# DEFAULT at CheckThat! 2024: Retrieval Augmented Classification using Differentiable Top-K Operator for Rumor Verification based on Evidence from Authorities

Sayanta Adhikari[1,*,†], Himanshu Sharma[1,†], Rupa Kumari[1,†], Shrey Satapara[1] and Maunendra Desarkar[1]

[1]*Indian Institute of Technology Hyderabad, Telangana, 502285, India*

**Abstract**

The paper describes Team DEFAULT's submission at CheckThat! 2024 Task-5 on Rumor Verification based on Evidence from Authorities: In this paper, we present an approach for rumor verification on Twitter, focusing on integrating evidence from authoritative accounts to determine the veracity of rumors. We propose an architecture and a training regime as the preferred method to ensure seamless gradient flow. We formulate rumor verification using evidence from authorities as a Retrieval-Augmented Classification (RAC) task. By re-parameterizing the Top-K operator and applying Entropy-based Smoothing, our method addresses the discontinuity issues faced after retrieval, enhancing the accuracy of rumor verification. Using this classification-aware retrieval, the retriever achieves Recall@5 0.778, outperforming the baseline, placing team DEFAULT third on the test data leaderboard for retrieval. For classification, our approach performs on par with the baseline.

**Keywords**

Rumor Verification, Retrieval Augmented Classification, Differential TopK, Optimal Transport

## 1. Introduction

In the present era, social media has become one of the widely used mediums for information sharing due to its capabilities of fast information sharing at a low cost. This has made online social media a preferred choice for many individuals and organizations for propaganda-driven misinformation sharing to influence public opinions and decisions [1]. The spread of rumors and misinformation through social media has become a significant concern. Verifying the veracity of rumors and combating the dissemination of misinformation is crucial for maintaining the integrity of online discourse. This paper proposes a novel approach, Retrieval-Augmented Classification (RAC), which combines document retrieval and classification techniques to address the problem of rumor verification [2].

The shared task "Rumor Verification using Evidence from Authorities" at CHECKTHAT! LAB at CLEF-2024 [3, 4] related to rumor verification contains two steps approach. The first step involves document retrieval, wherein authoritative tweets related to a rumor are analyzed to identify the most relevant tweets. These sources, including reputable organizations or subject matter experts, can provide valuable evidence supporting or refuting the rumor. The second step is classification, where the retrieved evidence is leveraged to determine the rumor's veracity, categorizing it as Supported (TRUE), Refuted (FALSE), or Unverifiable (NEUTRAL).

To illustrate the methodology, consider a scenario involving a rumor circulating on social media about a potential disease outbreak. The RAC method would first retrieve relevant documents using sophisticated algorithms. These documents would then be analyzed based on key features identified

The code can be found at https://github.com/SAYANTA-ADHIKARI/RAC-SOFT-TopK

*CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France*

*Corresponding author.

†These authors contributed equally.

ai22mtech12005@iith.ac.in (S. Adhikari); ai22mtech12008@iith.ac.in (H. Sharma); rupa06012000@gmail.com (R. Kumari); ai22mtech02003@iith.ac.in (S. Satapara); maunendra@cse.iith.ac.in (M. Desarkar)

0009-0008-3717-9223 (S. Adhikari); 0009-0000-6189-515X (H. Sharma); 0000-0001-6222-1288 (S. Satapara); 0000-0003-1963-7338 (M. Desarkar)

by machine learning models. The rumor would be classified as *true* if these sources corroborate the outbreak with compelling evidence. Conversely, if the sources refute the claim or lack sufficient evidence, the rumor would be labelled as *false* or *unverifiable*, respectively.

In traditional retrieval systems, the relevance of a document is determined solely by its similarity to the query. However, for tasks like rumor verification, the evidence required to validate or refute a claim may not necessarily resemble the claim itself. This discrepancy between the query and the desired evidence can lead to suboptimal retrieval performance when using traditional similarity-based techniques.

To address this challenge, we proposed a classification-aware retrieval approach by providing an alignment between the retriever and the classifier, resulting in better retrieval. To jointly train the retriever and classifier, we removed discontinuity associated with the Top-K document selection for retrieval by replacing it with Soft Top-K, which allows the gradients to flow between retrieval and classification module, resulting in end-to-end training using a common loss function. Details about the proposed approach and its performance, along with analyses, are discussed in the subsequent sections of the paper.

## 2. Related Works

Rumor verification and Fact-checking are well-known tasks in NLP that have attracted many researchers. There has been a lot of work related to rumor verification related to dataset collection and training methods. Fact Checking[5] is one of the early works on claim verification collected from claim verification websites. Fact Extraction and Verification (FEVER)[6] is a well-known shared task organized at SemEVal for fact verification.

Liu et al. (2020) [7] proposed an approach using a Kernel Graph Attention Network (KGAT). Bekoulis et al. (2021) [8] emphasized the importance of evidence-aware sentence selection, while Kruengkrai et al. (2021) [9] presented a multi-level attention model for integrating evidence. These studies provide valuable insights for developing effective RAC systems for rumor verification using evidence from authorities.

Recent rumor verification research on retrieval augmented verification [10] integrates retrieval and classification using a zero-shot approach by retrieving real-time web-scraped evidence and matching claim tests using pretrained language systems. Their graph-structured representation gathers evidence automatically and highlights unverifiable claim parts. There has been some work for a comprehensive rumor debunking system using an LLM (involving retrieval, discrimination, and guided generation)[11]. Various systems have been developed to enhance the extraction and application of clinical trial information. One such system is CliVER [12], an end-to-end system that uses retrieval-augmented techniques to automatically retrieve clinical trial abstracts, extract pertinent sentences, and apply the PICO framework to support or refute scientific claims. This system represents a significant advancement in integrating artificial intelligence and clinical research methodologies, streamlining the process of evidence synthesis and decision-making in clinical settings.

## 3. Preliminary

### 3.1. Retrieval Augmented Classification

Verifying facts or rumors is challenging due to the subjective nature of the task. It requires access to contextual information regarding the domain from the current timeline. The verification task can be reduced to evidence retrieval and claim verification based on the retrieved evidence. This aligns closely with the domain of retrieval-augmented generation (RAG) [13], where the task is to generate an answer in context with a retrieved document. Similarly, we posed rumor verification based on evidence from authoritative sources as a RAC task where a class needs to be predicted based on the original claim and retrieved evidence.

RAC can be approached in two ways: 1) training the retriever and classifier independently ( *Independent Training*), and 2) training the retriever and classifier together ( *Joint Training*). Independent training allows each component to be trained separately and then combined. However, a major drawback is the lack of alignment between the retrieval and classification processes despite being a pipeline. The classifier's performance is inherently linked to the retrieval quality, contradicting the notion of independence. The dependency between the retrieved relevant evidence and the classification of the given rumor highlights the need for a joint training objective. Joint Training allows for alignment between the retriever and classifier components, but the major challenge is the discontinuity between these processes.

### 3.2. Differential Top-K

To address the issue of discontinuity (Figure 1(a)), we referred to an Optimal Transport (OT) trick for reparameterizing the Top-K function with entropy regularisation (to make it smooth) [14]. This technique first formulates the extraction of Top-K elements from a vector into an Optimal Transport Problem and then applies entropy regularisation to facilitate smooth gradient flow. We used SOFT (Scalable Optimal transport-based diFferenTialble) Top-K operator in place of the Top-K operator to get the Top K elements.

**1. Problem Formulation** Consider the score vector (containing relevance scores for each of the tweets concerning the rumor tweet) to be $X = \{x_i\}_{i=1}^{n}$, where $n$ is the total number of tweets provided in the timeline. The standard top-$k$ operator returns indexes with Top-K elements, which is equivalent to a vector $A = [A_1, ..., A_n]$, such that

$$A_i = \begin{cases} 1 & \text{if } x_i \text{ is one of the top-}k \text{ relevant tweets in } X \text{with respect to the rumor tweet,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Using $A$, we can extract the Top-K elements from $X$. In the case of sorted Top-K, $A$ is a matrix that, when multiplied with the input $X$, provides us with the Top-K elements in sorted order.

**2. Re-parameterizing Top-K Operator as OT Problem:** Now, let's consider the probability associated with the score vector, $X = \{x_i\}_{i=1}^{n}$ and the output support space, $B = \{0, 1\}$ (0 to map all the Top-K elements and 1 for the remaining, $m = 2$) be $\mu = \frac{1}{n}\mathbf{1}_n$ and $\nu = \left[\frac{k}{n}, \frac{n-k}{n}\right]$ respectively, where $n$ is the total number of timeline tweets and $k$ represents the total number of evidence tweets that needs to be retrieved from the timeline.

$$\Gamma^* = \text{argmin}_{\Gamma \geq 0} \langle C, \Gamma \rangle, \quad \text{s.t.,} \quad \Gamma \mathbf{1}_m = \mu, \quad \Gamma^T \mathbf{1}_n = \nu, \quad \Gamma, \Gamma^* \in \mathrm{R}^{n \times m} \tag{2}$$

Here, $\Gamma_{i,j}$ represents the probability of mapping the input $x_i$ of $X$ to the output $b_j$ of $B$ and $c_{i,j}$ of $C$ represents the cost incurred to move from $x_i$ to $b_j$. Here, $\Gamma$ represents the joint probability distribution over the support $X$ cartesian product $B$.

**3. Solution:** Under the above conditions, the optimal transport plan $\Gamma^*$ is given by (in closed form):

$$\Gamma^*_{\sigma_i,1} = \begin{cases} \frac{1}{n}, & \text{if } i \leq k, \\ 0, & \text{if } k+1 \leq i \leq n \end{cases}, \quad \Gamma^*_{\sigma_i,2} = \begin{cases} 0, & \text{if } i \leq k, \\ \frac{1}{n}, & \text{if } k+1 \leq i \leq n \end{cases} \tag{3}$$

where $\sigma$ being the sorting permutation, i.e., $x_{\sigma_1} < x_{\sigma_2} < \cdots < x_{\sigma_n}$. Based on $\Gamma^*$ we define $A = n\Gamma^* \cdot [1, 0]^T$. The matrix $A$ is the mapping matrix that provides the position of Top-K elements.

**4. Smoothing by Entropy Regularization:** Employing entropy regularisation to the OT problem yields a smoothed approximation. The OT optimization problem further changes to:

$$\Gamma^*_\epsilon = \text{argmin}_{\Gamma \geq 0} \langle C, \Gamma \rangle + \epsilon H(\Gamma), \quad \text{s.t.,} \quad \Gamma \mathbf{1}_m = \mu, \;\; \Gamma^T \mathbf{1}_n = \nu, \;\; \epsilon > 0$$

where $H(\Gamma) = \sum_{i,j} \Gamma_{i,j} \log \Gamma_{i,j}$ is the entropy regularizer. Based on the above $\Gamma^*_\epsilon$ we define: $A^\epsilon = n\Gamma^{*,\epsilon} \cdot [1, 0]^T$ as the smoothed counterpart of the standard top-$k$ operator output ($A$ in Equation 1). Throughout our approach, we consider sorted Top-K. Using the Soft-Top-K operator in place of the Top-K operator helps train the model end-to-end and thus helps in aligning the retriever and the classifier accordingly.

# 4. Methodology

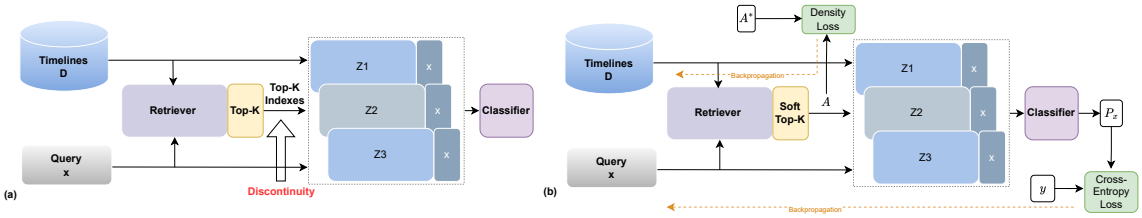


**Figure 1:** (a) This figure illustrates the discontinuity arising between the Retriever and Classifier phases, primarily because of the involvement of indices in the Top-K relevant tweets selection process for a given query tweet ($x$) among the timelines ($D$). (b) This figure provides the final architecture of our approach, where the Top-K operator is replaced with the Soft Top-K operator. The output $A$ of Soft Top-K is then used along with Timeline to get the embeddings related to Top-K evidences (denoted by $z1, z2, z3$). Then, these are passed through the classifier to get probabilities, which are then used to compute loss (Cross-Entropy Loss). Density loss is also used to guide the retriever further. The orange dotted lines in the figure show the flow of gradients for our architecture.

To perform RAC for rumor verification based on evidence from authorities, we propose a novel architecture that can be trained end-to-end. We propose a transformer-encoder-based architecture as a retriever followed by a Top-K operator to extract relevant evidence. This is then used to help the classifier classify the Query Tweet(x). As shown in Figure 1(a), this Top-K operator provides a discontinuity in the pipeline. To remove this discontinuity, we parameterized Top-K with a smoother version, Soft Top-K (details provided in subsection 3.2). Figure 1(b) shows the final architecture along with different losses (defined in subsection 4.1) that are used for training purposes.

If the classifier cannot classify correctly, then it won't be able to guide the retriever regarding the relevance of the tweets and vice versa. So, providing models with no information about the downstream tasks might lead to poor performance and sub-optimal convergence. To counter this effect, we propose a training method for our architecture. In this, we will first independently train the classifier and then. We will jointly train both the retriever and classifier. Further, we freeze the retriever and train the classifier again to increase the classifier's performance.

We define a Retriever, $R$, parameterized by $\psi$. It computes embeddings for each document (Timelines $D$) and the Query Tweet $x$. The similarity score between the embedding of $x$ and the embedding of $D$ is used to extract the relevant tweets. We define the score for $D$ as $S_D$. To extract the Top-K relevant tweets, we pass it through *Soft Top-K* function, which returns a matrix $A$, which gives us information about the Top-K relevant document indexes. We multiply the matrix $A$ with $D$ to extract Top-K relevant documents. As we are using Soft Top-K, directly multiplying $A$ to $D$ leads to a change in the token ids of the word, so we multiply $A$ with the classifier embedding corresponding to $D$ to get Top-K document embeddings. We define the classifier $H$ as a combination of two functions, $E$ parameterized by $\theta$ and $C$ parameterized by $\phi$. Here, $E$ represents the initial embedding layer of the BERT model, and $C$ represents the classification head. The classifier can be represented as a composition of $E$ and $C$, i.e., $H = C \circ E$. The classifier verifies $x$ in context with the embeddings of the extracted evidences ($Z = \{z_i\}_{i=1}^{K} = A \times E([x, D]; \theta)$). Providing this evidence with $x$ might lead to an overflow of the model's context window. To deal with this problem, we get logits concerning each evidence, i.e., $P_i = C(z_i, \phi), i = 1, 2, \cdots, K$, and then perform a weighted aggregation of logits using the relevance scores, i.e., $S_K = \{s_i\}_{i=1}^{K} = A \times S_D$, provided by the retriever for each of the evidence. The probability associated with query tweet $x$ based on the evidence set $Z$, denoted as $P_x$;

$$P_x = \frac{\sum_{i=1}^{K} s_i.P_i}{\sum_{i=1}^{K} s_i} \tag{4}$$

### 4.1. Losses and Optimization Objective

To train our model, we use cross-entropy loss ($\mathcal{L}_{CE}$) using $P_x$ and ground truth value $y$

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}.\log(P_{x_ic}) \tag{5}$$

where $N$ denotes the total number of samples and $C$ denotes the total number of classes. To provide better guidance to the Retriever, we introduced a new loss term, called the **Density Loss** $\mathcal{L}_{DL}$ over the output of the Soft-Top-K operator. The Soft-Top-K operator returns a matrix $A$ to provide the tweets that must be considered. While forming the data, we are already aware of those positions so that we can provide the ground truth $A^*$ matrix. We compute the density loss as a mean of cross-entropy of each row of $A$ for each row of $A^*$. Mathematically,

$$\mathcal{L}_{DL} = -\frac{1}{Nr}\sum_{i=1}^{N}\sum_{j=1}^{r}\sum_{c=1}^{C} A_i^*[j,c].\log\left(A_i[j,c]\right) \tag{6}$$

where $A_i$ and $A_i^*$ represents the predicted and ground truth $A$ matrix for the query input $x_i$ and $r$ denotes the total number of rows in $A_i$. The final loss is an aggregation of these two losses. As both the losses are of the same scale, we add them with equal weight. Based on the defined losses, we define our optimization problem as

$$\arg\min_{\theta,\phi,\psi} \mathcal{L}_{CE} + \mathcal{L}_{DL} \tag{7}$$

In practice, we use Adam optimizer to train this objective. For more details regarding the training process, refer to Algorithm 1. The required datasets for training as per Algorithm 1 are independent training classifier datasets ($\mathcal{D}_C$) and joint training datasets ($\mathcal{D}_J$). Further details of this dataset are provided in subsection 5.2.

## 5. Experimental Setup

### 5.1. Dataset Description

We utilized the dataset from CLEF 2024 CheckThat! Lab, Task 5 [4] includes Twitter data curated for rumor verification in English and Arabic. Our experiments focused on the English dataset, comprising 96 training and 32 validation samples. Each sample contains an **id** (unique identifier), **rumor** (tweet text), **timeline** (tweets from authorities during the rumor's timeframe), **label** (veracity: SUPPORTS, REFUTES, or NOT ENOUGH INFO), and **evidence** (tweets from authorities aiding classification). We augmented the dataset to increase the sample size.

**Table 1**
Table provides us with the final count of samples we got after augmenting the dataset samples.

| Dataset-stats | | Parts | Train samples | Val samples |
|---|---|---|---|---|
| Provided data | - | - | 96 | 32 |
| Total data created with augmentation provided in Section 5.2 | Independent Training | Retriever | 233 | 48 |
| | | Classifier | 297 | 51 |
| | Joint Training | - | 305 | 41 |

**Algorithm 1** Training Regime

**Input:** Independent Classification Dataset $\mathcal{D}_C$, Joint Training Dataset $\mathcal{D}_J$, Epochs $(T_{C_1}, T_{C_2}, T_J)$
**Parameters:** Number of Evidences ($k$), epsilon ($\epsilon$), Classifier Embedding function ($E$) parameterized by $\theta$, Classification function ($C$) parameterized by $\phi$, Retriever Function ($R$) parameterized by $\psi$

1: Initialize $H = C \circ E$ (Refer section 4). ▷ Can use pretrained weights.
2: **for** $t = 1, 2, \ldots, T_{C_1}$ **do** ▷ Initial training of classifier
3: **for** $(z, y) \in \mathcal{D}_C$ **do** ▷ Batched Training is also possible
4: Optimize $\theta, \phi$ using $\mathcal{L}_{CE}$ (Equation 5)
5: **end for**
6: **end for**
7: Initialize $R$ ▷ Can use pretrained weights.
8: **for** $t = 1, 2, \ldots, T_J$ **do** ▷ Joint Training
9: **for** $(x, D, A^*, y) \in \mathcal{D}_J$ **do**
10: $S_D \leftarrow R(D, x; \psi)$ ▷ Get Scores for all the documents
11: $A \leftarrow Soft\_TopK(S_D; k, \epsilon)$ ▷ Get Top-K indices
12: $S_k \leftarrow A \times S_D$ ▷ Get Top-K Scores($S_k$)
13: $Z_k \leftarrow A \times E([x, D]; \theta)$ ▷ Get Top-K Document Embedding ($Z_k$)
14: $P_k \leftarrow C(Z_k; \phi)$ ▷ Get Prediction Probabilities($P_k$)
15: Using $P_k$ and $S_k$, compute $P_x$ using Equation 4
16: Optimize $\theta, \phi, \psi$ using Equation 7
17: **end for**
18: **end for**
19: Freeze $\psi$
20: **for** $t = 1, 2, \ldots, T_{C_2}$ **do** ▷ Further training of classifier for boosting performance
21: **for** $(x, D, A, y) \in \mathcal{D}_J$ **do**
22: Repeat Steps 10 to 15 to get $P_x$
23: Optimize $\theta, \phi$ using $\mathcal{L}_{CE}(P_x, y)$ (Equation 5)
24: **end for**
25: **end for**
26: Return $(\theta, \phi, \psi)$

## 5.2. Data Augmentation and Training

### 5.2.1. Independent Training

**Retriever:** For independent training of the Retriever, we use contrastive training [15]. Each rumor tweet is paired with an evidence tweet as a positive sample and $l$ (3 in our experiments) non-evidence tweets from the timeline as negative samples. These triplets train the model with contrastive loss functions [15, 16]. We create multiple samples with randomly chosen negative tweets for robustness and exclude samples without any evidence tweets. We have considered multiple score functions for scoring the similarity between the tweets: (a) Euclidean Distance between the representation vectors, (b) Cosine similarity between the two representation vectors, and (c) MaxSim similarity proposed in the paper of ColBERT [15]. We initialize the retriever with colber-ir/colbertv2.0 checkpoint weights from huggingface. To further finetune the model, we use a batch size of 1 (fixed), epochs as 5 (using early stopping), learning rate as $5e-5$ with similarity score as MaxSim, and contrastive loss as provided in [15].
**Classifier:** For independent training of the Classifier, we create tweet pairs. Each rumor tweet is paired with an evidence tweet and labelled according to the original data label ("SUPPORTS" or "REFUTES") or paired with a non-evidence tweet and labelled as "NOT ENOUGH INFO." This process ensures a balanced class distribution in the final training dataset. After initializing with pretrained weights, we used a batch size of 2 (fixed), epochs of 7 (got using early stopping) and a learning rate of $1e-5$ to

fine-tune the classifier.

#### 5.2.2. Joint Training

For joint training, each rumor tweet is paired with a document set of size $n$ (64 in our experiments). The document set includes all, some, or none of the evidence tweets, filled to $n$ with non-evidence timeline tweets. Document sets with evidence are labelled based on the original data point, while those without evidence are labelled "NOT ENOUGH INFO." We shuffle the document sets to avoid bias from tweet order and ensure a balanced class distribution in the final dataset. To train the model, we used a batch size of 1 (fixed), with a learning rate of $1e-5$, a K value of 5 (given), and the epsilon value of Soft Top-K as 0.01 (fixed). We train the model for 5 epochs (using early stopping).

#### 5.2.3. Our Approach

As joint training starts training from pretrained weights, it is difficult for the classifier to guide the retriever and vice-versa. In our approach, we first independently train the classifier on the independent training dataset with a similar hyperparameter provided in subsubsection 5.2.1 for 5 epochs (got using early stopping). Then, we finetune the whole architecture (Retriever + Classifier) end-to-end. We used the dataset presented in subsubsection 5.2.2 to perform joint training. We use the hyperparameters provided in subsubsection 5.2.2 for joint training. After this, we again finetuned the classifier with a frozen retriever to further boost the classifier's performance. To further train, we used a batch size of 1 (fixed), with a learning rate of $1e-5$ (fixed) for 5 epochs.

We provide the statistics about the dataset we got after performing augmentation in Table 1. Further, we used these data to train our model. As we have tweets as our input data, we preprocessed each tweet by removing the links in the tweet. We converted each of the emojis in the tweet with their relevant text translation using the 'emoji' python package [17]. We used a single NVIDIA Tesla 32GB V100 GPU to train our models. It took around an hour to train the whole model on the dataset.

### 5.3. Evaluation Metrics

The primary measure for evaluating evidence retrieval is **Mean Average Precision** (*MAP*). Under this metric, systems receive no credit for retrieving tweets related to unverifiable rumors. Another important evaluation metric is **Recall@5** (*R@5*), which measures the proportion of relevant tweets retrieved among the top 5 retrieved tweets. We utilize the ***Macro-F1*** (M-F1) score for classification evaluation, which calculates the harmonic mean of precision and recall across all classes. Additionally, we consider a ***Strict Macro-F1*** score, where the correctness of a rumor label is contingent upon at least one retrieved authority evidence being correct.

## 6. Results

Table 2 provides results related to different experiments we have performed. We can conclude from the results that MaxSim similarity performs better than the other similarities we considered. We also observed that those initialized with ColBERT pretrained weights performed better than those initialized with BERT pretrained weights. This is trivial as ColBERT is specifically trained for information retrieval and matches between individual tokens of the two texts instead of comparing the overall pooled vectors representing the two texts- claim and candidate evidence tweet. Inspection at finer granularity helps it identify the matches better. We can also see that Joint Training performs better than Independent Training. We can also observe that our proposed training curriculum performs better than both purely Joint and Independent Training. We can also observe that it reduces performance when different pretrained models are used for the retriever and the classifier. Overall, we can observe that ColBERT-B with our Approach performs best among all our approaches. This approach can beat KGAT's retriever performance by a huge margin, but the classification performance is less than that of KGAT.

**Table 2**
This table provides the results of different experiments we performed. Additionally, it also provides a comparison between the different training techniques we used. It also compares different similarity metrics and different pre-trained backbone models. Here, BERT-B [18] means using the *bert-base-uncased* pretrained checkpoint, and ColBERT-B [19] initializes with *colbert-ir/colbertv2.0* pretrained checkpoints. All the results provided in this table are on the validation data provided in the task.

| Pretrained Models | | Method | Similarity Metric | Inference Performance | | | |
|---|---|---|---|---|---|---|---|
| | | | | Retriever Performance | | Classifier Performance | |
| Retriever | Classifier | | | MAP | R@5 | M-F1 | Strict-M-F1 |
| Baseline | | | - | 0.561 | 0.636 | **0.508** | **0.508** |
| BERT-B | - | Zero-shot | MaxSim | 0.084 | 0.123 | - | - |
| ColBERT-B | - | Zero-shot | MaxSim | 0.164 | 0.229 | - | - |
| - | BERT-B | Independent Training | - | - | - | 0.296 | - |
| BERT-B | | | Cosine | 0.268 | 0.456 | 0.249 | 0.199 |
| BERT-B | | | L2-norm | 0.245 | 0.35 | 0.21 | 0.16 |
| BERT-B | | | MaxSim | 0.27 | 0.48 | 0.251 | 0.2 |
| ColBERT-B | | | MaxSim | 0.466 | 0.524 | 0.269 | 0.233 |
| BERT-B | | Joint Training from Scratch | Cosine | 0.308 | 0.412 | 0.195 | 0.117 |
| ColBERT-B | | | Cosine | 0.581 | 0.646 | 0.364 | 0.309 |
| ColBERT-B | | | MaxSim | 0.606 | 0.662 | 0.362 | 0.321 |
| ColBERT-B | BERT-B | Our Approach | MaxSim | 0.404 | 0.508 | 0.193 | 0.09 |
| BERT-B | | | MaxSim | 0.388 | 0.441 | 0.256 | 0.195 |
| ColBERT-B | | | MaxSim | **0.733** | **0.778** | 0.472 | 0.472 |

**Table 3**
This table provides results of our approach and baseline on the test dataset

| Method | Retriever Performance | | Classifier Performance | |
|---|---|---|---|---|
| | MAP | R@5 | M-F1 | Strict M-F1 |
| Baseline | 0.335 | 0.445 | **0.495** | **0.495** |
| Our Approach | **0.559** | **0.634** | 0.482 | 0.454 |

Table 3 presents the results obtained from the test data provided by the CHECKTHAT! LAB for the specified task. From the result, it is evident that classifier-guided retriever training outperforms the baseline by a huge margin and classifier performance is on par with that of the baseline.

## 7. Conclusion

We present a joint training framework to simultaneously optimize an evidence retriever and a rumor classifier in an end-to-end fashion. We show that our approach performs better than both independent and joint individually. Our experiments have shown that merging these two approaches together leads to better performance. From the results, we can conclude that our approach can retrieve relevant tweets accurately and it can extract at least one relevant tweet for all the rumor claims as Macro-F1 and Strict-Macro-F1 are the same for ColBERT-B with our Approach.

Also, the results show the importance of joint training. Using the Soft Top-K operation as a differentiable approximation of the standard Top-K operation can not only encounter discontinuity but enhance the model's performance. Further, we conclude that having Soft-TopK-based reparameterization and independent training followed by joint training leads to better performance. Moreover, we observe that the classifier-guided retriever boosts the performance of the retriever, such that it outperforms the baseline by a huge margin, whereas the classifier's performance is on par with the baseline.

## References


[1] D. Varshney, D. K. Vishwakarma, A review on rumour prediction and veracity assessment in online social network, Expert Systems with Applications 168 (2021) 114208. URL: https://www.sciencedirect.com/science/article/pii/S0957417420309362. doi:https://doi.org/10.1016/j.eswa.2020.114208.

[2] J. Yu, J. Jiang, L. M. S. Khoo, H. L. Chieu, R. Xia, Coupled hierarchical transformer for stance-aware rumor verification in social media conversations, Association for Computational Linguistics, 2020.

[3] A. Barrón-Cedeño, F. Alam, T. Chakraborty, T. Elsayed, P. Nakov, P. Przybyła, J. M. Struß, F. Haouari, M. Hasanain, F. Ruggeri, X. Song, R. Suwaileh, The CLEF-2024 CheckThat! Lab: Check-worthiness, subjectivity, persuasion, roles, authorities, and adversarial robustness, in: N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, I. Ounis (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2024, pp. 449–458.

[4] F. Haouari, T. Elsayed, R. Suwaileh, Overview of the CLEF-2024 CheckThat! Lab Task 5 on Rumor Verification using Evidence from Authorities, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CLEF 2024, Grenoble, France, 2024.

[5] A. Vlachos, S. Riedel, Fact checking: Task definition and dataset construction, in: Proceedings of the ACL 2014 workshop on language technologies and computational social science, 2014, pp. 18–22.

[6] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, A. Mittal, The fact extraction and VERification (FEVER) shared task, in: J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, A. Mittal (Eds.), Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1–9. URL: https://aclanthology.org/W18-5501. doi:10.18653/v1/W18-5501.

[7] Z. Liu, C. Xiong, M. Sun, Z. Liu, Fine-grained fact verification with kernel graph attention network, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7342–7351. URL: https://aclanthology.org/2020.acl-main.655. doi:10.18653/v1/2020.acl-main.655.

[8] G. Bekoulis, C. Papagiannopoulou, N. Deligiannis, Understanding the impact of evidence-aware sentence selection for fact checking, in: A. Feldman, G. Da San Martino, C. Leberknight, P. Nakov (Eds.), Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, Association for Computational Linguistics, Online, 2021, pp. 23–28. URL: https://aclanthology.org/2021.nlp4if-1.4. doi:10.18653/v1/2021.nlp4if-1.4.

[9] C. Kruengkrai, J. Yamagishi, X. Wang, A multi-level attention model for evidence-based fact checking, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 2447–2460. URL: https://aclanthology.org/2021.findings-acl.217. doi:10.18653/v1/2021.findings-acl.217.

[10] A. U. Dey, A. Llabrés, E. Valveny, D. Karatzas, Retrieval augmented verification: Unveiling disinformation with structured representations for zero-shot real-time evidence-guided fact-checking of multi-modal social media posts, arXiv preprint arXiv:2404.10702 (2024).

[11] J. Xu, L. Xian, Z. Liu, M. Chen, Q. Yin, F. Song, The future of combating rumors? retrieval, discrimination, and generation, 2024. arXiv:2403.20204.

[12] H. Liu, A. Soroush, J. G. Nestor, E. Park, B. Idnay, Y. Fang, J. Pan, S. Liao, M. Bernard, Y. Peng, C. Weng, Retrieval augmented scientific claim verification, JAMIA Open 7 (2024) ooae021. doi:10.1093/jamiaopen/ooae021.

[13] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[14] Y. Xie, H. Dai, M. Chen, B. Dai, T. Zhao, H. Zha, W. Wei, T. Pfister, Differentiable top-k with

optimal transport, Advances in Neural Information Processing Systems 33 (2020) 20520–20531.
[15] O. Khattab, M. Zaharia, Colbert: Efficient and effective passage search via contextualized late interaction over bert, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 39–48.
[16] I. Malkiel, D. Ginzburg, O. Barkan, A. Caciularu, Y. Weill, N. Koenigstein, Metricbert: Text representation learning via self-supervised triplet training, in: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 1–5. doi:10.1109/ICASSP43922.2022.9746018.
[17] emoji — pypi.org, https://pypi.org/project/emoji/, 2024. [Accessed 31-05-2024].
[18] J. D. M.-W. C. Kenton, L. K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019, pp. 4171–4186.
[19] O. Khattab, M. Zaharia, Colbert: Efficient and effective passage search via contextualized late interaction over bert, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 39–48.