Information retrieval using fuzzy fingerprints

Gonçalo Raposo[1,2][0000−0001−7806−6526], João Paulo Carvalho[2][0000−0003−0005−8299], Luísa Coheur[2][0000−0002−2456−5028], and Bruno Martins[2][0000−0002−3856−2936]

[1] Unbabel goncalo.raposo@unbabel.com
[2] INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
joao.carvalho@inesc-id.pt
{luisa.coheur,bruno.g.martins}@tecnico.ulisboa.pt

Abstract. Fuzzy fingerprints, derived from language model embeddings, have shown promise in classification tasks. This paper extends their application to information retrieval, using the well-established MS MARCO dataset. We assess the performance of these fingerprints against dense retrieval methods, particularly focusing on the use of both general and retrieval-optimized encoders, and decreasing the vector sizes. Our findings indicate that while fuzzy fingerprints may slightly underperform compared to dense retrieval, their performance remains comparable, especially with smaller vector sizes. This suggests their potential as a memory efficient retrieval method, while also showcasing the significant data representation capabilities inherent in the positions of embeddings.

Keywords: Fuzzy fingerprints · Information retrieval · Embeddings · Transformer-based models

1 Introduction

Information retrieval (IR) is a pivotal aspect of our digital age, enabling users to locate and access vast amounts of information efficiently. At its core, IR involves the searching, retrieval, and ranking of information [8]. The advent of pretrained language models has revolutionized IR by providing advanced embeddings that capture the nuanced semantics of text [16]. The embeddings, tailored to optimize similarity functions between relevant text pairs, often result in a vector space whose interpretability is not immediately apparent. This inherent complexity can obscure the semantic relationships encoded within the embeddings, making it challenging to intuitively understand or interpret the underlying textual similarities. Among other approaches are fuzzy fingerprints [9] – compact vector representations derived from textual features, which have been shown to offer a unique approach to text classification [17]. Furthermore, fuzzy fingerprints enhance interpretability through their design, which facilitates the examination of the most significant positions within the embeddings, offering insights into what the model deems most relevant [14].

This paper is the first to explore the use of fuzzy fingerprints as a tool for retrieval tasks. In particular, we compute them based on the embeddings from two sources: RoBERTa [11], known for its robust performance in natural language processing, and sentence-transformers [16,20], specifically fine-tuned for IR and similarity search. We explore various methods of generating fuzzy fingerprints from these models and conduct a comprehensive evaluation on the MS MARCO dataset [13]. Our research presents the first in-depth analysis of fuzzy fingerprints in the context of IR, highlighting their potential for enhancing the interpretability of pretrained language model embeddings and offering a more efficient approach to store the documents' vector representation.

2 Background and Related Work

The evolution of information retrieval methodologies can be traced from traditional sparse retrieval methods [18], which emphasize term frequency and inverse document frequency, to modern dense retrieval techniques [10]. Sparse retrieval, although effective in certain contexts, often struggles with semantic intricacies. Dense retrieval, powered by neural embeddings from pretrained language models like BERT [6] and GPT [2], addresses this by capturing deeper semantic relationships, though at the cost of interpretability.

The interpretability of these embeddings remains a significant challenge. While dense embeddings encapsulate rich semantic information, they often exist in high-dimensional, abstract spaces that are difficult to decipher. This complexity makes it challenging to intuitively understand the semantic similarities and relationships encoded within the embeddings [1].

Fuzzy fingerprints [9], initially conceptualized for identification and classification tasks [19,17], are constructed based on feature frequency, such as word frequency in texts. For a given class, a composite fingerprint is derived from the aggregate of individual document fingerprints within that class. During classification, the class is determined by comparing fingerprint similarity. Given the resemblance of large-scale classification to information retrieval, where numerous classes are akin to a vast array of documents, the application of fuzzy fingerprints appears promising for information retrieval tasks. This approach suggests a natural extension of its utility beyond classification, adapting to the complexities of retrieving relevant information from extensive datasets.

To access the performance of information retrieval systems, the MS MARCO dataset [13] has emerged as a key resource [4]. It provides a comprehensive collection of real-world queries and a solid framework for assessing both sparse and dense retrieval methods, making it a pivotal tool in benchmarking the efficacy of diverse retrieval strategies. Consequently, it is an excellent choice for our analysis.

In summary, the field of information retrieval is at a pivotal juncture with the integration of dense embeddings from pretrained language models and the innovative application of fuzzy fingerprints. This integration promises to enhance both the effectiveness and interpretability of retrieval systems.

3 Methodology

This study investigates the application of fuzzy fingerprints in information retrieval, leveraging pretrained language models to generate the embeddings that form the basis for their computation. Dense retrieval methodologies involve the direct comparison of embeddings through the use of lightweight similarity functions. However, in the context of fuzzy fingerprints, we will elaborate on the algorithm employed for their computation and describe the methodology for assessing similarity between two such fingerprints. Fuzzy fingerprints are distinguished by their flexibility in vector size, enabling an interesting comparison with dense retrieval approaches utilizing vectors of reduced dimensionality.

3.1 Pretrained Language Models

Pretrained language models have revolutionized natural language processing [6,15,2,3]. These models, trained on vast textual datasets, are adept at extracting and understanding semantic features from text. A key characteristic of these models is their ability to encode textual information into meaningful representations. Focusing on encoder models, they transform raw text into dense vectors that encapsulate the underlying semantic information. This transformation allows for a nuanced understanding and processing of language, far beyond simple keyword matching [24].

3.2 Dense Retrieval using Embeddings

In the realm of information retrieval, embeddings are a critical component [12]. These vectors represent textual content, encapsulating its semantic core. Various methods exist for generating embeddings, such as averaging the final layer outputs of a language model or using the CLS token from models like BERT [16]. In this setup, we employ a bi-encoder architecture, which processes texts independently to produce embeddings. This contrasts with cross-encoders that evaluate concatenated text pairs.

The essence of dense retrieval is calculating a similarity function between these embeddings, using metrics like cosine similarity, dot product, or Euclidean distance. The bi-encoder model efficiently scales for large datasets, making it particularly relevant for tasks requiring extensive document comparisons. The choice of similarity metric is tailored to the task's specific needs and the characteristics of the embeddings [16].

Training the encoder model to optimize similarity scores between similar texts is crucial. This training process involves adjusting the model so that the embeddings of texts with similar meanings are close to each other in the vector space, according to the chosen similarity metric. This fine-tuning enhances the model's ability to accurately retrieve information based on semantic content, rather than relying on superficial text matching [23].

3.3 Fuzzy Fingerprints and Similarity

Fuzzy fingerprints derive from embeddings generated by pretrained language models, capturing the semantic essence of text. Initially conceived for classification tasks, fuzzy fingerprints were originally applied to aggregate class information [9]. Contrasting with their initial application, our adaptation for information retrieval purposes shifts the focus from broad class aggregates to the granularity of individual documents. This change introduces a novel variant within the domain of fuzzy fingerprints, thus. naming it ”fuzzy retrieval fingerprints” would be more appropriate. This adaptation tailors the methodology to meet the demands of information retrieval by leveraging document-specific information.

Initially, the algorithm takes the absolute values of the embeddings vector $\mathbf{v}$ to ensure all values are non-negative. The essence of this approach lies in selecting the top-k elements of this transformed vector, which are deemed to be the most critical features represented by the embeddings, since values close to 0 would not have a significant effect in the output. These elements are indexed, and each is assigned a unique value calculated using a membership function, which takes into account its position and the total size of the fingerprint, k. This step ensures that the fingerprint captures the most prominent aspects of the text's semantic space, thus creating a fuzzy yet precise representation of its content. The final output is a set of key-value pairs, where each key is an index of a significant feature, and the value is its assigned membership value, forming the Fuzzy Fingerprint $\boldsymbol{\phi}$ of size k. In Algorithm 1 we detail their calculation.

Algorithm 1 Computation of Fuzzy Fingerprint from Embeddings

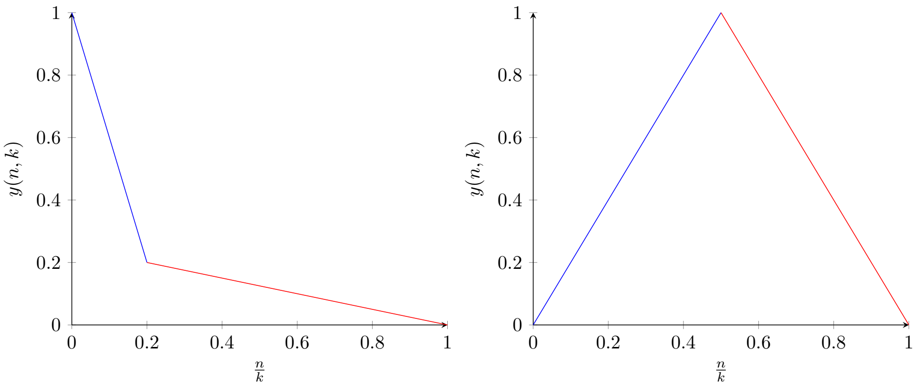
```
Require: Embeddings vector $\mathbf{v}$, fingerprint size $k$
Ensure: Fuzzy fingerprint $\boldsymbol{\phi}$ of size $k$
$\mathbf{v}' \leftarrow abs(\mathbf{v})$
Compute the top-$k$ values of $\mathbf{v}'$ to get indices $I$
$\boldsymbol{\phi} \leftarrow \{\}$
$n \leftarrow 0$
for each index $i$ in $I$ in decreasing order of value do
    Append the pair $(i, \mu(n, k))$ to $\boldsymbol{\phi}$
    $n \leftarrow n + 1$
end for
return $\boldsymbol{\phi}$
```

As for the membership function required in the fuzzy fingerprints calculation, we tried two variations, which we referred to as: decreasing and triangular. These are described in Equations 1 and 2 and illustrated in Figure 1. Note that if the size of the fingerprint is $k = 1$, then the membership value is always $\mu(n, 1) = 1$. The decreasing function was designed to prioritize the most significant positions within the embeddings, enhancing their influence in the fingerprint. This takes inspiration from the Pareto principle, giving more importance to initial positions.

Conversely, the triangular function adopts a more balanced approach, emphasizing moderately significant positions while diminishing the impact of both the highest and lowest extremes. The reason of this triangular function was that, for some language models, we observed a higher variability in the middle positions of the sorted embedding.

$$\mu(n,k) = \begin{cases} 1 - \frac{(1-a)}{a} \cdot \frac{n}{k} & \text{, if } 0 \leq \frac{n}{k} < a \\ \frac{a}{1-a} \cdot \left(1 - \frac{n}{k}\right) & \text{, if } a \leq \frac{n}{k} \leq 1 \end{cases} \tag{1}$$

$$\mu(n,k) = \begin{cases} \frac{1}{a} \cdot \frac{n}{k} & \text{, if } 0 \leq \frac{n}{k} < a \\ \frac{1}{1-a} \cdot \left(1 - \frac{n}{k}\right) & \text{, if } a \leq \frac{n}{k} \leq 1 \end{cases} \tag{2}$$



(a) Decreasing ($a = 0.2$). (b) Triangular ($a = 0.5$).

Fig. 1: Membership functions used when calculating fuzzy fingerprints.

Finally, given two fingerprints ϕ_A and ϕ_B, we can compute their similarity as formulated in 3, which basically consists in aggregating and normalizing the intersection of 2 discrete fuzzy sets. This function employs the concept of the fuzzy AND operation. The Gödel t-norm, or minimum operation, is chosen for this purpose, aligning with established practices in fuzzy logic. However, other t-norms are also viable alternatives.

$$sim(\phi_A, \phi_B) = \frac{1}{\sum_{j=0}^{k-1} \mu(j,k)} \sum_{i \in I_A \cap I_B} \min(m_{iA}, m_{iB}) \tag{3}$$

where I_A and I_B are the sets of indices of the top-k values of embeddings A and B, and $m_{i,a}$ and $m_{i,b}$ are the membership values associated with the rank indices that are present in both fingerprints. The first term is a normalizing factor that corresponds to the sum of all membership values, such that the similarity score is bounded between 0 and 1.

Example 1. Suppose we have two vector embeddings obtained from two texts A and B. We will compute the corresponding fuzzy fingerprints of size $k = 3$ using Algorithm 1 with the decreasing membership function described in Equation 1 with $a = 0.2$. Then, we will compute the similarity between the two fingerprints using Equation 3.

$$\mathbf{v_A} = [0.7, -0.5, 0.2, -0.8, -0.1] \quad \text{and} \quad \mathbf{v_B} = [0.0, -0.2, 0.1, -0.9, 0.1]$$

Following Algorithm 1 (and coloring common indices to facilitate):

1. $\mathbf{v'_A} = [0.7, 0.5, 0.2, 0.8, 0.1] \quad \text{and} \quad \mathbf{v'_B} = [0.0, 0.2, 0.1, 0.9, 0.1]$
2. $\text{top-}3_A = \{0.8, 0.7, 0.5\} \quad \text{and} \quad \text{top-}3_B = \{0.9, 0.2, 0.1\}$
3. $I_A = \{3, 0, 1\} \quad \text{and} \quad I_B = \{3, 1, 2\}$
4. $\phi_A = \{(3, \mu(0,3)), (0, \mu(1,3)), (1, \mu(2,3))\}$
   $\phi_B = \{(3, \mu(0,3)), (1, \mu(1,3), (2, \mu(2,3))\}$
5. $\phi_A = \{(3, 1.0000), (0, 0.1667), (1, 0.0833)\}$
   $\phi_B = \{(3, 1.0000), (1, 0.1667), (2, 0.0833)\}$

And then using Equation 3 to calculate the similarity:

$$\begin{aligned} sim(\phi_A, \phi_B) &= \frac{1}{\mu(0,3) + \mu(1,3) + \mu(2,3)} \sum_{(i,n):i\in\{3,0,1\}\cap\{3,1,2\}} \min\{\mu_A(n,3), \mu_B(n,3)\} \\ &\approx \frac{1}{1.0000 + 0.1667 + 0.0833} (\min\{\mu(0,3), \mu(0,3)\} + \min\{\mu(2,3), \mu(1,3)\}) \\ &\approx 0.8667 \end{aligned}$$

3.4 Reducing vector size

We previously describe two distinct approaches for information retrieval: computing a similarity score directly from embeddings or using fuzzy fingerprints. Each approach has its own advantages and trade-offs, making them suitable for different scenarios based on the specific requirements of the task at hand. One important factor is the efficiency of each approach, which can be directly influenced by the size of the involved vectors. We aim to assess the performance implications of employing reduced vector sizes in retrieval tasks—innately facilitated by the design of fuzzy fingerprints and achieved for dense retrieval through post-hoc dimensionality reduction using Principal Component Analysis (PCA).

Dense Retrieval In dense retrieval, reducing the size of embedding vectors is a crucial task, and several strategies exist for this purpose [22]. A possibly highly effective, though computationally expensive, method involves introducing a linear layer after the encoder to project the embedding into a smaller dimension. This layer reduces the dimensionality of the vectors, and the entire system is then trained to optimize the similarity function between these compressed embeddings. While effective, this approach is computationally expensive due to the additional extensive training required.

Another, more expedient approach is to use Principal Component Analysis (PCA) [7]. PCA reduction involves computing a transformation matrix using the embeddings of a large set of documents and then applying this transformation to both the query and document embeddings. This method, while faster, may not always preserve the fine-grained semantic relationships as effectively as the linear layer approach. It also requires computing the embeddings of all the documents in order to obtain the transformation.

Fuzzy Fingerprints For fuzzy fingerprints, the process of reducing the size of the vector is more straightforward. The reduction is achieved by simply adjusting the parameter k in the top-k selection process of the embedding. This approach directly controls the granularity of the fingerprint, balancing between detail and computational efficiency. The smaller k is, the less detailed but more computationally efficient the fingerprint becomes. This method provides a simple yet flexible way to manage the trade-off between the fidelity of the semantic representation and the computational resources required. Moreover, it is something that can be decided on-the-fly, that is, without the need of processing the entire set of documents.

4 Experimental Setup

Our experiments will focus on evaluating the performance of fuzzy fingerprints in relevant passage retrieval. We will try embeddings from different models, different membership functions, and varying the size of the fingerprints. Additionally, we will analyze how the performance compares with dense retrieval.

4.1 Dataset

The experimental framework for evaluating dense retrieval and fuzzy fingerprints is designed around the MS MARCO dataset [13], accessed through the BeIR benchmark [21] – Table 1. This is a large-scale dataset that gathers questions or short keyword-based queries, collected from real Bing searches, and focus on relevant passage retrieval. Additionally, it contains an official training set of query-passage pairs for fine-tuning.

Table 1: Technical details about the evaluated dataset.

Dataset	# passages	# queries	# train pairs
MS MARCO	8,841,823	6,980	532,761

Besides queries and documents, this dataset also contains qrels, which are records of what documents are relevant to each query. Listing 1.1 illustrates the structure of this dataset.

Listing 1.1: Mock example of the MS MARCO dataset.

```
corpus = {
    "doc1" : {
        "title": "Albert Einstein",
        "text": "Albert Einstein was a German-born theoretical physicist
             (...)"
        },
    "doc2" : {
        "title": "",
        "text": "Wheat beer is a top-fermented beer which is (...)"
    },
}

queries = {
    "q1" : "Who developed the mass-energy equivalence formula?",
    "q2" : "Which beer is brewed with a large proportion of wheat?"
}

qrels = {
    "q1" : {"doc1": 1},
    "q2" : {"doc2": 1},
}
```

4.2 Encoder Models

As an off-the-shelf encoder model, we try RoBERTa [11], a Transformer model pretrained with the masked language modeling (MLM) objective in English text. As this model is not specifically optimized for retrieval tasks, its performance serves as a baseline, reflecting general encoder capabilities without retrieval-specific tuning.

As for the model optimized retrieval, we used one of the top models from Sentence Transformers [16]. This model started from the pretrained model MPNet [20] and was fine-tuned with a contrastive learning objective: a sentence pair is mixed with other randomly sample sentences and the model should match the original pair. This model is expected to demonstrate superior performance in retrieval tasks, thanks to its fine-tuning. Table 2 summarizes some details about these models.

Table 2: Technical details about the evaluated encoder models.

Model	# parameters	embedding size	pooling	retrieval fine-tuning
roberta-base	125M	768	CLS	✗
multi-qa-mpnet-base-dot-v1	110M	768	CLS	✓

4.3 Embeddings and Fuzzy Fingerprints Specifications

Both models generate embeddings based on the CLS token vector representation. This choice is driven by the CLS token's design to capture the essence of the entire input sequence [6].

For evaluating dense retrieval, we consider two similarity measures to compare embeddings: cosine similarity and dot product. For the model specifically fine-tuned for retrieval, the authors recommend the dot product as a suitable scoring function.

As for the fuzzy fingerprints, which are derived from the CLS token, we consider both the decreasing and the triangular membership functions. Additionally, we tested different variation by changing the a parameter described in Equations 1 and 2.

4.4 Performance Metrics

To evaluate the retrieval performance in our experiments, the main metric was mean average precision (mAP) [25]. This metric offers a comprehensive view of retrieval effectiveness across different query-document pairs and can be calculated as:

$$mAP = \frac{1}{N_q} \sum_{q \in \text{queries}} \frac{1}{N_{rd}} \sum_{n=1}^{N_d} P(n) r(n) \tag{4}$$

where N_q is the total number of queries, N_{rd} is the number of relevant documents for query q, N_d is the total number of documents, $P(n)$ is the precision at n, and $r(n)$ is the relevance of the n^{th} retrieved document (1 if relevant and 0 if not relevant).

Additionally, the Spearman correlation coefficient [5] is calculated between the scores derived from the sentence-transformers model and the other methods. This approach assumes as reference the scores obtained with the sentence-transformers model using dot product, therefore, the Spearman correlation with itself will be 1 and all the others lower than 1.

5 Results and Discussion

5.1 Comparative analysis of dense and fuzzy fingerprint retrieval

Our first experiments used the embeddings of RoBERTa and compared the performance of using cosine similarity or dot product operations for dense retrieval against using fuzzy fingerprints. Additionally, for the fuzzy fingerprints, we experimented with different membership functions, which we present in Table 3.

Since RoBERTa was not specifically fine-tuned for retrieval, its embeddings are not particularly good for information retrieval. Although this encoder model exhibits good semantic representations in other applications, since the queries

Table 3: Evaluation of RoBERTa model for retrieval on the MS MARCO dataset. Embedding and fingerprint of size $k = 768$.

Model	Method	Membership function	a	mAP	Correlation with DP
roberta-base	cosine similarity	-	-	0,0124	0,1198
	dot product	-	-	0,0054	0,05484
	fuzzy fingerprint	decreasing	0,9	0,0101	0,1146
	fuzzy fingerprint	decreasing	0,5	**0,0164**	0,1746
	fuzzy fingerprint	decreasing	0,2	0,0147	0,1568
	fuzzy fingerprint	decreasing	0,1	0,0119	0,1314
	fuzzy fingerprint	triangular	0,5	0,0119	0,0899
	fuzzy fingerprint	triangular	0,2	0,0151	0,1561
	fuzzy fingerprint	triangular	0,1	0,0160	**0,1781**

and passages of MS MARCO are very different (see Listing 1.1), the embeddings are unavoidably too different. Although the scores were very low for both models, the fuzzy fingerprints showed slightly higher values, which may indicate a promising potential when no specific training of the encoder model occurred. By replacing the encoder with one that was optimized for retrieval, the retrieval performance was readily improved, as is shown in Table 4. Since this model was optimized with dot product scoring, that was the function we selected for dense retrieval. Note, once again, that the Spearman correlation of the first entry is 1 because it is also the reference.

Table 4: Evaluation of Sentence Transformers model for retrieval on the MS MARCO dataset. Embedding and fingerprint of size $k = 768$.

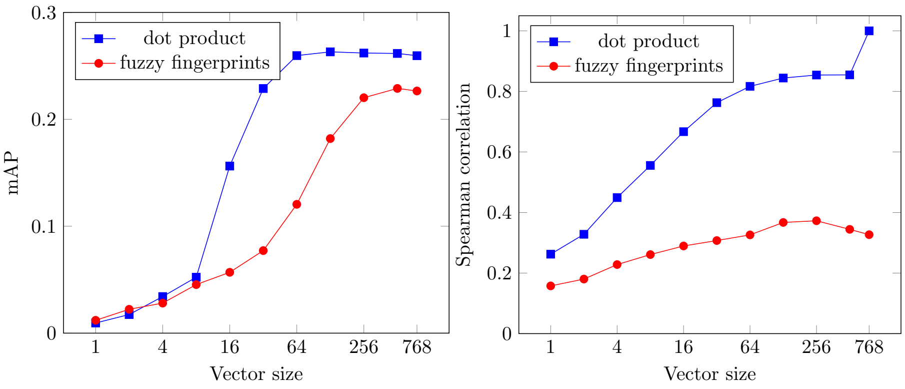
Model	Method	Membership function	a	mAP	Correlation with DP
multi-qa-mpnet- -base-dot-v1	dot product	-	-	0,2595	1
	fuzzy fingerprint	decreasing	0,9	0,0734	0,1304
	fuzzy fingerprint	decreasing	0,5	0,2231	0,2619
	fuzzy fingerprint	decreasing	0,2	**0,2374**	0,3123
	fuzzy fingerprint	decreasing	0,1	0,2265	**0,3268**
	fuzzy fingerprint	triangular	0,5	0,0534	0,0931
	fuzzy fingerprint	triangular	0,2	0,0539	0,0856
	fuzzy fingerprint	triangular	0,1	0,1185	0,1358

The information retrieval performance of fuzzy fingerprints changed drastically with the membership function used. Nonetheless, for the decreasing membership function with $a \sim 0.2$ the mAP scores were comparable to those of dense retrieval performed with dot product between embeddings. These results indicate that the values of the embedding vectors are not as relevant and meaningful for information retrieval tasks. From our results, the most excited positions of the embedding already contain enough information for the similarity search.

Despite the overhead in calculating the fingerprints, these actually obtained a good retrieval performance. This approach could be utilized for a more memory efficient alternative to embeddings. Since they only consider the positions and not the values, the stored vectors could be represented by smaller integer values: we only need 756 positions instead of 756 float values.

5.2 Reducing vector size

To evaluate how efficient the embeddings and fuzzy fingerprints are in retaining information about the semantics of each document and query, we did an experiment where we decreased the size of the embedding vectors used for dense retrieval (using PCA reduction) and of the fuzzy fingerprints, which we report in Figure 2.



(a) Mean average precision scores.

(b) Spearman correlation with dense retrieval.

Fig. 2: Evaluation on how the vector and fingerprint sizes affect the retrieval performance. The fuzzy fingerprints used the decreasing membership function and $a = 0.2$.

As expected, the performance of both approaches decreased as the vector size decreased. Nonetheless, the embedding vectors retained more information for smaller vector sizes, which might be attributed to the efficiency of the PCA reduction in keeping the separation between points in the embedding space. Interestingly, for vector sizes equal or smaller than 8, the performance of dense retrieval using the dot product and fuzzy fingerprints, measured with the mAP score, is practically equivalent.

6 Conclusion and Future Work

In this work, we explored applying fuzzy fingerprints in the domain of information retrieval. Our methodology involved a comparative analysis against traditional dense retrieval methods, specifically focusing on the dot product between embeddings. The MS MARCO dataset, a renowned benchmark in this field, served as our evaluation platform. We utilized two distinct encoder models for this purpose: RoBERTa, and a model from the Sentence Transformers that was specifically fine-tuned for retrieval.

Our findings reveal that when employing the general encoder model RoBERTa, fuzzy fingerprints exhibited a slightly superior performance compared to conventional dense retrieval. This outcome hints at the untapped potential of fuzzy fingerprints, particularly in scenarios where general encoder models are in play. In contrast, for the model fine-tuned for retrieval, the expected better performance of dense retrieval was observed. However, it is noteworthy that fuzzy fingerprints still managed to deliver comparable results, despite encapsulating significantly less information. This aspect underscores the efficiency of fuzzy fingerprints, positioning them as a viable, memory-efficient alternative to full embeddings in certain contexts. A critical observation from our study is the impact of reducing the size of fingerprints and embeddings. As anticipated, a decrease in size correlated with a diminished performance. However, a point of convergence in the retrieval performance was noted for sizes equal to or less than 8. It is also noteworthy that the size of fuzzy fingerprints can be changed on demand without any additional computation, while to reduce the size of the embeddings used for dense retrieval it was necessary to process the entire set of documents for the PCA reduction.

As future work, it would be relevant to investigate the application of fuzzy fingerprints across diverse information retrieval datasets, spanning various domains. This would provide a broader understanding of their adaptability and effectiveness. Moreover, it would be very interesting to optimize the embeddings specifically for fingerprint similarity. Given the non-differentiable nature of fuzzy fingerprints, one potential pathway could involve the utilization of genetic algorithms. This way, fuzzy fingerprints could be compared against dense retrieval using embeddings in a fairer setting.

Acknowledgments

This research was supported by the Portuguese Recovery and Resilience Plan through the project C645008882-00000055 (Center for Responsible AI), and FCT, *Fundação para a Ciência e a Tecnologia*, under the project UIDB/50021/2020 (DOI:10.54499/UIDB/50021/2020)

References

1. Allen, C., Hospedales, T.: Analogies explained: Towards understanding word embeddings. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 223–231. PMLR (09–15 Jun 2019), https://proceedings.mlr.press/v97/allen19a.html
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
3. Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y.T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M.T., Zhang, Y.: Sparks of artificial general intelligence: Early experiments with gpt-4 (Mar 2023). https://doi.org/10.48550/ARXIV.2303.12712
4. Dalton, J., Xiong, C., Callan, J.: Cast 2020: The conversational assistance track overview. In: The Twenty-Ninth Text REtrieval Conference (TREC 2020) Proceedings (2020), https://trec.nist.gov/pubs/trec29/trec2020.html
5. Daniel, W.W.: Applied nonparametric statistics. Duxbury classic series, Duxbury, Pacific Grove, CA [u.a.], 2. ed. edn. (1990)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
7. Greenacre, M., Groenen, P.J.F., Hastie, T., D'Enza, A.I., Markos, A., Tuzhilina, E.: Principal component analysis. Nature Reviews Methods Primers **2**(1) (Dec 2022). https://doi.org/10.1038/s43586-022-00184-w
8. Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., Wu, C., Croft, W.B., Cheng, X.: A deep look into neural ranking models for information retrieval. Information Processing & Management **57**(6), 102067 (Nov 2020). https://doi.org/10.1016/j.ipm.2019.102067
9. Homem, N., Carvalho, J.P.: Authorship identification and author fuzzy "fingerprints". In: 2011 Annual Meeting of the North American Fuzzy Information Processing Society. IEEE (Mar 2011). https://doi.org/10.1109/nafips.2011.5751998
10. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., tau Yih, W.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics (Apr 2020). https://doi.org/10.18653/v1/2020.emnlp-main.550
11. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (Jul 2019). https://doi.org/10.48550/ARXIV.1907.11692

12. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval. Transactions of the Association for Computational Linguistics **9**, 329–345 (2021). https://doi.org/10.1162/tacl_a_00369, https://aclanthology.org/2021.tacl-1.20
13. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A human generated machine reading comprehension dataset. In: Besold, T.R., Bordes, A., d'Avila Garcez, A.S., Wayne, G. (eds.) Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016. CEUR Workshop Proceedings, vol. 1773. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf
14. Pereira, P., Ribeiro, R., Moniz, H., Coheur, L., Carvalho, J.P.: Fuzzy fingerprinting transformer language-models for emotion recognition in conversations. In: 2023 IEEE International Conference on Fuzzy Systems (FUZZ). pp. 1–6 (2023). https://doi.org/10.1109/FUZZ52849.2023.10309719
15. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020), http://jmlr.org/papers/v21/20-074.html
16. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), https://arxiv.org/abs/1908.10084
17. Ribeiro, R., Pereira, P., Coheur, L., Moniz, H., Carvalho, J.P.: Fuzzy Fingerprinting Large Pre-trained Models, pp. 232–243. Springer Nature Switzerland (2023). https://doi.org/10.1007/978-3-031-39965-7_20
18. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Foundations and Trends® in Information Retrieval **3**(4), 333–389 (2009). https://doi.org/10.1561/1500000019
19. Rosa, H., Carvalho, J.P., Calado, P., Martins, B., Ribeiro, R., Coheur, L.: Using fuzzy fingerprints for cyberbullying detection in social networks. In: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE (Jul 2018). https://doi.org/10.1109/fuzz-ieee.2018.8491557
20. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: Mpnet: Masked and permuted pre-training for language understanding. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 16857–16867. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/c3a690be93aa602ee2dc0ccab5b7b67e-Paper.pdf
21. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021), https://openreview.net/forum?id=wCu6T5xFjeJ
22. Tonellotto, N., Macdonald, C.: Query embedding pruning for dense retrieval. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. CIKM '21, ACM (Oct 2021). https://doi.org/10.1145/3459637.3482162

23. Xu, C., Guo, D., Duan, N., McAuley, J.: LaPraDoR: Unsupervised pretrained dense retriever for zero-shot text retrieval. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Findings of the Association for Computational Linguistics: ACL 2022. pp. 3557–3569. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.findings-acl.281, https://aclanthology.org/2022.findings-acl.281
24. Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating text generation with bert. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=SkeHuCVFDr
25. Zhu, M.: Recall, precision and average precision. Department of Statistics and Actuarial Science, University of Waterloo, Waterloo **2**(30), 6 (2004)