

Top-Down Hierarchical GMM and Distribution Trending

Arvid Kok

NATO Communications and Information (NCI) Agency
The Hague, The Netherlands
arvid.kok@ncia.nato.int

Adelica Ndoni

NATO Communications and Information (NCI) Agency
The Hague, The Netherlands
adelica.ndoni@ncia.nato.int

António Carvalho

NATO Communications and Information (NCI) Agency
The Hague, The Netherlands
antonio.carvalho@ncia.nato.int

Michael Street

NATO Communications and Information (NCI) Agency
The Hague, The Netherlands
michael.street@ncia.nato.int

Abstract—In the last decade, we witnessed significant technological advancements that had huge impacts across all different aspect of everyday life. New technologies present important topics that need to be overseen from their early beginnings to be able to capture their break through and prepare organizations for needed transitions. This research tries to detect weak signals of emerging technologies based on scientific articles published over the last decade covering several categories.

To capture emerging technology trends, topics, we propose a method we refer to as "Top-Down Hierarchical GMM and Distribution Trending", including our clustering algorithm optimized for (very) high-dimensional data. In our method we take advantage of state-of-the-art transformer-based architectures in Natural Language Processing (NLP) for capturing the topics represented in scientific articles. Our "Top-Down Hierarchical GMM-EM" algorithm detects topic clusters, without the requirement to bring down the dimensionality (which would cause information loss). The algorithm detects the right number of clusters and the results include their variance, density and back-trace to the input articles. In the final step of the method the course that all clusters take over time is trended.

We believe our method and optimized algorithm can give valuable insights about when topics emerge and are picked-up by the larger community. Low-variance clusters that trend to increase in density (more articles on a narrow topic) can be a weak signal of an emerging technology; low-variance clusters that trend to higher variance can signal the spread of technology to new areas. Combined with other insights like affiliations with nations, organizations or academics, the method can help reveal the bigger picture.

Index Terms—Weak Signal, ArXiv, Scientific Articles, Transformer Model, BERT, Time-series, Topic Trending, Gaussian Mixed Model, GMM, Hierarchical Clustering, Natural Language Processing, NLP, Text Sequence Embedding

I. INTRODUCTION

The importance of tracking emerging and disruptive technologies for future military activities has long been recognized and is reflected in current NATO and national military strategies. Their early detection plays an important role in military environments and is valuable input to help prepare for (future) battle-fields. For NATO to maintain its scientific and technological advantage, the NATO Science & Technol-

ogy Organization (STO) and NATO Chief Scientist research and report on the impact of technological trends. The work described in this paper feeds into this effort.

The ability to detect future technological changes is crucial to make the right decisions today. Early detection of new technologies enables organizations to prepare for opportunities and threats they present. A study from 2019 by the European Commission (EC) refers to the topic as "Weak Signals in Science and Technologies". [1]

As an alternative approach to – and in support of – capturing weak signals of emerging technologies, we propose a new method: "Top-Down Hierarchical GMM and Distribution Trending". The proposed method can be used in combination with metadata and other methods that enrich understanding of articles, topics and technology fields. In our wider research, we combined the method with information from Microsoft Academic Graph Database [2] and another method we introduced and named "Carbon-dating Articles with Transformer Series" [3].

Weak signals in scientific articles:

- What indicators can be found in scientific articles of emerging technology;
- What impact will new technology have on the NATO alliance;
- What countries, industries and academia are investing in the technology that can influence the future;
- What topics and technologies should NATO invest in preparing for a safe future for over 1 billion people?

Top-Down Hierarchical GMM and Distribution Trending:

- What clusters of scientific articles – by what countries, organizations and academia – are showing increased attention, with increasing numbers of articles (density increase);
- If scientific articles are representative for what scientific topics are relevant/trending, how much influence – driving the focus of nearby topics (variance increase) – do these topics have?

This paper describes a method that can help answering the questions above.

II. DATASETS

For the purpose of (scientific) articles clustering and trending the method describes the use of a transformer natural language model (see III) on scientific articles to extract topic-representing features, also known as text embeddings. The method is flexible and can be applied on many text-based datasets, as long as they comply with a few rules:

- All articles (or other entity) are uniquely identifiable.
- The text sequence(s) for each article are information dense, so that represented topics in dataset are well distinguishable (e.g. abstracts).
- All articles to be used for clustering have a known publish date.

A. ArXiv up-to-date

Our research utilized the ArXiv dataset and metadata of – on date of extraction – 1.9M+ scholarly papers as offered on Kaggle. [4]

“For nearly 30 years, ArXiv has served the public and research communities by providing open access to scholarly articles, from the vast branches of physics to the many subdisciplines of computer science to everything in between, including math, statistics, electrical engineering, quantitative biology, and economics. This rich corpus of information offers significant, but sometimes overwhelming depth. In these times of unique global challenges, efficient extraction of insights from data is essential. To help make the arXiv more accessible, we present a free, open pipeline on Kaggle to the machine-readable arXiv dataset: a repository of 1.7 million articles, with relevant features such as article titles, authors, categories, abstracts, full text PDFs, and more...”

To prepare the ArXiv data set for our experiment, the metadata for each article was retrieved, including the unique identifier, the most recent publish date (latest version) and the corresponding abstract text.

B. Synthetic data

Our research was technically tested with synthetic data. This data allowed us to test specific behavior of the method on different aspects. Characteristics are explained in the sections where synthetic data is used.

III. PROBABILITY DENSITY FUNCTION

In this section we will take a quick look at the theory behind the concepts that we apply for this algorithm. We also take the chance to introduce our notation.

Our model assumes the dataset follows a Gaussian mixed model, that is, it assumes the dataset can be separated into the sum of various Gaussian distributions. We are also interested in applying the algorithm to high dimensional datasets such

as BERT [5] text embeddings of ArXiv paper abstracts [4], therefore, we showcase the probability density function (PDF) of Gaussian distributions at high dimensions and the necessary transformations that we had to perform to this function to tackle a few problems one is faced with when attempting our objective, which is, to use the PDF as a basis for the input of an expectation maximization algorithm we will describe further ahead.

A. Gaussian Distribution PDF

A random variable x is said to follow a Gaussian (or normal) distribution, $\phi(\mu, \sigma)$, with μ as the mean of the distribution and σ is its standard deviation, when its probability density function follows the equation bellow [6]

$$PDF(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}. \quad (1)$$

This way, the distribution is completely defined by knowing the scalars μ and σ .

B. Multivariate Gaussian Distribution PDF

We now move to the high dimensional version of the Gaussian distribution, often called multivariate Gaussian distribution [7].

We consider a d -dimensional vector, \vec{x} , as our random variable. If we do this, the generalization of equation (1) for d dimensions takes the shape

$$PDF(\vec{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma^{-1}(\vec{x} - \vec{\mu})\right], \quad (2)$$

where the mean of the distribution, $\vec{\mu}$, is a vector with d dimensions and the covariance matrix of the distribution, Σ , is a $d \times d$ matrix with $|\Sigma|$ as its determinant. The distribution is fully defined and any dimension, d , once we know the vector $\vec{\mu}$ and matrix Σ .

IV. DEALING WITH HIGH DIMENSIONALITY

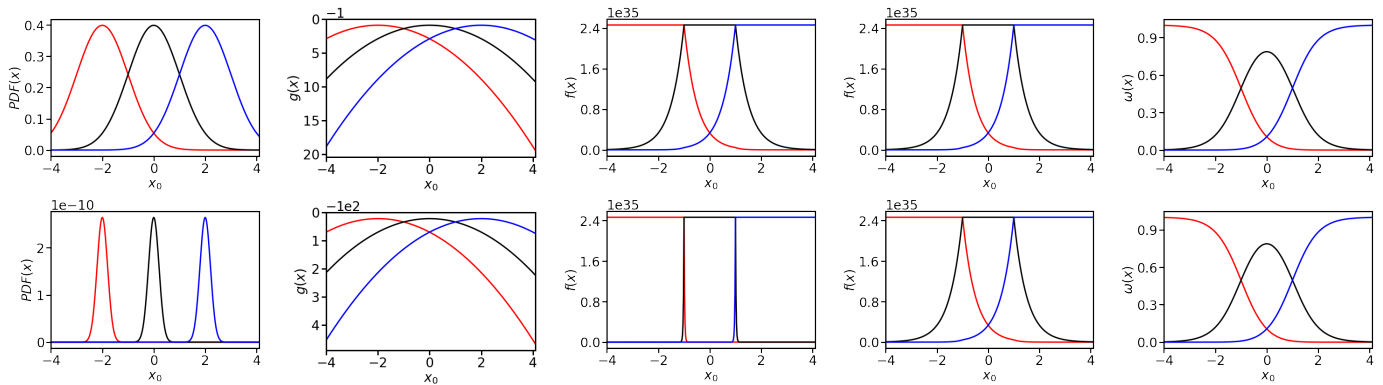
In this section we will talk about the problems that we faced when working with high dimensional data and how we solved them.

Once we start trying to evaluate equation (2) for large values of d we start facing numerical problems. The outputs become very small and we require very high precision in order to get meaningful values *i.e.*, values different than 0. This problem is exacerbated if \vec{x} takes values that are far away from $\vec{\mu}$ where equation (2) has its global maximum as can be seen for any of the clusters in Fig. 1 a).

Looking at equation (2) we can see how this problem can be expected since as the number of dimensions, d , increases, the values for $f(\vec{x})$ start decreasing exponentially.

A. Logarithm of the PDF

To tackle the precision problem we chose to evaluate the logarithm of the probability density function. This value can



a) Output slice of equation (2) for $d = 1$ and $d = 24$. No compression. b) Output slice of equation (3) for $d = 1$ and $d = 24$. No compression. c) Output slice of equation (4) for $d = 1$ and $d = 24$. No compression. d) Output slice of equation (4) for $d = 1$ and $d = 24$. With compression. e) Converged EM probability for $d = 1$ and $d = 24$. With compression.

Fig. 1: All the plots are one dimensional slices of the mentioned outputs for three clusters at $\bar{\mu}_1 = -2$, $\bar{\mu}_2 = 0$ and $\bar{\mu}_3 = 2$. Fig. 1 a) shows the output of equation (2), Fig. 1 b) shows the output of equation (3), Fig. 1 c) shows the output of equation (4). The final two plots are obtained after applying the compression shown in equation (6). Fig. 1 d) shows the output of equation (4) and Fig. 1 e) shows the output of equation (7).

be obtained by simply taking the natural logarithm of equation (2). Doing this yields

$$g(\vec{x}) \equiv \ln(PDF(\vec{x})) = -\frac{1}{2} \left[d \ln(2\pi) + \ln(|\Sigma|) + (\vec{x} - \bar{\mu})^T \Sigma^{-1} (\vec{x} - \bar{\mu}) \right]. \quad (3)$$

By evaluating this equation we are effectively changing the codomain of the probability density function expressed in equation (2) by replacing the values that are very close to zero (and therefore require high numerical precision to evaluate) with negative, non vanishing values. This makes the numerical precision problem easier to handle.

Equation (3) also preserves the important behaviour of decreasing monotonically as x moves away from μ . This is crucial since we intend to use the computed values on our EM algorithm.

The evaluation of the probability density function and its logarithm can be seen in Fig. 1 a) and Fig. 1 b) respectively. In Fig. 1 a) we can see how the codomain of equation (2) goes from 0 to $PDF(\bar{\mu})$. In Fig. 1 b) we see how the codomain of equation (3) goes from $-\infty$ to $\ln(PDF(\bar{\mu}))$. By evaluating equation (3) we no longer have to deal with the precision requirements necessary to calculate the values that are very close to 0.

B. Relative PDF

In order to deal with the numerical precision problem even further we take the logarithm of the PDF as expressed in equation (3) and do the following transformation

$$f(\vec{x}) = \exp[g(\vec{x}) - \max(g(\vec{x})) + C], \quad (4)$$

with

$$C = \ln\left(\frac{10^{38}}{d}\right), \quad (5)$$

where C is a constant added in order to shift $f(\vec{x})$ into a position where it uses all the float32 available range increasing the range where it can be represented. Besides increasing the representation range of the outputs, this transformation also makes a relative normalization of the outputs.

Evaluating equation (4) yields the plot shown in Fig. 1 c).

C. Compression of \vec{x}

As the numbers of dimensions start to increase, a squeeze can be seen in the width of the various one dimensional slices of the Gaussian distribution's probability density function. The behaviour can be seen in Fig. 1 a).

This squeeze is carried through the transformations made to the PDF, as can be seen in Fig. 1 c), and ends up giving numerical problems.

To solve the issue we can make a dimension dependent compression of the coordinates \vec{x} . To discover this change in coordinates we force the output of equation (2) to be the same as when $d = 1$ for all dimensions. Doing this yields the following change in coordinates

$$\vec{x}_{compressed} = \frac{\vec{x}}{d}. \quad (6)$$

If we look at Fig 1 c) and Fig 1 d) we can see the differences that such a change creates. At $d = 1$ there is no change as expected from equation (6). At $d = 24$ however, we can see how the coordinate change shapes $f(\vec{x})$ to avoid numerical problems.

V. THE CLUSTERING ALGORITHM

Lets explain what we mean when we say top-down hierarchical GMM-EM clustering algorithm. The top-down hierarchical part means that the algorithm assigns an hierarchy to its clusters and that it proceeds to find them starting

from the clusters at the top of the hierarchy. The GMM-EM part of the name refers to the combination of the EM (Expectation Maximization) algorithm with a Gaussian Mixed Model (GMM). In the following sections we will explain these concepts in more detail.

A. The Gaussian Mixed Model

A Gaussian mixed model assumes that, when dealing with a random collection of points, these points will follow one of k multivariate Gaussian distributions, each with a probability density function like the one shown in equation (2), with their own mean $\vec{\mu}_j$ and with their own covariance matrix Σ_j , where j goes from 1 to k .

The mixture of k Gaussian distributions themselves follow a multinomial distribution, ϕ . The probability of random point belonging to distribution j is denoted by ϕ_j .

B. The Expectation Maximization Algorithm

When applied to our problem, the EM Algorithm is an iterative algorithm with two steps [8], the E-step and the M-step. The algorithm is meant to be used on a collection of n points.

First we have the E-step, where we try to estimate the probability of the i^{th} point with position \vec{x}_i belonging to the j^{th} distribution

E-Step

$$w_j(\vec{x}_i) := \frac{f_j(\vec{x}_i)\phi_j}{\sum_{l=1}^k f_l(\vec{x}_i)\phi_l}. \quad (7)$$

Then we have the M-step, where we use the previous result to estimate the following parameters of our model

M-Step

$$\begin{aligned} \phi_j &:= \frac{1}{n} \sum_{i=1}^n w_j(\vec{x}_i), \\ \vec{\mu}_j &:= \frac{\sum_{i=1}^n w_j(\vec{x}_i) \vec{x}_i}{\sum_{i=1}^n w_j(\vec{x}_i)}, \\ \Sigma_j &:= \frac{\sum_{i=1}^n w_j(\vec{x}_i) (\vec{x}_i - \vec{\mu}_j)^T (\vec{x}_i - \vec{\mu}_j)}{\sum_{i=1}^n w_j(\vec{x}_i)}. \end{aligned}$$

The EM algorithm is supposed to be repeated until convergence. The results of the E-step are "guesses" that the M-step assumes to be true.

C. The Top-Down Hierarchical Part

The EM Algorithm introduced in the previous section requires the user to know *a priori* the value of k , the number of separable Gaussian distributions contained in our Gaussian mixed model. This poses a problem when one does not know this value beforehand. To tackle this problem a top-down hierarchical approach was taken when deploying our algorithm.

The algorithm is run recursively. On the first iteration, the algorithm will use the EM algorithms to find k clusters. A threshold is then defined on the variance of the found clusters. If the variance of any found cluster is larger than the defined

threshold, the algorithm will dive on those clusters and find k sub-clusters from the original ones.

D. The Initialization

To initialize the algorithm we will need to have initial parameters to run the EM algorithm. The value of k has to be imputed *a priori* but, as we discussed in the previous chapter, that won't impose a maximum in our number of found clusters.

The rest of the initial parameters can be crudely obtained by taking random k splits of the dataset and calculating their means and covariance matrices.

A more efficient way is run the dataset through a k-means algorithm and use our parameter k as the input for the k-means. This method proved to be the most efficient.

E. Sampling

In the various tests done to the algorithm, it was apparent that the EM algorithm would converge faster if we had larger amounts of data points. We took advantage of this by sampling our dataset and confirmed that the the EM algorithm benefited greatly from the sampling. In order to alter the original dataset as little as possible we did a "similar document" sampling. By this we mean that the created points were generated very close to the original ones. Therefore they can be interpreted as documents that have the same (or very similar) meaning to their counterparts.

VI. TESTING THE ALGORITHM

To test the algorithm we showcase its application to two toy datasets, evaluate the results and conclude on performance.

The toy datasets will consist of n clusters of 768-dimensional points, C_n , $n = 1, 2, 3...$ with each dimension following a Gaussian distribution, $\mathcal{N}(\mu, \sigma)$, where μ is the mean of the distribution and σ is its standard deviation.

A. Test A

Dataset A has 4 clusters with 768 dimensions following normal distributions $\mathcal{N}(\mu, \sigma)$:

- C_1 (**Black**): 300 points, all dimensions follow $\mathcal{N}(-3, 0.3)$
- C_2 (**Red**): 150 points, all dimensions follow $\mathcal{N}(-2.5, 0.3)$
- C_3 (**Blue**): 300 points, all dimensions follow $\mathcal{N}(-2, 0.3)$
- C_4 (**Green**): 150 points, all dimensions follow $\mathcal{N}(2, 0.7)$

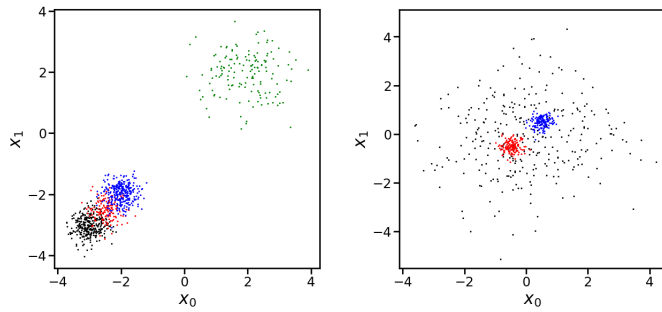
Two dimensions of the dataset can be seen in Fig. 2 a). Since all dimensions are treated equally, in Fig. 2 a) is a good representation of the dataset.

When using the algorithm, all the four clusters were completely separated

B. Test B

Dataset B has 3 clusters with 768 dimensions following normal distributions $\mathcal{N}(\mu, \sigma)$:

- C_1 (**Black**): 150 points, all dimensions follow $\mathcal{N}(0, 1.5)$
- C_2 (**Red**): 150 points, all dimensions follow $\mathcal{N}(-0.5, 0.2)$
- C_3 (**Blue**): 150 points, all dimensions follow $\mathcal{N}(0.5, 0.2)$



a) Two dimensions of dataset A. b) Two dimensions of dataset B.

Fig. 2: Results obtained after applying the clustering algorithm to the toy datasets.

Two dimensions of the dataset can be seen in in Fig. 2 b). Since all dimensions are treated equally, in Fig. 2 b) is a good representation of the dataset.

When using the algorithm, all the three clusters were completely separated

VII. DETECTING TRENDS

If the points in our dataset are BERT embeddings of abstracts from arXiv papers, then the developed algorithm can be used to detect clusters of similar papers. The properties from these clusters of papers and the way they interact with other clusters has important meaning and can be used to evaluate interactions between research fields and their topics.

If we then do this dynamically, by taking timestamped snapshots of these embeddings, we can track how these systems of clusters evolve and behave. This can be specially useful as a way to quickly and autonomously detect emerging trends in various research areas.

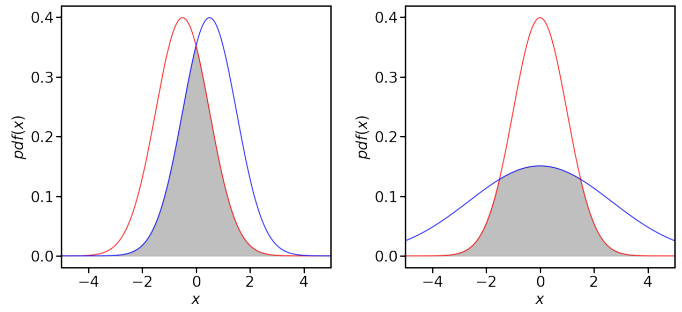
To achieve this with our algorithm, we ingest a sliding time window of dataset snapshots. More precisely, we get monthly ArXiv paper snapshots, ingest 12 months of data and use this ingestion to obtain the first snapshots of clusters. We then advance one month in our snapshot window of 12 months and repeat the process. If we keep doing this we will have the evolution of these clusters.

A. Linking Clusters

Every time we shift the time window our clusters will be updated. We can keep track of the evolution of each cluster by evaluating their similarity across time windows. To do this evaluation we make use of the Bhattacharyya coefficient, BC .

This coefficient can take values between 0 and 1 and is closely related to the measurement of the intersection of probability density functions between two distributions \mathcal{N}_1 and \mathcal{N}_2 [9]. Before computing the Bhattacharyya coefficient we must first compute the Bhattacharyya distance, D_B as such

$$D_B(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{8} (\vec{\mu}_1 - \vec{\mu}_2)^T \Sigma^{-1} (\vec{\mu}_1 - \vec{\mu}_2) + \frac{1}{2} \ln(|\Sigma|) - \frac{1}{4} \ln(|\Sigma_1|) - \frac{1}{4} \ln(|\Sigma_2|). \quad (8)$$



a) Two Gaussian distributions with $\mu_1 = -0.5, \sigma_1 = 1, \mu_2 = 0.5, \sigma_2 = 1. D_B = 0.883$. b) Two Gaussian distributions with $\mu_1 = 0, \sigma_1 = 1, \mu_2 = 0, \sigma_2 = 7. D_B = 0.813$.

Fig. 3: The distributions on the right plot have the same mean but the ones on the left plot are more similar since they have a higher D_B .

We can then compute $BC(\mathcal{N}_1, \mathcal{N}_2)$

$$BC(\mathcal{N}_1, \mathcal{N}_2) = e^{-D_B(\mathcal{N}_1, \mathcal{N}_2)}. \quad (9)$$

Using this coefficient yields a better measure of similarity between two Gaussian distributions than just using the difference between their means as can be seen in Fig 3.

VIII. OBTAINED RESULTS

A. Weak Signal Detection

The clusters obtained from the computer science category of the ArXiv dataset can be visualized in Fig. 4 where we have highlighted a cluster in red for further analysis. In this figure the x -axis is the number of documents contained in a certain cluster in a certain time window divided by the number of total documents in that time window. This quantity is referred to as average cluster frequency (ACF).

The y -axis is the increasing rate of these ACF value for all periods as a geometric mean (y -axis).

In this study we adopt the approach proposed by Yoon [10] to detect weak signals and differentiate them from the strong ones. From this quantitative point of view, the determination of weak signals follows the rationale that clusters with a low occurrence frequency but a high increasing rate can be classified as weak signals and clusters exhibiting a high occurrence frequency and high increasing rate as strong signals. This said, the weak signals are placed in the top-left quadrant of the cluster emergence map. This approach leads us to conclude that the highlighted cluster on Fig. 4 is very likely to be a weak signal.

To interpret the topics encoded in the clusters shown in the emergence map we can show the most relevant keyword in a frequency based word cloud. We do this for our highlighted cluster in Fig. 5 a).

If we look at Fig. 5 a) we can see a bag of words with the most relevant keywords for the topic represented by our isolated cluster. Having a look at the relevant keywords associated with this cluster it is noticed that the its associated topic revolves around an emerging wireless technology. This

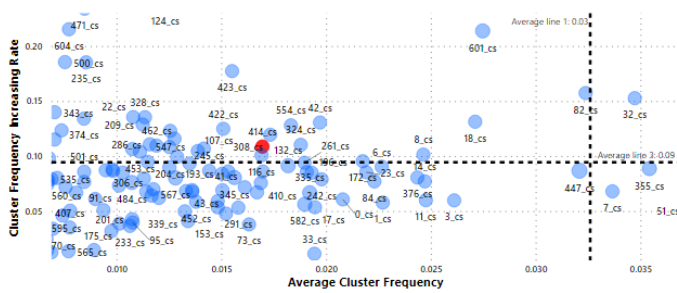
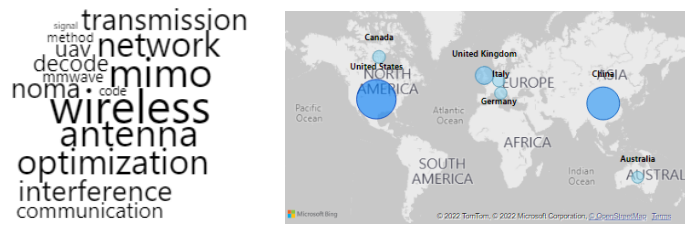


Fig. 4: Cluster emergence map. The red cluster is isolated for further analysis. The topic this cluster represents can be seen in the bag of words in Fig. 5 a).



a) Cluster bag of words for isolated cluster. b) Comparison of amount of papers published on the topic of the isolated cluster.

Fig. 5: On the left we have a bag of words helping us understand the topic from the highlighted cluster in Fig. 4. On the right we have a comparative visualization of the amount of paper published on this topic for each country.

is also closely related with 5G communication systems, where MIMO (Multiple Input Multiple Output) antenna is one of its key enabling techniques.

B. Leading Research on Weak Signal

We can also see the amount of papers published in the topic represented by our highlighted cluster for various countries in Fig. 5 b).

From Fig. 5 b) we can see that the US and China are leading the world when it comes to this topic in particular. This result appears to be valid since researchers from US and China are focusing their research more and more on 5G wireless technologies [11].

IX. DISCUSSION

We believe our method of clustering ArXiv articles can reveal indicators of weak signals in scientific articles. It can reveal the countries, companies and academia that are leading in these topics and it can be used to study the evolution of these weak signals by tracking their corresponding clusters through time.

As was demonstrated, the algorithm is capable of working with highly dimensional datasets ($d \geq 768$). This ability was a hard requirement for us to work with BERT embeddings. This possibility is not available when working with most clustering algorithms such as LDA or a more traditional implementation of the GMM algorithm such as the one implemented in *sci-kit*

learn [12]. These algorithms are bound to face numerical problems at these regimes and first require a drastic dimensional reduction of the data (using a PCA algorithm for example) in order to run. The resulting dimensionally reduced dataset will only partially include the information contained in the original high dimensional data thus, reducing the accuracy of any algorithm used to ingest it.

The algorithm also benefits from its hierarchical structure. This structure allows us to detect any number of clusters, *i.e.*, we don't need to set a predefined number of clusters before running the algorithm. This is of great importance for our use case since we can not know *a priori* the amount of weak signals that can emerge.

Our method brings an approach that looks at the written abstracts instead of added keywords, references and received citations. It does this to pick up on weak signals that may translate into the beginning of a technological leap.

REFERENCES

- [1] Eulaerts, O., Joanny, G., Girdali, J., Fragkiskos, S., Perani S.(2019). Weak signals in Science and Technologies
- [2] Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.-J. (paul), & Wang, K. (2015). An overview of Microsoft academic service (MAS) and applications. Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion.
- [3] Kok A, Wingerden A, Ndoni A, Street MD. (2022). Carbon-dating Articles with Transformer Series.
- [4] Clement, C. B., Bierbaum, M., O’Keeffe, K. P., & Alemi, A. A. (2019). On the Use of ArXiv as a Dataset. <https://arxiv.org/abs/1905.00075>
- [5] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional Transformers for language understanding. <https://arxiv.org/abs/1810.04805>
- [6] Multivariate normal distribution. In Wikipedia. https://en.wikipedia.org/wiki/Multivariate_normal_distribution
- [7] Normal distribution. In Wikipedia. https://en.wikipedia.org/wiki/Normal_distribution
- [8] Ng, Andrew. "CS229 Lecture notes", Lecture 11, 3 of May of 2021, Stanford University
- [9] Bhattacharyya distance. In Wikipedia. https://en.wikipedia.org/wiki/Bhattacharyya_distance
- [10] Yoon, J. (2012). Detecting weak signals for long-term business opportunities using text mining of Web news. <https://doi.org/10.1016/j.eswa.2012.04.059>
- [11] Techsciresearch. <https://www.techsciresearch.com/blog/top-ten-countries-with-largest-5g-network/263.html>
- [12] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.