

Survey on Model Monitoring in NLP

Yash Jain¹, Sidhant Khamankar², Husain Fatepurwala³, Suruchi Dedgaonkar⁴, Dr. Priya Shelke⁵
and Riddhi Mirajkar⁶

¹⁻⁶Vishwakarma Institute of Information Technology, Pune, India
Email: yash.21910699@viit.ac.in, {sidhant.21910598, husain.21910817,
suruchi.dedgaonkar, priya.shelke, riddhi.mirajkar }@viit.ac.in

Abstract— Model Monitoring is an operational stage which comes after model deployment in machine learning lifecycle. It includes monitoring ML models for changes like model performance degradation, resultant data drift, and concept drift, and ensuring that the developed model is maintaining an optimum level of performance. ML teams can use model performance metrics like accuracy, precision, recall to monitor the real time or live performance of production models. However, these metrics need ‘ground truth’ or labels for these real-time predictions. Although labels are always available in a training data set, they might not always be available in production for a given use case. In the absence of inputs or to complement the visibility of the performance metrics, monitoring live changes in production features and prediction distributions can be used as an indicator and troubleshooting tool for issues in performance. Proper model monitoring can help improve model performance, increase transparency, and ensure that models are deployed ethically and responsibly. We provide an overview of the importance of model monitoring in NLP and highlight some of the key techniques and tools used for this purpose. The main aim of this research will be to understand the capabilities of novel drift algorithms and how they can be used to automate the ML-OPS pipeline used in the industries. Monitoring NLP Models can save a lot of time and effort of data scientists. It will also help organizations in keeping their large language models to be more consumer friendly and ethical.

Index Terms— nlp, model, monitoring

I. INTRODUCTION

Natural Language Processing (NLP) is a one of the branch in artificial intelligence which focuses on interaction between human language and computers. It involves using algorithms and computational techniques to enable machines to interpret, understand, and generate natural language. NLP is very popular in today's era and it has a wide range of uses and applications, including machine language translation, sentiment analysis, speech recognition, chatbots, and text summarization. It is used in various types of industries such as finance, healthcare, customer service, and marketing to automate tasks and improve human efficiency. NLP techniques can be divided into several categories, including rule-based systems, deep learning models and statistical models. Rule-based systems use sets of linguistic rules to analyze and generate natural language, while statistical models use probabilistic models to analyze language patterns. Deep learning models, such as neural networks, use large amounts of data to learn patterns and generate accurate outputs. NLP faces several challenges, including language ambiguity, context-dependent meaning, and the need to process large amounts of unstructured data. However, advances in machine learning and computational power have led to significant improvements in NLP performance, and it continues to be an area of active research and development.

The most important part of any machine learning model is the dataset it is trained on. The model can perform only as well as the data which is fed to it. If the production dataset is not in alignment with the training dataset used, the results may not be as expected. To save a lot of time for researchers and developers of such models, this data drift or concept drift can be detected beforehand. For this, we use model monitoring techniques. So basically, model monitoring is the concept of tracking or observing the performance of a machine learning model over time to ensure that it continues to produce accurate and reliable results. Model monitoring is essential for detecting and diagnosing performance issues, identifying drifts or changes in the data, and updating or retraining the model as needed. It typically involves tracking various metrics related to the model's output like model performance, such as precision, accuracy, F1 score, recall, and AUC-ROC. These metrics can be monitored on a regular basis, such as daily or weekly, to ensure that the model is operating within acceptable thresholds.

For structured data it is very easy to calculate these metrics and monitor the model, but in the case of unstructured data which is used in the case of NLP models, these metrics are not so easy to find. Different approaches like making word embeddings and applying statistical tests like Jensen-Shannon test and Adversarial Validation. A wide range of application cases, including text summarization, chatbots, virtual assistants, and machine translation, all benefit from natural language understanding. It's crucial that the data utilised in training and production scenarios originates from the same distribution in order to ensure that these apps work as intended. Data drift is a phenomena that occurs when the data used for inference, or production utilised data, changes from the data used during model training. When this happens, the model is no longer applicable to the use case and the data in production and, if used further, performs worse than anticipated. It's crucial to regularly examine the inference data and contrast it with the training data.

To detect drift accurately and efficiently in NLP, word embeddings are used, which are low-dimensional text representations. Embeddings can be obtained using various language models such as BERT and Word2Vec where BERT is a transformer-based models. Bidirectional Encoder Representations from Transformers (BERT) comes from the family of masked-language models which was published in 2018 by Kristina Toutanova, Kenton Lee, Ming-Wei Chang, and Jacob Devlin. The primary goal of these models is to preserve the semantic information from the input text while also projecting high-dimensional data onto low-dimensional areas. The outcomes are in the form of dense vectors with contextual meaning that may be utilised for a variety of downstream activities, such as monitoring for data drift.

Model monitoring has many different applications in natural language processing. Key areas in NLP where model monitoring is important include: Data analysis for tracking input and output data can assist in identifying data quality problems, such as missing or incorrectly labelled data, which can adversely affect model performance. NLP models are susceptible to bias, which can result in unfair outcomes and discrimination. By analysing data and model outputs, model monitoring can assist in detecting and mitigating this bias. NLP models may have ethical ramifications like as invasions of privacy, the perpetuation of stereotypes, and the influencing of public opinion. Ensuring ethical considerations are taken into account is possible with monitoring models.

II. LITERATURE SURVEY

There are a variety of methods explored by researchers for the creation of word embeddings from text data. Arthur in his paper [1] and Luke Vilnis in his paper about Gaussian Embeddings [2] have discussed how probability densities can be calculated for embedding words. Mapping of each embedded word to a density provides many advantages instead of mapping each embedded word to a point vector in low-dimensional space. These advantages have been discussed in their paper about Gaussian Embeddings where they have presented a method for learning different representations in the space of Gaussian distributions. Arthur's paper on 'Embedding Words as Distributions with a Bayesian Skip-Gram Model' introduces a Bayesian model for generating word embeddings and has compared it to Gaussian distribution. The edge of their proposed method is that they are able to capture context-based densities as well which has many potential applications.

In [3], Chunting Zhou has proposed a new approach for matching two densities by a method known as normalizing flow. Here the monolingual probability densities are calculated through a GMM (Gaussian Mixture Model) on their respective embedding spaces. A future scope to his paper can be to integrate the works of Luke Vilnis' on Gaussian embeddings with their approach. The normalizing flow technique used in this can be used as a density model which we will be using in our experiments. Various authors like David Evholt [4] and Sarvagya Srivastava [5] have tried Generative Adversarial Networks (GANs) for economic forecasting. In David's paper, he has used an NLP model DistilBERT on a Twitter dataset. This can be considered to be a stage setter for NLP-GAN models. A good amount of work has already been done on concept drift and data drift detection as well. Firas Bayram [6]

in his paper has reviewed different types of terms used for concept drift. In their paper they have also reviewed and classified different concept drift detection techniques on the basis of their performance.

Robert Feldhans in his paper [7] has proposed another element to this by considering the data input. He has experimented on unstructured data which is used for NLP models. In his experimentation, they have used multiple data drift detector algorithms and two state of the art NLP models. From their experimentation, they concluded that the LSDD (Least-Squares Density Difference) and KTS (Kernel Two-Sample) drift detectors give superior performance as compared to other drift detectors. Also, regarding embedding dimensions, lower numbers of dimensions tend to produce better results from the drift detectors. The Table 1 below shows comparative analysis of different papers which were reviewed.

TABLE I. COMPARISON OF DIFFERENT PAPERS

Sr No	Title	Abstract	Inference
1	Word representations via Gaussian Embeddings	Instead of point vectors use density distributions.	Density distributions give a better performance over word to point mapping
2	Embedding Words as Distributions with a Bayesian Skip-gram Model	Used a Bayesian model for generating word embeddings.	Bayesian gave better results when compared to Gaussian distributions.
3	Density Matching for Bilingual Word Embedding	Used a GMM for monolingual probability density.	GMM can be another alternative for probability densities. Other methods could be integrated with it as well.
4	From concept drift to model degradation: An overview on performance-aware drift detectors	Review and classification of concept drift detection techniques based on their performance.	Concept Drift can be detected through many techniques but it does not take data input into consideration.
5	Drift Detection in Text data with Document Embeddings	Experimented with different drift detectors on state of the art NLP models.	Adds the missing element of data input to detect data drift on NLP models.

A. Amazon SageMaker Model Monitor

One may use Amazon SageMaker Model Monitor [8] to continually and in real-time check the efficacy of machine learning models. When any drift in model performance happens, Model Monitor may be used to set up alerts that will warn users and initiate appropriate actions. Without having to do any manual intervention or build extra tools and technologies, proactive and early identification of these deviations enables us to take remedial steps, such as gathering fresh training data based on ground truth, reviewing upstream systems, and retraining models.

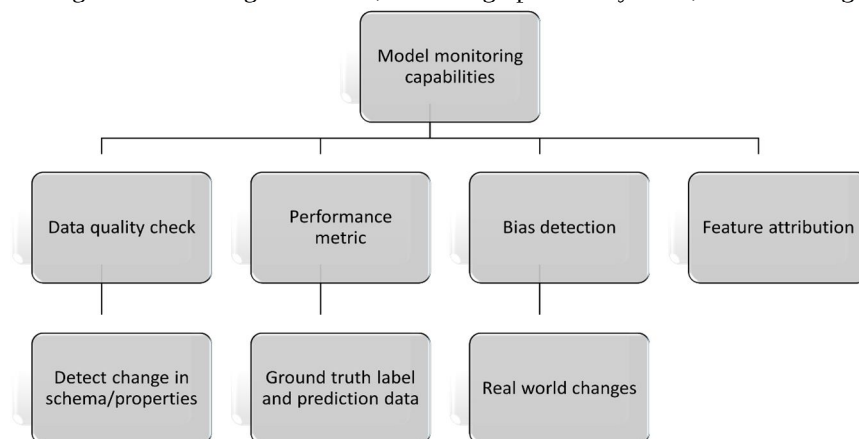


Figure 1. Types of model monitoring capabilities

As shown in the figure 1, to prevent and identify model data drift in live and real-time settings, Model Monitor has four basic types of monitoring capabilities:

- To find changes in independent variable attributes and schemas, a data quality check is available.
- Model performance metric is computed using ground truth labels collected from user applications which is in turn fed or merged into prediction data to compute metric. This can be used to monitor performance characteristics like precision or accuracy.
- Bias detection can be done with the help of Amazon SageMaker Clarify. This improves visibility of potential bias. Bias is generated into the model due to the changes taking place in the world.
- Alert is raised when a change occurs in feature attributions.

B. Amazon Methodology –

To detect drift accurately and efficiently in NLP, word embeddings are used, which are low-dimensional text representations. Embeddings can be obtained using various language models such as BERT and Word2Vec where BERT is a transformer-based models. Bidirectional Encoder Representations from Transformers (BERT) comes from the family of masked-language models which was published in 2018 by Kristina Toutanova, Kenton Lee, Ming-Wei Chang, and Jacob Devlin. These models main aim is to project high-dimensional data into low-dimensional spaces along with preserving the semantic information from the input text. The results are in the form of contextually meaningful and dense vectors, which can be used for other use cases like in various downstream tasks which includes monitoring for data drift.

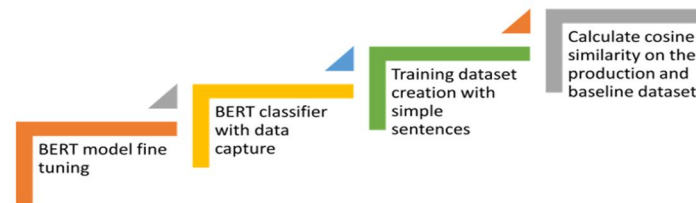


Figure 2. Steps for drift detection

Embeddings is used in detecting covariate shift of English sentences. Model Monitor is utilized to facilitate continuous monitoring for a text classifier which is deployed on a production environment. Our approach consists of the following steps as shown in the figure 2:

- A BERT model is fine-tuned using SageMaker.
- A BERT classifier with data capture is deployed as a real-time endpoint.
- A baseline dataset is created that consists of sample sentences which is used in training the BERT Classifier.
- A custom SageMaker monitoring job is created to calculate cosine similarity for the data present in the production and in the baseline dataset.

Model Monitor provides the following types of approaches for monitoring:

- Monitoring drift in data quality.
- Monitoring drift in model quality metrics like accuracy.
- Monitoring bias drift present in Models in Production.
- Monitoring Feature Attribution Drift present in Models in Production

C. Monitoring data drift with Azure + Data drifts

Azure Machine Learning [9] [10] makes it easy to detect drift by computing a single metric. It abstracts the complexity related to the datasets which are being compared. These datasets may have large number of features along with large number of rows. Drill down into feature which is causing the drift is possible. Feature level metrics can be inspected to debug and find out the root cause for occurrence of drift.

This approach simplifies monitoring data instead of traditional rules-based techniques. Rules-based techniques includes allowed unique values or allowed data range. These can be error prone and time consuming.

D. With Azure ML dataset monitors, we can:

- Understand how the data changes over time by analyzing drift in data.
- Collect data from deployed model and monitor differences between serving and training dataset.
- Monitor differences between target and baseline dataset.
- Track changes of statistical properties by profiling features in data.

- Create early warning alerts for probable issues due to data drift.
- A new version for dataset can be created when it is determined that data has drifted too much.

E. For Dataset monitoring following procedure is followed:

- Whenever new data is present is added, detect and alert on data drift.
- Historical data analysis
- New data is profiled.

Data drift algorithm is helpful in providing overall measure of change. It indicates features which are responsible for this and requires further investigation. In time series dataset, monitors produces other types of metrics by profiling new data.

Custom alerting can be set up on all metrics generated by the monitor through Azure Application Insights. Dataset monitors can be used to quickly catch data issues and reduce the time to debug the issue by identifying likely causes.

F. Creating a data drift monitor:

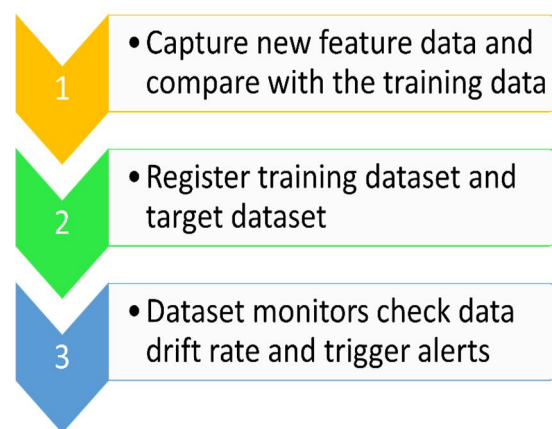


Figure 3. Creating data drift monitor

As shown in the figure 3, Creating a data drift monitor involves following steps

- One can capture new feature data in a dataset and compare it to the training dataset based on which model was trained.
- Two datasets need to be registered
 - Training data or the baseline dataset.
 - A target dataset which will be used for comparison with baseline at different time intervals. A feature wise column is required and another column to record the timestamp. This helps in calculating rate of data drift.
- Dataset monitors can be defined to trigger alerts whenever the rate of data drift is more than the set threshold. This task can be accomplished with the of visual interface of Azure ML Studio or by Azure ML SDK.

G. Scheduling alerts:

Data drift monitoring works by running a comparison at scheduled frequency, and calculating data drift metrics for the features in the dataset that we want to monitor. One can define a schedule to run every Day, Week, or Month. Data drift is measured using a calculated magnitude of change in the statistical distribution of feature values over time.

H. Metrics:

- Numeric features
- To transform the baseline distribution to target, Wasserstein distance is the minimum amount of work which is required.
- Average or Mean value for the feature.
- Minimum value for the feature.
- Maximum value for the feature.

I. Categorical features:

- Euclidean distance – It is computed for categorical columns. It is computed on two vectors which are generated from two datasets with same categorical column over empirical distribution. The column is drifted when it deviates from 0. Time series plot over this metric can reveal drifting feature.
- Unique values or cardinality in terms of number of the feature.

III. PROPOSED SYSTEM

Using two common datasets, we will examine sophisticated pre-trained models such as GloVe and BERT. The textual data will be processed and transformed into word embeddings and density models, and the effectiveness of the two methods will be contrasted. To guarantee that our results are trustworthy and consistent, we shall set clear guidelines and standards for the trials.

We conducted our experiment using two datasets: the Twitter Election Dataset and the Amazon Product Review. In the Datasets section below, comprehensive details on these datasets are provided. First, text data is preprocessed. For the Twitter dataset, the pre-processing processes include removing unclear and non-English messages. Since there is no non-English text in the Amazon Review dataset, this step is not necessary. The removal of hashtags, URLs, stop words, and punctuation is then necessary. Tokenization is the next action to take. Here, all the words in the text data are divided into tokens. Different methods can be used to create these tokens. In our use case, word tokenization will be used to divide the text into segments, with whitespace serving as the delimiter. Punctuation has already been eliminated during pre-processing, making it simple to tokenize words using simply whitespace. A final corpus created by tokenization can be fed to embedding models. Word embeddings are a vector representation of the corpus created by embedding models. For this, various embedding model types that are detailed in the following sections can be employed. We immediately apply the cosine similarity measure to the embeddings for semantic similarity derived from word embeddings. We can apply a classifier on the embeddings to determine whether the distribution of training and production data differs.

A. Datasets

Amazon Product Reviews: Over 30,000 customer reviews of products that were offered for sale on the Amazon website between January 1 and March 31, 2020 are included in the Amazon Product Review dataset. We will combine user reviews with their text summaries to detect drift. We will also assign ratings of minus one to one for negative, neutral, and positive reviews. The importance of this score being closely related to the texts the user contributed and reflecting the tone of those writings must be emphasised. We commonly utilise the review ratings as classes for retraining the models.

Twitter Election Datasets: Tweets that mention either #Biden or #Trump in relation to the two candidates for president of the United States in 2020 are included in the Twitter Election dataset. With $t_{min} = 2020-10-14$ to $t_{max} = 2020-11-07$, all of the offered data points were produced in the last three weeks before the election. The original dataset contains about 1.7 million distinct data points, of which 777,00 use the hashtag #Biden and 971,00 use #Trump. The two classes in this dataset are often divided into #Biden and #Trump. All ambiguous tweets including both the #Biden and #Trump hashtags have been removed as a result.

We presented and contrasted various methods for identifying drift in text data. We employed both a common method known as adversarial validation and a commercial method that computes the cosine similarity of embeddings. In order to synthesise semantically drifted data, we have also created an algorithm that injects synonyms and other terms into text data. Along with the drift metric, the use of embeddings, particularly BERT and GLOVE embeddings, was emphasised as a potent technique for detecting drift.

IV. CONCLUSION

To summarize, this literature review focused on various methods and techniques for model monitoring in natural language processing (NLP) to detect concept drift and data drift. The use of embeddings, especially BERT embeddings, was highlighted as a powerful tool for detecting drift. Additionally, the use of density models, such as Gaussian distributions and Generative Adversarial Networks (GANs), was explored to better capture the context-based densities of words. Continuous monitoring of machine learning (ML) models deployed in a production environment is critical to detect and mitigate any drift in model performance. With the use of Amazon SageMaker Model Monitor, we can monitor data quality, model bias, model quality, and model explainability to proactively identify any potential issues and take corrective actions. Embeddings obtained from language models like BERT can be utilized to monitor for data drift in NLP tasks. Similarly, Azure Machine Learning dataset

monitors can be used to alert and detect data drift on newer data in a dataset, profile new data over time, and analyze historical data for drift. The overall goal is to ensure that ML models continue to perform accurately and reliably over time, thereby improving their overall effectiveness and impact.

REFERENCES

- [1] T. Arthur Bražinskas, Serhii Havrylov, “Embedding words as distributions with a bayesian skip-gram model,” ACL Anthology, vol. Proceedings of the 27th International Conference on Computational Linguistics, 2018. [Online]. Available: <https://aclanthology.org/C18-1151>
- [2] L. Vilnis and A. McCallum, “Word representations via gaussian embedding,” 12 2014. [Online]. Available: <http://arxiv.org/abs/1412.6623v4>
- [3] C. Zhou, X. Ma, D. Wang, and G. Neubig, “Density matching for bilingual word embedding,” in Proceedings of the 2019 Conference of the North. Association for Computational Linguistics, 2019. [Online]. Available: <http://dx.doi.org/10.18653/v1/n19-1161>
- [4] O. L. DAVID EVHOLT, Generative Adversarial Networks and Natural Language Processing for Macroeconomic Forecasting, 2020. [Online]. Available: <https://www.divaportal.org/smash/get/diva2:1431649/FULLTEXT02.pdf>
- [5] R. V. Sarvagya Srivastava, Vishwaas Khare, “Economic forecasting using generative adversarial networks,” INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT), vol. 10, 2021. [Online]. Available: <https://www.ijert.org/economic-forecastingusing-generative-adversarial-networks>
- [6] F. Bayram, B. S. Ahmed, and A. Kassler, “From concept drift to model degradation: An overview on performance-aware drift detectors,” Knowledge-Based Systems, vol. 245, p. 108632, 6 2022. [Online]. Available: <http://dx.doi.org/10.1016/j.knosys.2022.108632>
- [7] S. H. M. H. S. Robert Feldhans, Adrian Wilke, “Drift detection in text data with document embeddings,” Springer, vol. 13113, 2021.
- [8] T. C. Vikram Elango and R. Ramesha. (2022) Detect nlp data drift using custom amazon sagemaker model monitor. [Online]. Available: <https://aws.amazon.com/blogs/machine-learning/detect-nlp-data-drift-using-custom-amazon-sagemaker-model-monitor/>
- [9] Microsoft. (2023) Detect data drift on datasets (preview) - azure machine learning. [Online]. Available: <https://learn.microsoft.com/en-us/azure/machine-learning/v1/how-to-monitor-datasets?tabs=python&view=azureml-api-1>
- [10] M. Training. (2023) Monitor data drift with azure machine learning. [Online]. Available: <https://learn.microsoft.com/en-us/training/modules/monitor-data-drift-with-azure-machine-learning>