competitivo en la métrica analizada es necesario conservar el $90\,\%$ (o el total) de las dimensiones. Esto sugiere que ANCE distribuye la importancia de las dimensiones de forma más homogénea. Este comportamiento es consistente con los resultados de otros estudios (Faggioli et al., 2024) con métodos dependiente de las consultas.

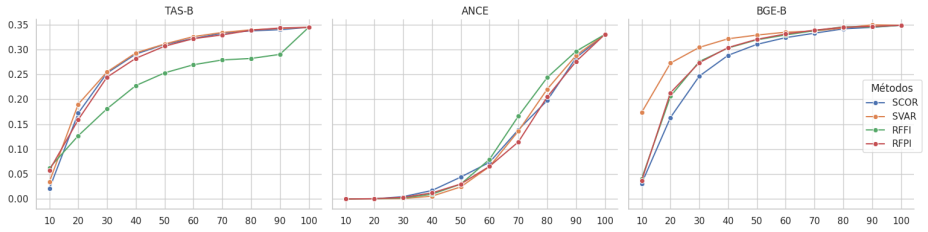

Figura 1. Rendimiento de métodos y modelos en la tarea de recuperación. Eje-x: porcentaje de dimensiones mantenidas en el vector de *embeddings*. Eje-y: valor de MRR@10.

Dim. Mantenidas (%)	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
Eficacia	0.50	0.78	0.87	0.92	0.94	0.96	0.97	0.98	1.00	1.00
Mejora Eficiencia	0.54	0.47	0.41	0.36	0.29	0.23	0.18	0.12	0.06	0.00

Tabla 1. Relación eficacia/eficiencia de acuerdo al porcentaje de dimensiones mantenidas usando BGE-B y SVAR.

La Tabla 1 muestra la mejora en el tiempo de búsqueda y la eficacia por cada punto de poda de dimensiones. Siguiendo con el ejemplo anterior, con el $50\,\%$ de las dimensiones se pueden reducir los tiempos de consulta en $\approx 30\,\%$. Como se puede apreciar, usar muy pocas dimensiones (ej. $10\,\%$) no incrementa la eficiencia en proporciones equivalentes ya que hay un costo fijo de inicialización de las estructuras de datos previo al proceso de búsqueda, lo que se traslada al resto de las mediciones. Estos resultados pueden ser de interés en implementaciones en producción de sistemas de búsqueda basados en vectores densos.

[5] https://github.com/facebookresearch/faiss/

4. Conclusiones y Trabajos Futuros

En este trabajo se proponen y evaluan métodos para rankear dimensiones de vectores densos para RI. Una primera observación es que hay que analizar el comportamiento del modelo en cuanto a cómo distribuye la importancia de las dimensones (ej. ANCE vs TAS-B). Si bien no hay un método que claramente supere al resto, para 2 de los modelos estudiados es posible alcanzar una performance de $\approx 90\,\%$ usando la mitad de las dimensiones (con una mejora en tiempo del $\approx\ 30\,\%$). Esto tiene impacto directo en la performance de la recuperación. También es interesante observar que los métodos más simples (SVAR) igualan o mejoran a los basados en un modelo de clasificación. Finalmente, se están evaluando métodos más complejos basados en grafos y nuevos modelos (ej. ModernBERT), aún no considerados en trabajos previos.

Referencias

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance:a corrected feature importance measure. *Bioinformatic*, *26*.

Bengio, Y., Courville, A., & Vincent, P. (2014). Representation Learning: A Review and New Perspectives.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, *3*.

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., & Liu, Z. (2023). BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation.

Faggioli, G., Ferro, N., Perego, R., & Tonellotto, N. (2024). Dimension Importance Estimation for Dense Information Retrieval. *SIGIR'24*.

Hofstätter, S., Lin, S.-C., Yang, J.-H., Lin, J., & Hanbury, A. (2021). Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. *SIGIR'21*.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense Passage Retrieval for Open-Domain Question Answering. En B. Webber, T. Cohn, Y. He & Y. Liu (Eds.), *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature Selection: A Data Perspective. *ACM Comp. Surv.*, *50*.

Obaid, H. S., Dheyab, S. A., & Sabry, S. S. (2019). The Impact of Data Pre-Processing Techniques and Dimensionality Reduction on the Accuracy of Machine Learning. *Information Technology, Electromechanical Engineering and Microelectronics Conference*.

Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P. N., Ahmed, J., & Overwijk, A. (2021). Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *International Conference on Learning Representations*.