FONDO TERRITORI LAVORO E CONOSCENZA CGIL, CISL, UIL

Deliverable D1.0

Territori Aperti: Research on state-of-the-art technologies for Information Retrieval

http://territoriaperti.univaq.it

Project Title	:	Territori Aperti

Deliverable Number	:	D1.0
Title of Deliverable	:	Territori Aperti: Research on state-of-the-art technologies for Information Retrieval
Nature of Deliverable	:	Report, Other
Dissemination level	:	Public
Licence	:	–
Version	:	1.0
Contractual Delivery Date	:	
Actual Delivery Date	:	
Contributing WP	:	
Editor(s)	:	Andrea D'Angelo (Università degli studi dell'Aquila)
Author(s)	:	Andrea D'Angelo (Università degli studi dell'Aquila)
Reviewer(s)	:	

Abstract

The deliverable describes the state of the ongoing research of state-of-the-art methods and models for Information Retrieval, Passage Retrieval, Natural Language Processing, and semantic embeddings with neural networks.

Keyword List

Information Retrieval, Passage Retrieval, BERT

Table Of Contents

List Of Figures ..... **VII**

1 Report on Recent Technologies for Information Retrieval **1**

1.1 Transformer Neural Networks *1*

1.2 Evaluating an Information Retrieval model *1*

1.3 Vector space model *2*

1.3.1 TF-IDF function *2*

1.3.2 Distance metrics *3*

1.4 Documents as clouds of vectors *3*

1.4.1 Local Outlier Factor *5*

1.4.2 DBSCAN *6*

Bibliography **9**

List Of Figures

Figure 1.1: Set of retrieved vs. relevant documents 2

Figure 1.2: Example of cosine similarity between query and two documents. 3

Figure 1.3: Example of a cloud of vectors representing a document. 4

Figure 1.4: Example of a cloud of vectors representing a document, in three dimensions. 7

Figure 1.5: Example of reach distances from o for k = 3 8

1 Report on Recent Technologies for Information Retrieval

In recent years, Transformers and BERT in particular have dominated the landscape of recent progresses in Information Retrieval. Just by looking at the recent European Conference on Information Retrieval (ECIR 2022) proceedings [1], it is clear that much of the current research is actively pursuing semantic IR with the use of transformers. Many systems are able to achieve state-of-the-art results by employing and fine-tuning Google's BERT. A good summary of these projects is presented in [2], that details how BERT has been used with great success in question answering.

1.1. Transformer Neural Networks

Transformer Neural Networks were first introduced in the Attention is All You Need paper [3], that argued how dropping the recurrent nature of the former state-of-the-art models to focus exclusively on Attention Functions produced better results while speeding up training considerably, due to the new-found opportunity for parallelization. In particular, an Attention Function as defined in the aforementioned paper maps a Query and a Set of Key,Value pairs to an output, where all of the involved elements are Vectors.

The computed output is a weighted sum of the values, where the assigned weight to each value is the result of a Compatibility function between the Query and the Key.

Only two years later, in 2019, Google published the famous Google BERT paper [4], presenting their new Language model capable of outstanding performances. By allowing the end user to fine-tune it, BERT is extremely customizable and flexible and has thus been used by several systems to obtain state-of-the-art results in many subfields of Passage Retrieval. BERT is still a transformer, so the way it works is similar to what was previously mentioned: given a collection of documents, BERT transforms each token or sentence into semantically embedded vectors. What this means is that the same token will be transformed into drastically different vectors if it appears in different contexts. This is crucial to understand the model that we are about to propose.

1.2. Evaluating an Information Retrieval model

At their core, each IR system's goal is the same: retrieving information that the user finds relevant according to their query, among a collection of documents. There are a plethora of metrics who aim to evaluate such a system. The most basic ones are **Precision and Recall**.

Defining True Positives (TP), False negatives (FN), True Negatives (TN) and False Positives (FP) as in figure 1.1, we can then identify precision and recall as:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

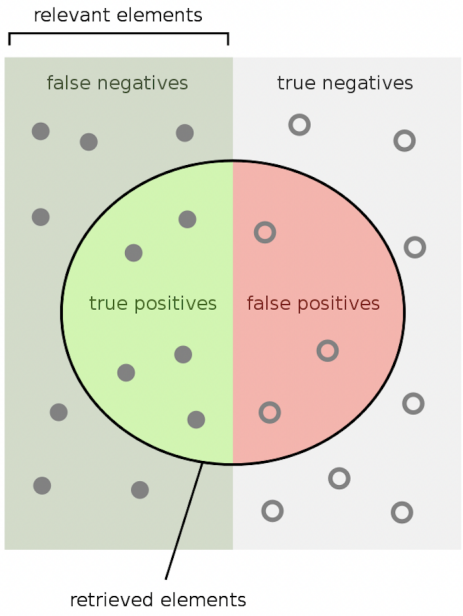

Figure 1.1: Set of retrieved vs. relevant documents

Another very important metric is Mean Average Precision mAP. mAP is defined, of course, as the mean of the average precision of all classes.

$$mAP = \frac{1}{n} * \sum_{k=1}^{n} AP_k$$

There are a bunch of tools created with the purpose of evaluating IR systems, for instance **trec_eval**. Trec eval asks for the collection's qrels and the results, given in a specific format. A collection's qrels is a dataset composed of a sample of specific queries embedded with the hand-picked relevant documents for each. They provide a stable ground-truth for relevant documents relating to a specific query and allows the system to be objectively evaluated.

1.3. Vector space model

In classical Information Retrieval, the vector space model is how documents are usually represented. Simply put, each document will be characterized by a single vector of n dimensions, where n is the size of the vocabulary in the collection.

Therefore, for document i, we will have a vector such as:

$d_i = (w_{1,i}, w_{2,i} \dots w_{n,i})$

where each $w_{j,i}$ corresponds to a separate term. In Boolean retrieval, we would have 1 if the term appears in the document and 0 if it does not. In more modern and sophisticated models, we would instead have the TF-IDF score of that term in the document.

1.3.1. TF-IDF function

The TF-IDF weighting schema is defined as:

$$TF - IDF(t,d) = TF(t,d) * IDF(t,D) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} * \log \frac{N}{|\{d \in D : t \in D\}|}$$

which is basically the product between the relative Term Frequency TF of that term t in document d and the inverse document frequency of the term t in the collection D.

The idea behind the TF-IDF function is to assign an high score to a term that appears frequently in the document (indicating that the document focuses on that concept) but not too many times, because in that case it might be a stop-word or a word that is not really meaningful. Its use is now widespread among classical Information Retrieval systems and has been the standard for a long time.

1.3.2. Distance metrics

In order to retrieve ranked documents, standard IR systems encode the query in the same way as the document. The query can thus be defined as

$q = (w_{1,q}, w_{2,q}, \ldots, w_{n,q})$

and this allows to consider multiple options for distance metrics. The most natural one would be to just compute the euclidean distance between each document and the query. However, the most popular solution by far is the one of **cosine similarity**. With cosine similarity, we compute the cosine of the angle between the vector representing the query and the ones representing the documents:

$$\cos(d_j, q) = \frac{\mathbf{d_j} \cdot \mathbf{q}}{\|\mathbf{d_j}\| \, \|\mathbf{q}\|} = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}}$$

Recall that no element of the document vectors or the query vector is negative, since they were defined using the TF-IDF weighting schema. Because of this, a cosine similarity of 0 implies that the document vector and the query vector are orthogonal, so they have no match whatsoever. The higher the cosine similarity, the higher the document should be ranked among the results.

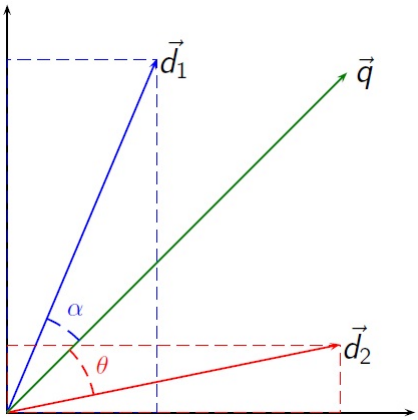

Figure 1.2: Example of cosine similarity between query and two documents.

A visual representation of the cosine similarity metric is shown in figure 1.2. Naturally, this figure either implies that the figure has been simplified or that there are only two terms in the collection, since they were plotted in two dimensions. It is not easy to visualize results of this kind when dealing with a realistic collection.

1.4. Documents as clouds of vectors

While Transformers are fairly common, representing documents as a set of vectors is far less popular. To the best of our knowledge, the only works that moved in this direction are an earlier version of this project [5] and a work involving the Word2Vec embedding [6]. In particular, Word2Vec is a simple two-layer Neural Network which aims, like BERT, to embed semantics into a vector representation of

the tokens. The score is then calculated by a probabilistic function, and distances itself considerably from our work, where we propose a density function to be used as distance metrics. We want to emphasize that, despite the entire Transformer technologies being fairly recent, the true novel approach is a byproduct of this density metric.

Representing a document as a set (from here on, ”cloud”) of vectors is drastically different to the classical IR approach of representing a document with a vector of n dimensions, where n is the number of terms in the collection. It allows us to avoid using sparse vectors and focus on few dimensions that are significant. Of course this kind of model also helps us with the visualization of concepts and distance metrics. While a single vector with thousands of dimensions might be hard or even meaningless to visualize, the same is not true with a Cloud of vectors representing tokens, which might give us a good idea on how the document is structured and whether or not it contains multiple topics, semantically distant from each other.

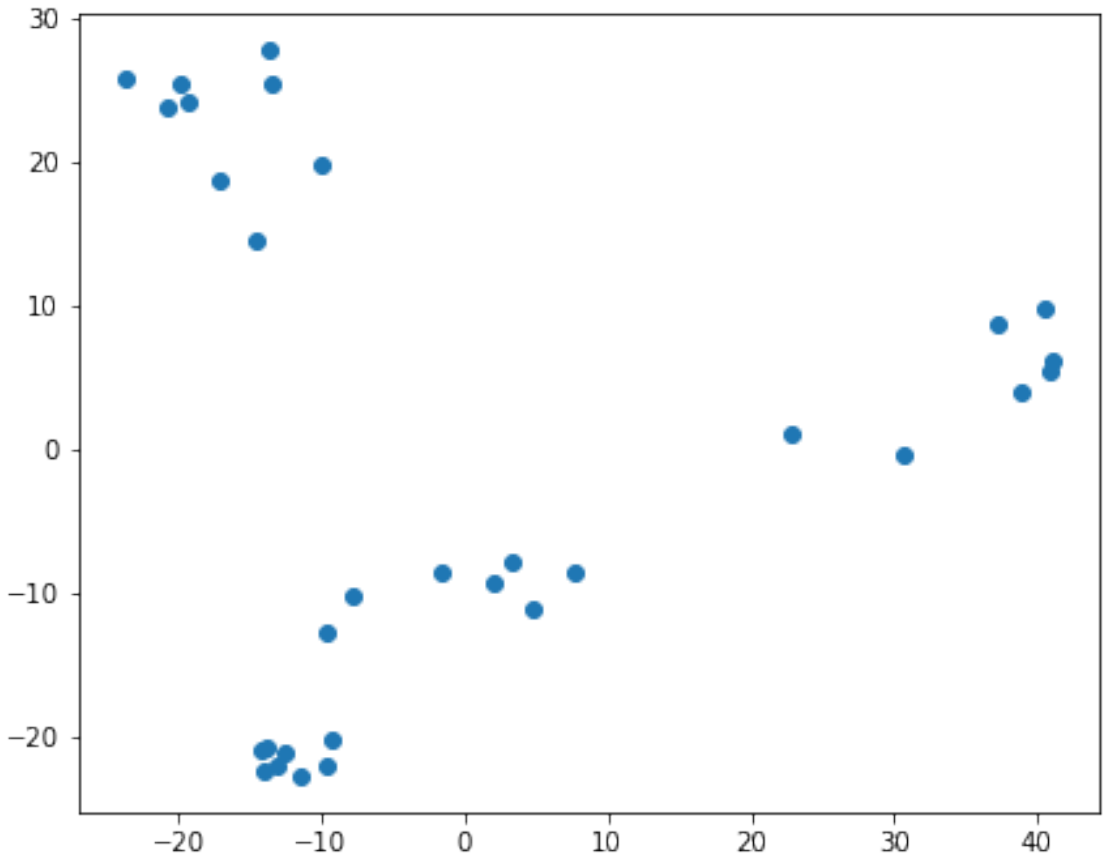

Figure 1.3: Example of a cloud of vectors representing a document.

Referring to figures 1.3 and 1.4, these images represent a document that was transformed by BERT and then visualized by applying PCA on its vectors. It is easy to see that this document contains three clusters of semantically related terms, and we can thus infer that it focuses on three main topics. Naturally, this is just a basic example, but it is helpful to understand the idea behind the model at a glance.

The aim of the model that we are currently developing, however, is not only to consider the implementation of such tools to rank documents. The true novel approach is to use a density-based metric to find which documents are the most relevant to the query. References for such metric were many outlier metrics, including Local Outlier Factor LOF [7].

Our goal was to score a specific vector given its closeness to clusters of other vectors, or in other words, based on how much the density around it is similar to the density of vectors around the others. This score would give us a numerical answer on how much of an outlier that vector is in respect to the others. We found that the Local Outlier Factor LOF, while not being exactly what we needed, was a good starting point.

1.4.1. Local Outlier Factor

In order to explain the implemented outlier metric, a deeper look into how Local Outlier Factor (LOF) works is in order. To do so, a bunch of definitions need to be explicated. All of them are taken from the LOF paper [7].

Definition 1 (k-distance)**.** For any positive integer k, the k distance of an object p, denoted as k-distance(p), is defined as the distance d(p,o) between p and an object o $\in$ D such that:

- for at least k objects o' $\in D \setminus \{p\}$ it holds that d(p,o') $\leq$ d(p,o)
- for at most k-1 objects o' $\in D \setminus \{p\}$ it holds that d(p,o') $<$ d(p,o)

Definition 2 (k-distance neighborhood)**.** Given k-distance(p), the k-distance neighborhood of p contains every object whose distance from p is not greater than the k-distance. In mathematical terms,

$$N_{k-distance(p)}(p) = \{q \in D \setminus \{p\} \mid d(p,q) \leq k - distance(p)\}$$

Definition 3 (Reachability distance)**.** Let k be a natural number. The reachability distance of object p is defined as

$$reach - dist_k(p,o) = \max\{k - distance(o), d(p,o)\}$$

Simply put, the reachability distance from p to o with respect to k is the distance between p and o, but at least the k-distance of o, as defined in definition 1.

Figure 1.5 depicts a nice example. Let's take point o as an example, with k = 3. The reachability distance between o and p2 is simply their distance, since p2 is not part of the k-neighborhood of o. But since p1 is, their reachability distance gets "pushed back" to the k-distance of o. The reason is that in so doing, the statistical fluctuations of d(p,o) for all the p's close to o can be significantly reduced.

From now on, we will refer with MinPts to the minimum number of objects that make up a cluster. With that in mind, we can define:

Definition 4 (Local reachability density)**.** The Local Reachability density of an object p is defined as:

$$lrd_{MinPts}(p) = \frac{1}{\frac{\sum_{o \in N_{MinPts}(p)} reach-dist_{MinPts}(p,o)}{|N_{MinPts}(p)|}}$$

Simply put, the LRD is the inverse of the average reachability distance based on the Minpts-neighborhood of p. For instance, if all the Minpts-neighbors of p are in the same exact spot of p, the local reachability distance of p will be ∞. This also entails that, if the reachability distance from p to its Minpts neighbors is higher, the local density will be - naturally - lower.

Therefore, the higher the local density of a point p, the higher it is closer or part of a well-defined cluster.

Definition 5 (Local Outlier Factor)**.** The Local Outlier Factor of p is defined as:

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|}$$

So the Local Outlier Factor of p is the average of the ratio of the local reachability density of p and the local reachability density of the MinPts around it. Basically, the lower lrd(p) and the higher the lrd of its MinPts neighbors, the higher the LOF of p will be. This is in order to capture the intuition that if p's local density is much smaller than the ones of its neighbors, it's likely that p is an outlier.

In fact, the paper also demonstrates (with a sketch proof) that the LOF for any point p "deep inside" a cluster C is approximately 1, with a margin of error of ϵ. This is because, of course, the local reachability density of a point in a cluster will be very similar to the ones of nearby points.

1.4.2. DBSCAN

Another Density-Based outlier metric is DBSCAN [8]. It is by far the most commonly used in literature, and simply put, it connects regions of points with high density. In particular, DBSCAN defines a cluster as:

Definition 6 (Cluster)**.** Let D be a database of points. A **cluster** C with respect to ϵ and MinPts is a non-empty subset of D satisfying the following conditions:

1. $\forall p, q$: if $p \in C$ and q is density-reachable from p wrt. ϵ and MinPts, then $q \in C$
2. $\forall p, q \in C$: p is density-connected to q wrt. ϵ and MinPts.

This definition makes use of some terms, like density-reachable and density-connected, that needs to be defined as well.

Definition 7 (ϵ-neighborhood)**.** The ϵ-neighborhood of a point p, denoted by $N_\epsilon(P)$, is defined as: $N_\epsilon(P) = q \in D | dist(p, q) \leq \epsilon$

Definition 8 (Direct density-reachability)**.** A point p is directly density-reachable from a point q wrt. ϵ, MinPts if:

$p \in N_\epsilon(q)$ and $|N_\epsilon| \geq$ MinPts

where MinPts is the minimum amount of points that must be inside the ϵ -neighborhood of a point inside a cluster.

Definition 9 (Density - reachability)**.** A point p is density-reachable from a point q wrt. ϵ and MinPts if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$ and $p_n = p$ such that p_{i+1} is density reachable from p_i.

These definitions are all fairly simple. In order to fully understand definition 6, we still only need to define Density connectivity:

Definition 10 (Density - connectivity)**.** A point p is density-connected to a point q wrt. ϵ and MinPts if there is a point o such that both p and q are density-reachable from o wrt. to ϵ and MinPts.

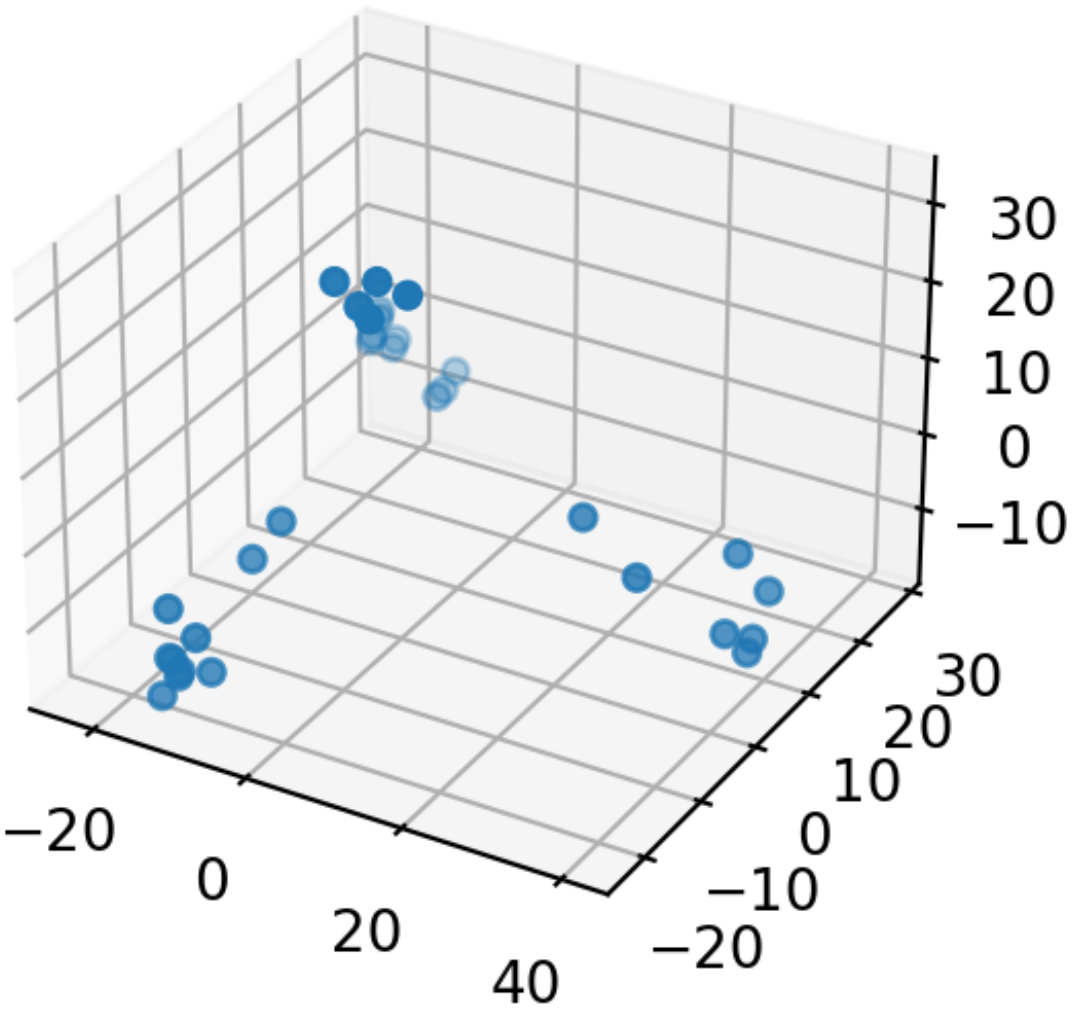

Figure 1.4: Example of a cloud of vectors representing a document, in three dimensions.

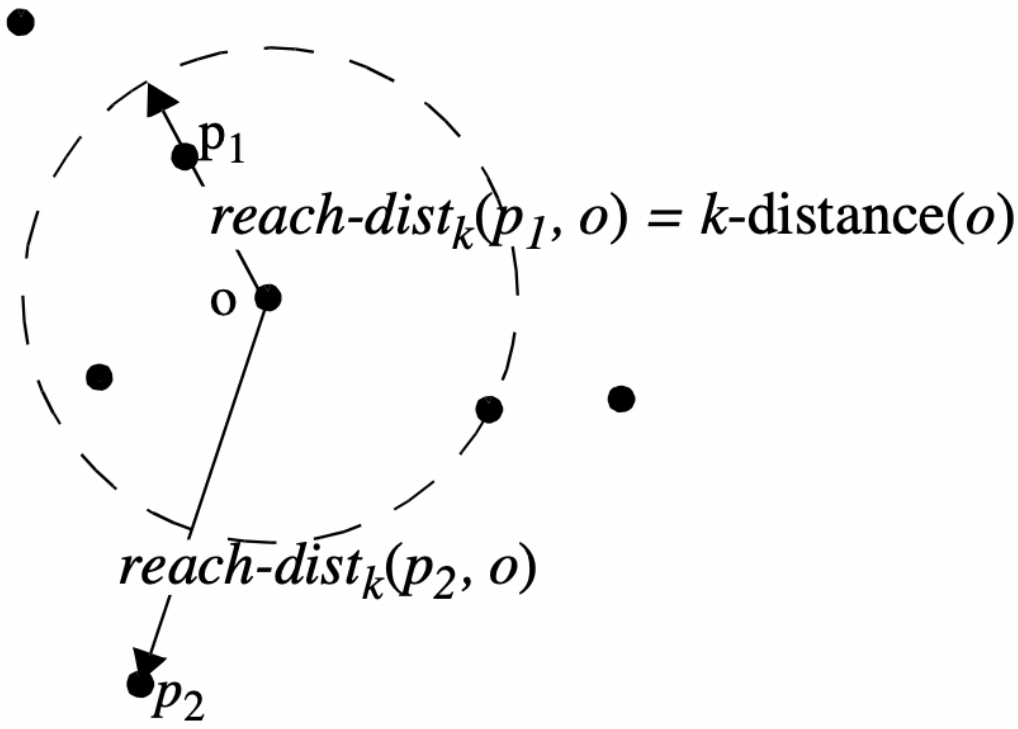

Figure 1.5: Example of reach distances from o for k = 3

Bibliography

[1] *Advances in Information Retrieval*. Springer Cham.

[2] W. Yang, H. Zhang, and J. Lin, “Simple applications of bert for ad hoc document retrieval.”

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018, cite arxiv:1810.04805Comment: 13 pages. [Online]. Available: http://arxiv.org/abs/1810.04805

[5] L. Bitawi, “Density-based approach for information retrieval documents ranking in embedding space.”

[6] D. R. D. Ganguly, M. Mitra, and G. J. Jones, “Representing documents and queries as sets of word embedded vectors for information retrieval.”

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: Identifying density-based local outliers,” *SIGMOD Rec.*, vol. 29, no. 2, p. 93–104, may 2000. [Online]. Available: https://doi.org/10.1145/335191.335388

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proc. of 2nd International Conference on Knowledge Discovery and*, 1996, pp. 226–231.