



University of Manouba  
National School of Computer Sciences



## REPORT OF THE DESIGN AND DEVELOPMENT PROJECT

---

Subject: Automated Zoosanitary Surveillance  
System in Tunisia

---

*Authors :*

Mrs. Eya METHNANI

Mr. Ghaissen SEBAI

*Supervisors :*

Pr. Anja HABACHA

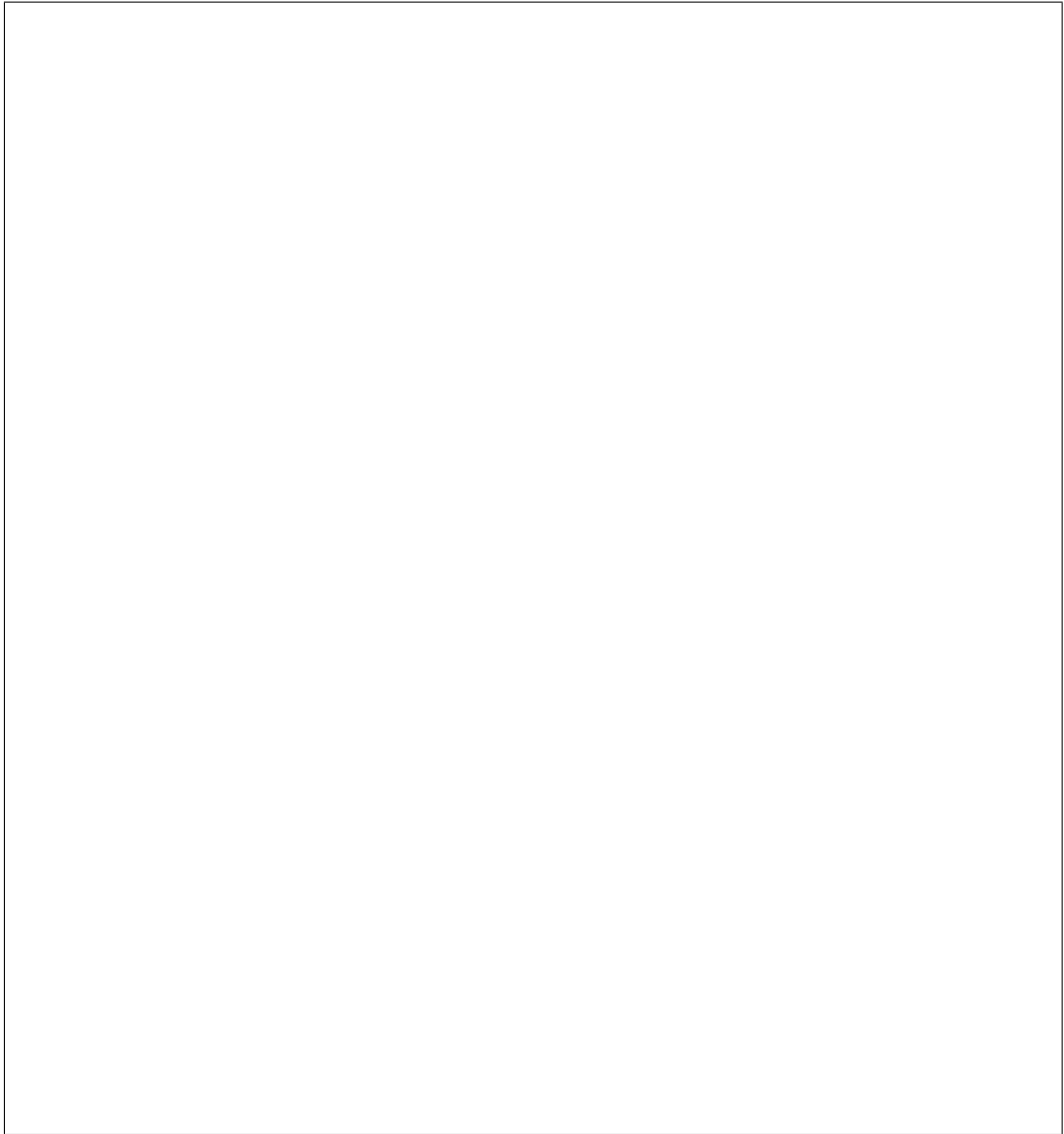
Dr. Ferihane KBOUBI



---

Academic Year : 2023 /2024

## Appréciations et signature de l'encadrant

A large, empty rectangular box with a thin black border, intended for the appreciations and signature of the supervisor.

# Acknowledgement

We are deeply grateful and would like to express our sincere appreciation to everyone who supported us during this memorable journey.

Firstly, we extend our profound gratitude to our supervisors, Professor Anja Habacha and Dr. Ferihane Kboubi, for their invaluable guidance and advice. We are also thankful to CNVZ for proposing this project, and to Wael Seddik, an ENSI alumnus, for his continuous support and mentorship.

Additionally, we appreciate the esteemed jury members who honored us by reviewing this work. Lastly, we thank everyone who offered their help, advice, or encouragement.

# Contents

<b>Introduction</b>	<b>1</b>
<b>1 State of the Art</b>	<b>3</b>
1.1 Description of functionalities . . . . .	3
1.1.1 Actor identification . . . . .	3
1.1.2 Functional requirements . . . . .	3
1.1.3 Non-Functional requirements . . . . .	3
1.2 Natural Language Processing Techniques . . . . .	4
1.2.1 Natural Language Processing Applications . . . . .	4
1.2.2 Classification models . . . . .	5
1.3 Large Language Model . . . . .	5
1.4 Related Work . . . . .	6
1.5 Conclusion . . . . .	9
<b>2 Methodology</b>	<b>10</b>
2.1 Proposed Solution . . . . .	10
2.2 Data Description and Collection . . . . .	11
2.3 Data exploration . . . . .	13
2.4 Preprocessing and Feature Extraction . . . . .	16
2.5 Conclusion . . . . .	18
<b>3 Approach and Results</b>	<b>19</b>
3.1 Machine Learning Models Training and Evaluation . . . . .	19
3.1.1 Machine Learning Model Training: . . . . .	19
3.1.2 Classification Models . . . . .	20
3.2 Large Language Model (BERT) . . . . .	20
3.2.1 Environment Setup . . . . .	20
3.2.2 Multilingual BERT Implementations . . . . .	21
3.3 Comparison ML Models vs. LLM Models . . . . .	23
<b>4 Conclusions and Perspectives</b>	<b>25</b>



# List of Figures

1.1	Taxonomy of event prediction problems and techniques. . . . .	6
1.2	Distribution of ML methods for health-related text classification by social media based surveillance systems. . . . .	7
1.3	Illustration of event detection and evolution from multi-lingual text streams.	7
1.4	AI techniques in Internet-based global epidemic monitoring. . . . .	8
1.5	Word cloud of statistical and machine learning methods discovered in review.	8
2.1	Proposed Architecture . . . . .	11
2.2	CNVZ articles . . . . .	11
2.3	Scrapped Articles . . . . .	12
2.4	Char Word count . . . . .	13
2.5	Average word length . . . . .	14
2.6	Polarity and Subjectivity Distribution . . . . .	14
2.7	POS Tagging distribution . . . . .	15
2.8	POS Tagging distribution . . . . .	16
3.1	Confusion Matrix . . . . .	23

# List of Tables

1.1	Articles Analysis . . . . .	9
2.1	Comparison of Text Before and After Preprocessing . . . . .	17
3.1	Models comparaison table . . . . .	24

# Liste des sigles et acronymes

<b>CNVZ</b>	<i>"Centre national de veille zoosanitaire "</i>
<b>SVM</b>	<i>Support Vector Machine</i>
<b>LLM</b>	<i>Large Language Models</i>
<b>NER</b>	<i>Named Entity Recognition</i>
<b>NLP</b>	<i>Natural language processing</i>
<b>LDA)</b>	<i>Latent Dirichlet Allocation</i>
<b>MLEM)</b>	<i>Multi-Lingual Event Mining</i>
<b>ML</b>	<i>Machine Learning</i>
<b>CNN</b>	<i>Convolutional Neural Networks</i>
<b>RNN</b>	<i>Recurrent Neural Networks</i>
<b>AI</b>	<i>Artificial intelligence</i>
<b>TF-IDF</b>	<i>Term Frequency-Inverse Document Frequency</i>
<b>EDA</b>	<i>Exploratory Data Analysis</i>
<b>HTML</b>	<i>HyperText Markup Language</i>
<b>PROPN</b>	<i>proper nouns</i>
<b>NOUN</b>	<i>common nouns</i>
<b>PUNCT</b>	<i>punctuation</i>
<b>VERB</b>	<i>verbs</i>
<b>SVC</b>	<i>Support Vector Classifie</i>
<b>BERT</b>	<i>Bidirectional Encoder Representations from Transformers</i>



POS     *Part-of-Speech*

GPU     *Graphics processing unit*

# General Introduction

In light of the rising health risks affecting animals and humans, Tunisia is leading the way, in incorporating technologies into its animal health surveillance systems. This joint project, driven by the collaboration between the National Institute of Veterinary Research and the National school of computer science, aims to enhance the detection of zoonotic diseases through the use of cutting-edge technology.

The main goal of this project is to create an automated system that utilizes learning and Large Language Models (LLM) to analyse data gathered from various sources such, as news outlets and social media platforms. This approach is designed to detect the spread of zoonotic diseases, providing early warnings and enabling timely interventions.

## **Context and Problem Definition**

Zoosanitary vigilance plays a crucial role in keeping animals diseases under control, thus affecting zoosanitary health and agricultural economics. In the traditional scheme, surveillance is typically procedurally reporting-based and manual; it is thus slow, lacking in its accuracy. As a result, the response time during an epidemic can make a difference. These factors highlight the need for a solution that automates the process and is able to do so with certainty. This project aims to develop a system that can automatically detect the spread of epidemic time, which ultimately will assist in the prevention of diseases.

## **Project Framework**

This project is part of a broader initiative under the guidance of Tunisia's National Institute of Veterinary Research, supported by the National School of Computer Science. It also aligns with the global movement towards digital transformation in healthcare and veterinary services, adopting a technology-driven approach to traditional practices.

**Goals and Objectives of the Project:** The overarching goal of this project is to develop a technological solution that automates the detection of zoosanitary events. Specific objectives include:

- The deployment of a deep learning model capable of processing and analyzing large datasets to identify potential zoonotic threats.

- The creation of a robust Large Language Model that can contextualize and prioritize information based on its relevance to public health and veterinary concerns.

Our report consists of three chapters: The first chapter, "State of the Art": will delve into the assets utilized to comprehend our project and the tools employed. The second chapter, "Methodology": will elucidate the functionalities of our project and detail the data preparation procedures. The third chapter, "Modeling":will showcase the implemented models and their evaluations.

In summary, this project aims to revolutionize how zoonotic events are detected and managed in Tunisia, using advanced technologies to safeguard public health and animal welfare. By automating the detection process, the project not only increases the speed and accuracy of responses but also supports the global efforts in managing and preventing zoonotic diseases.

# Chapter 1

## State of the Art

In this chapter, we will dive into a comprehensive overview of various methodologies and models employed in the event prediction and AI-enabled data analysis, particularly focused on the processing and interpretation of text data.

### 1.1 Description of functionalities

#### 1.1.1 Actor identification

An actor corresponds to a role that interacts directly with the system. The main actors in our system are: CNVZ employees.

#### 1.1.2 Functional requirements

The model help CNVZ in:

- Predicting the event.
- Acting fast to prevent the spread of the disease.

#### 1.1.3 Non-Functional requirements

- Performance: The system is designed to achieve high accuracy in information classification Usability.
- Scalability: Capable of adapting to an increase in data volume and the need for expanded functionalities.

## 1.2 Natural Language Processing Techniques

### 1.2.1 Natural Language Processing Applications

Based on [URL, d] natural language processing, or NLP, combines computational linguistics—rule-based modeling of human language—with statistical and machine learning models to enable computers and digital devices to recognize, understand and generate text and speech.

Here, we explore the top seven techniques NLP employs to derive valuable data from text.

- *Sentiment Analysis:* This technique is important in gathering public opinion, market research, and customer service by analyzing emotions in the text. Sentiment analysis algorithms classify the polarity of a text at various levels.
- *Named Entity Recognition (NER):* NER is instrumental in information retrieval, organizing content, and data structuring. It identifies and categorizes key elements in text into predefined groups like names of people, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.
- *Summarization:* Automatic summarization tools condense large volumes of information into concise summaries, preserving crucial information and the original message's intent. This is especially useful for digesting news articles, long documents, and when quick content appraisal is needed.
- *Topic Modelling:* Topic modelling discovers abstract themes within a document collection. Techniques like Latent Dirichlet Allocation (LDA) often used to uncover hidden topical patterns that are present across a text corpus, enabling better content categorization and browsing.
- *Text Classification:* Text classification is a core task in NLP, where a body of text is assigned to one or more categories. It can be used for spam detection, language detection, genre classification, or tagging content as per predefined topics.
- *Keyword Extraction:* Keyword extraction (also known as term extraction or keyword analysis) involves automatically selecting significant phrases that best describe the subject of a document. It's a crucial technique for indexing data and generating tag clouds.
- *Lemmatization and Stemming:* Both techniques aim to reduce a word to its base or root form, but they differ in approach. Lemmatization considers the word's proper usage according to its morphological analysis, aiming to find the base dictionary form of a word, known as the lemma. Stemming, on the other hand, is a cruder heuristic process that chops off the ends of words in the hope of achieving this goal correctly

most of the time, which is useful for consolidating different grammatical variations of a word.

In conclusion, these seven NLP techniques serve as powerful tools for transforming text into structured, actionable data. As the digital universe expands, the role of NLP becomes ever more vital in processing and understanding the world's rapidly growing textual information.

### 1.2.2 Classification models

- **Random Forest Classifier** According to [URL, e], random forest is a commonly-used machine learning algorithm, trademarked by Leo Breiman and Adele Cutler, that combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.
- **SVM Classifier** According to [URL, f], a support vector machine (SVM) is a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane that maximizes the distance between each class in an N-dimensional space.
- **Logistic Regression** According to [URL, b] Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given data set of independent variables.

## 1.3 Large Language Model

Based on [URL, a] a large language model (LLM) definition is a type of machine learning (ML) model that can perform a variety of natural language processing (NLP) tasks, such as generating and classifying text, answering questions in a conversational manner, and translating text from one language to another. The label "large" refers to the number of values (parameters) the language model can change autonomously as it learns. Some of the most successful LLMs have hundreds of billions of parameters. LLMs are trained with immense amounts of data and use self-supervised learning (SSL) to predict the next token in a sentence, given the surrounding context. The process is repeated until the model reaches an acceptable level of accuracy.

### **Bert-base-multilingual-cased**

As defined in [URL, c] BERT is a transformers model pretrained on a large corpus of multilingual data in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts. More precisely, it was pretrained with two objectives:

- Masked language modeling (MLM): taking a sentence, the model randomly masks 15% of the words in the input then run the entire masked sentence through the model and has to predict the masked words. This is different from traditional recurrent neural networks (RNNs) that usually see the words one after the other, or from autoregressive models like GPT which internally mask the future tokens. It allows the model to learn a bidirectional representation of the sentence.
- Next sentence prediction (NSP): the models concatenates two masked sentences as inputs during pretraining. Sometimes they correspond to sentences that were next to each other in the original text, sometimes not. The model then has to predict if the two sentences were following each other or not.

This way, the model learns an inner representation of the languages in the training set that can then be used to extract features useful for downstream tasks: if you have a dataset of labeled sentences for instance, you can train a standard classifier using the features produced by the BERT model as inputs.

## 1.4 Related Work

In [Zhao, 2021], event prediction involves forecasting the time, location, and details of events. The use of natural language processing (NLP) methods including sanitization, tokenization, POS tagging, and named entity recognition (NER) are essential for extracting event information, with a focus on accuracy as the main performance metric.

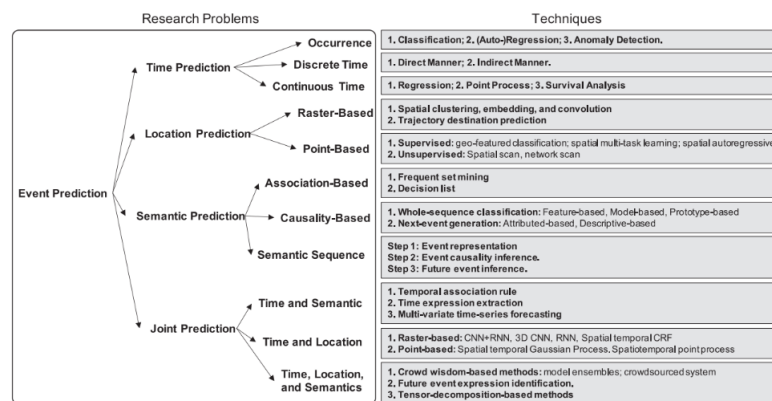


Figure 1.1: Taxonomy of event prediction problems and techniques.

The Figure 1.1 shows the different techniques used in this article for event detection problems.

In [Gupta and Katarya, 2020], the analysis of health-related Twitter data analysis reveals that the Multinomial Naive Bayes Modal outperformed other classifiers with an F-measure of 0.811. This study evaluated 1240 publications indexed in multiple scientific databases from 2010 to 2018.

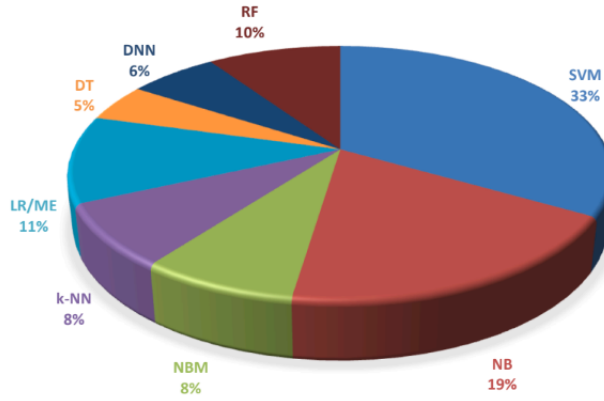


Figure 1.2: Distribution of ML methods for health-related text classification by social media based surveillance systems.

NB: Naive Bayes, NBM: Multinomial Naive Bayes, k-NN: k-Nearest Neighbor, ME: Maximum Entropy, LR: Logistic Regression, DT: Decision Tree, DNN: Deep Neural Network, RF: Random Forest, SVM: Support Vector Machine.

The Figure 1.2 demonstrate the distribution of ML methods for Health related text classification and it shows that the most technique used is Naive Bayes.

In [Liu et al., 2020], the Multi-Lingual Event Mining (MLEM) model, integrating Word2vec for synonym merging, was emphasized for its efficiency and effectiveness in detecting multilingual events. This technique is suitable for the language diversity in social media platforms.

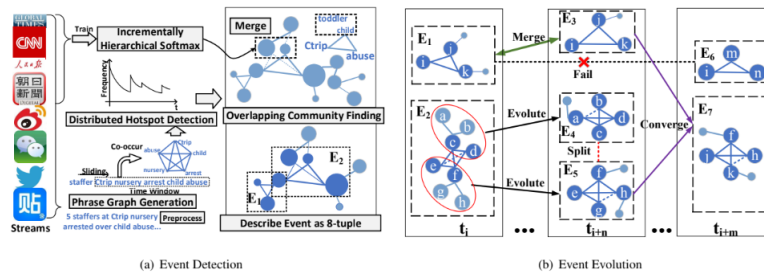


Figure 1.3: Illustration of event detection and evolution from multi-lingual text streams.

The Figure 1.3 illustrates the processes of event detection and evolution in multilingual



text streams, showcasing methods for identifying, merging, and tracking the development of events over time using various computational models and data representations.

In [Zeng et al., 2021], a variety of AI models were explored, focusing on infectious disease detection and offering early warning and trend prediction capabilities. The use of these models underscores the potential of AI in enhancing public health initiatives.

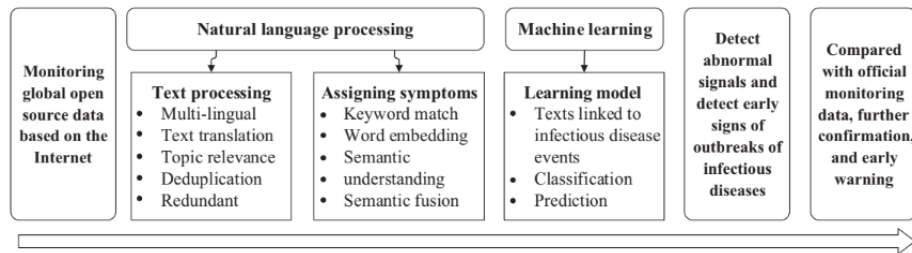


Figure 1.4: AI techniques in Internet-based global epidemic monitoring.

This Figure 1.4 outlines the application of AI techniques in Internet-based global epidemic monitoring, demonstrating a workflow from data monitoring and natural language processing to machine learning models that detect, classify, and predict epidemic events based on textual data analysis.

In [Edo-Osagie et al., 2020], supervised Learning was the most commonly used approach to classify tweets or predict outcomes related to public health issues. Techniques such as Support Vector Machines (SVM), Naive Bayes classifiers, and logistic regression were frequently employed for tasks like classifying tweets into disease-related categories or sentiment analysis related to health discussions.

Some studies utilized unsupervised learning methods, such as clustering and topic modelling, to uncover patterns, trends, or topics within health-related tweets.

Recent studies have begun to explore deep learning techniques, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), for more complex tasks such as analysing tweet images for harmful algal blooms detection or utilizing sequence models to predict disease outbreaks based on tweet sequences.

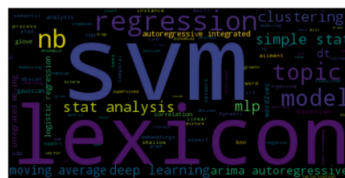


Figure 1.5: Word cloud of statistical and machine learning methods discovered in review.

This Figure 1.5 displays a word cloud of statistical and machine learning methods such

as "SVM", "regression", and "lexicon", highlighting the key techniques and concepts discussed or utilized in tweets analysis.

The table 1.1 presents a summary of the techniques, performance, and significant contributions to the field, highlighting the increasing reliance on machine learning and AI for data-driven insights in public health and event prediction domains.

Article Title	Model used	Language	Key Highlights
Event Prediction in the big data era [Zhao, 2021]	CNN	- .	NLP methods: sanitization, tokenization, POS tagging, NER
Social media-based surveillance [Gupta and Katarya, 2020]	Multinomial Naive Bayes EnglishModal	English	Syndromic surveillance systems
Event Detection and Evolution in Multilingual Social Streams [Liu et al., 2020]	MLEM model (Word2vec to merge synonyms)	Multiple languages	Incremental word2vec, Multilingual event detection
AI-enabled Public Health Surveillance [Zeng et al., 2021]	Various AI models	English	AI in infectious disease detection, Early warning and trend prediction
AI Techniques in Public Health [Edo-Osagie et al., 2020]	Deep learning, reinforcement learning, Bayesian networks	English	Novel data sources for surveillance, AI enhances data analytics

Table 1.1: Articles Analysis

## 1.5 Conclusion

This chapter outlines the critical NLP techniques and AI models that shape the landscape of text-based data analysis and event prediction. By integrating these advanced technologies, we aim to develop a model that enhances the prediction and management of zoonosanitary events, thereby improving public and animal health surveillance.

# Chapter 2

## Methodology

Preprocessing in both machine learning and LLM involves preparing raw data for analysis or model training. It includes steps like handling missing values, scaling features, and in the case of LLM, tasks such as removing punctuation, tokenization, and removing stopwords. The aim is to enhance data quality, reduce noise, and facilitate model learning for more accurate predictions or language understanding.

### 2.1 Proposed Solution

We started by exploring the distribution of articles by language, providing insights into the corpus composition, which is crucial for targeted NLP tasks. And standard preprocessing steps such as removal of stop words, lemmatization, stemming are suggested to prepare the text data for modelling.

For Data Visualization, we used libraries like Plotly for dynamic visualization of data distributions, which aids in better understanding and presentation of the data insights.

For preprocessing, we have begun by loading a dataset of articles related to zoosanitary events. To ensure data quality, duplicates are identified and removed based on the 'Article' column. Finally, The dataset contains articles in multiple languages, which are filtered and processed separately to accommodate linguistic differences in the model training.

To address the insufficiency of the dataset provided by CNVZ, web scraping is utilized to gather additional articles. This helps in creating a balanced dataset for training the model. The original and scraped data are combined, cleaned, and deduplicated to form a comprehensive dataset for training and validation.

Using text-based features such as TF-IDF vectors to capture the importance of words in documents. While specific models are not detailed in the provided text, typical approaches would include logistic regression for classification and BERT (or similar LLMs) for deep learning-based text analysis.

Given the diversity in language, separate models or multilingual models like BERT could be employed to handle text in different languages effectively.

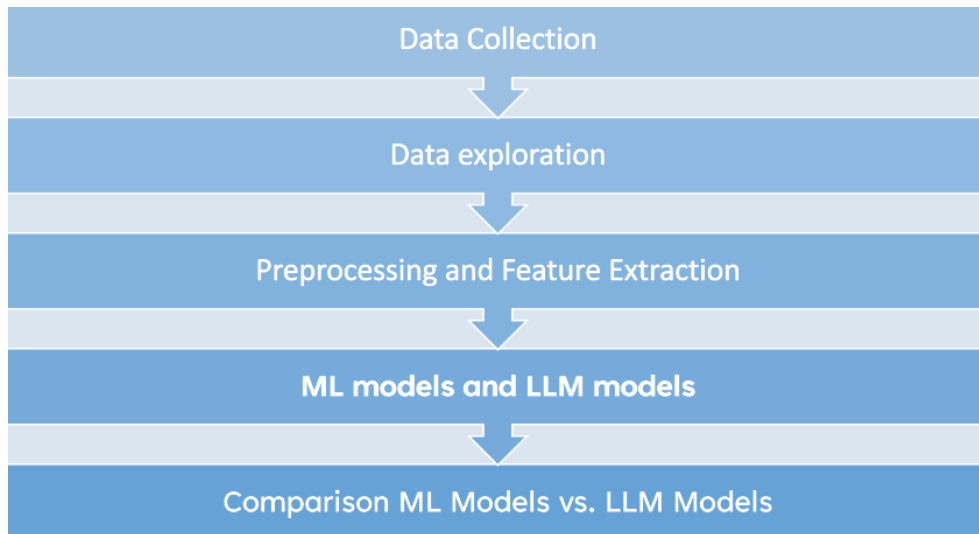


Figure 2.1: Proposed Architecture

This Figure 2.1 illustrates the proposed architecture for event detection workflow, beginning with Data Collection, followed by Data Exploration, Preprocessing and Feature Extraction. Finally, in the Chapter 3 ML models and LLM models, and concluding with comparison between the two approaches, each step building upon the previous to systematically develop and assess a predictive model.

## 2.2 Data Description and Collection

In the dataset provided by CNVZ there were English, French, and Arabic-language articles describing animal illness epidemics. We systematically categorized those articles by language, We had to identify the number of articles in every language. For that, we created this plot:

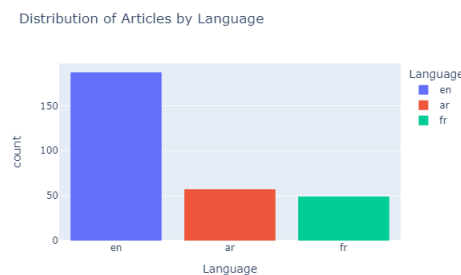


Figure 2.2: CNVZ articles

As the Figure 2.2, we have 49 articles in French, 57 in Arabic and 187 in English. Based

on the plot, we had to scrap nearly the same amount of articles in every language. Given the nature of the content, all articles were tagged as events, reflected in a column labelled STATE with the designation "yes event". Then, we considered moving to web scrapping to balance our data.

### Web Scrapping Strategy

The choice of scrapping tools will depend on the websites targeted for data extraction. Our choice was Python library BeautifulSoup based on the complexity nature of the web pages. Then our next step is to identify a list of credible and relevant websites such as news outlets, veterinary blogs, health forums, and research databases that regularly publish content related to animal health, general health topics, and environmental concerns.

First, we extract from a Danish Website that describes many illnesses and their treatments<sup>1</sup>. Second, we extract descriptions from Wikipedia<sup>2</sup>. Finally, we extract descriptions of human illnesses from the World Health Organization<sup>3</sup>. These articles were diverse to maintain the quality of our data. For the data extraction, we need to define the HTML structures of the target websites to extract textual content efficiently and the articles will be labelled according to their relevance to zoosanitary (no event) events. These are Non-relevant articles. They will serve as negative instances, which are just as crucial for the model's training.

After scrapping the amount of articles we need, we visualize the articles in this plot to compare it to the previous one:

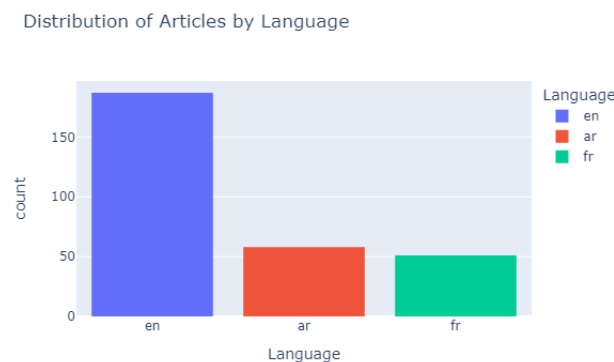


Figure 2.3: Scrapped Articles

<sup>1</sup>Danish website: <https://www.ssi.dk/aktuelt/nyheder>.

<sup>2</sup>Wikipedia: [https://ar.wikipedia.org/wiki/%D9%85%D8%B1%D8%B6\\_%D8%A7%D9%84%D8%AF%D9%88%D8%A7%D8%AC%D9%86%D8%A8%D8%B9%D8%B6\\_%D8%A7%D9%85%D8%B1%D8%A7%D8%B6\\_%D8%A7%D9%84%D8%AF%D9%88%D8%A7%D8%AC%D9%86](https://ar.wikipedia.org/wiki/%D9%85%D8%B1%D8%B6_%D8%A7%D9%84%D8%AF%D9%88%D8%A7%D8%AC%D9%86%D8%A8%D8%B9%D8%B6_%D8%A7%D9%85%D8%B1%D8%A7%D8%B6_%D8%A7%D9%84%D8%AF%D9%88%D8%A7%D8%AC%D9%86).

<sup>3</sup><https://www.who.int/health-topics/>

As the Figure 2.3 shows, we end up with 49 articles in French, 57 in Arabic and 187 in English, which is nearly equal to the articles CNVZ has provided.

Now it's time to merge our scrapped and original data. Integrating web-scraped data with existing datasets to enrich the data, especially useful for enhancing the dataset size for training machine learning models. After merging datasets, we need to shuffle the rows.

## 2.3 Data exploration

After collecting all data, we have 296 articles for no events (general topics) and 294 for yes events (zoosanitary event). We have nearly the same distribution as the Figure .

Moving to adding new features, we added two columns:

- char count: it contains the number of characters for each article.
- word count: it contains the number of words for each article.

The Figure 2.4 pinpoints those numbers:

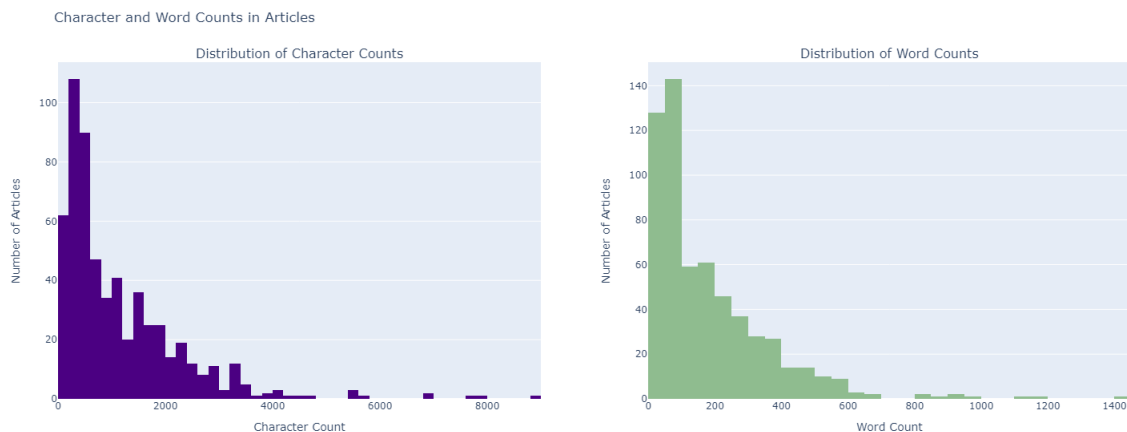


Figure 2.4: Char Word count

From the observation most articles lean towards 200 characters and 50 words, with outliers having significantly more words (800, 850, 900, 950, 1,100, and 1,400 words), we can draw several conclusions and consider different actions. The majority of articles in our dataset are relatively short, averaging around 200 characters and 50 words. Our approach is to remove these outliers in the preprocessing techniques.

Then, we calculated the average word length for each article and created a histogram shown in Figure 2.5 to visualize the distribution of average word lengths:

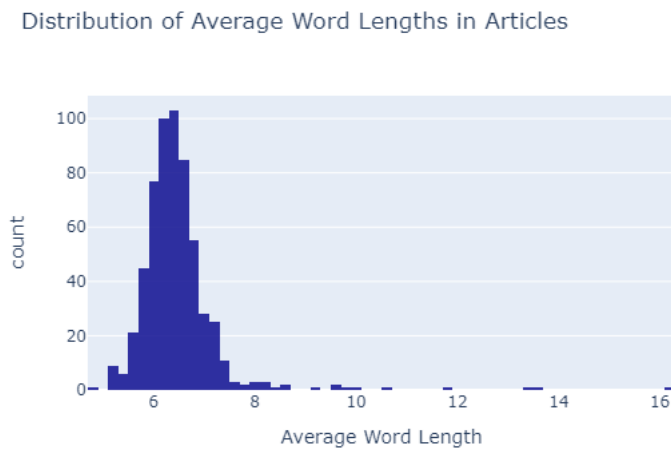


Figure 2.5: Average word length

We might consider additional preprocessing steps, such as stemming, lemmatization. Sentiment Score: This would typically give us a polarity score (how positive or negative the text is) and subjectivity (how subjective or opinionated the text is)



Figure 2.6: Polarity and Subjectivity Distribution

In The Figure 2.6 the polarity scores range from negative to positive, indicating a mix of negative, neutral, and positive sentiments across the dataset. Most of the data points are clustered around the center, suggesting a large portion of the text might be relatively neutral in sentiment. This result is justified because the majority of this text are articles and should be neutral.

The subjectivity scores are spread out, with many data points appearing toward the lower end of the scale, which suggests that a significant portion of the text is likely to be objective or factual in nature. This result is also justified because the majority of this text are

articles and should be objective.

For preprocessing we consider: **Noise Removal:** Ensure that the text is cleaned of noise such as HTML tags, URLs, emojis, and special characters that do not contribute to sentiment, as these can distort analysis results. **Normalization:** Text should be normalized by converting it to lowercase, removing punctuation, and using stemming or lemmatization to reduce words to their base or root form. **Handling Outliers:** Investigate outliers with extremely high or low polarity or subjectivity scores to understand if they're due to actual sentiment or a preprocessing error.

Now moving to Part-of-Speech Tagging to tag each word in the text with its part of speech (noun, verb, adjective, etc.), and then we can count the frequency of each part of speech. We draw a plot to showcase the distribution:

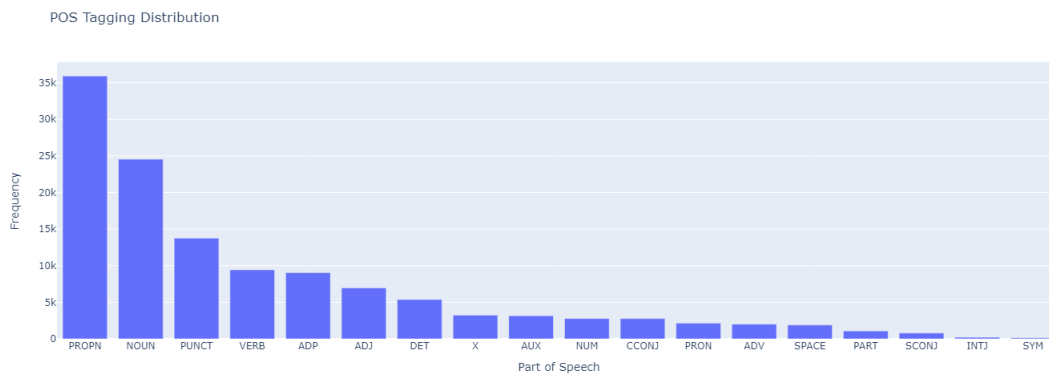


Figure 2.7: POS Tagging distribution

In the Figure 2.7 the most frequent POS tags are proper nouns (PROPN), common nouns (NOUN), punctuation (PUNCT), and verbs (VERB). This suggests that the dataset contains a substantial amount of descriptive or narrative content, which is typical for informative texts. The high frequency of nouns and proper nouns could indicate the presence of specific entities, which are often crucial in zoosanitary contexts as they can refer to diseases, species, locations, etc.

For preprocessing, we consider: **Noise Reduction:** Punctuation marks are highly frequent. We intend to remove punctuation because it does not contribute to the understanding of zoosanitary events.

We can develop features based on the frequency of specific POS tags that could be indicative of zoosanitary events. For example, an unusually high number of proper nouns might correlate with texts describing outbreaks or specific case reports.



For word frequency count, we created this plot :

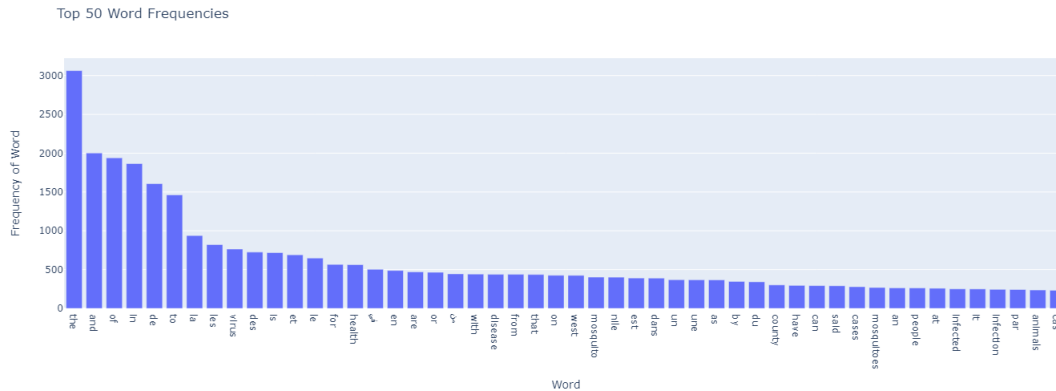


Figure 2.8: POS Tagging distribution

Notice anything odd about the words that appear in this Figure 2.8 ? Do these words actually tell us much about the articles? These words are all so commonly occurring words which you could find just anywhere else. Not just in Zoosanitary articles. Therefore, we must find some way to preprocess our dataset first to strip out all these commonly occurring words, which do not bring much to the table.

## 2.4 Preprocessing and Feature Extraction

First, we convert all text to lowercase to maintain consistency. Next, we eliminate punctuation, which lacks relevance in understanding zoosanitary events, and discard overly lengthy articles that could mislead our model. We also remove common stopwords to concentrate on more significant terms, tailoring this process for English, French, and Arabic texts. For categorical variables like language and state, we apply label encoding to transform each category into a unique numerical identifier, enhancing the data's compatibility with machine learning algorithms.

### Features extraction

To enhance our analysis, we introduced several new features, including character count and word count, which measure the length of each article in characters and words, respectively. We also calculated the average word length for each article. These features are critical for understanding the typical structure of the texts we are dealing with, which aids in subsequent modeling. Additionally, we derived polarity and subjectivity scores, providing a preliminary insight into the emotional and factual content of the articles. Part-of-Speech tagging was utilized to further dissect the textual data, identifying the grammatical structure of the words used, which helps in recognizing the most commonly used types of words

and their roles in the texts. Finally, we used the Term Frequency-Inverse Document Frequency (TF-IDF). vectorization method to transform the preprocessed text into numerical features. These enhancements in feature engineering are aimed at improving the robustness and accuracy of our predictive models in identifying zoonosanitary events from text data.

In the Table 2.1 we can see the differences between the articles before and after preprocessing.

Before Preprocessing	After Preprocessing
<p>En Gironde après des signalements fin juillet, l'agence régionale de santé confirme ce mercredi l'existence de sept cas humains du virus du Nil occidental dans le département. Quatre cas sont avérés et trois sont en attente de confirmation. Sept cas du West Nile virus ont été détectés en Gironde, prévient ce mercredi l'agence régionale de santé de Nouvelle-Aquitaine. C'est une première souligne l'ARS. Jusqu'à présent, les seuls cas détectés en France métropolitaine se situaient sur le pourtour méditerranéen. Le virus du Nil occidental se transmet à l'homme via le moustique Culex, l'espèce prédominante dans l'hexagone lui-même infecté par des oiseaux contaminés. Des chevaux ont également été testés positifs à l'automne 2022. Ce virus ne se transmet pas d'homme à homme et dans 80% des cas, l'infection est asymptomatique. Pour les 20% restants, les symptômes ressemblent à ceux d'une grippe: fièvre, douleurs et maux de tête parfois accompagnés d'une éruption cutanée. Dans moins de 1% des cas, le virus du Nil occidental peut provoquer des complications neurologiques, en particulier chez les sujets immunodéprimés. Si vous avez des symptômes, parlez-en à votre médecin et si vous êtes positifs au virus du Nil occidental, déclarez-vous auprès de l'ARS, c'est obligatoire afin d'identifier les lieux possibles de contamination et repérer d'autres cas.</p>	<p>gironde, après, signalements, fin, juillet, lagence, régionale, santé, confirme, mercredi, l'existence, sept, cas, humains, virus, nil, occidental, département, quatre, cas, avérés, trois, attente, confirmation, sept, cas, west, nile, virus, détectés, gironde, prévient, mercredi, lagence, régionale, santé, nouvelleaquitaine, quatre, cas, avérés, trois, autres, cours, confirmation, cest, première, souligne, lars, jusqu'à, présent, seuls, cas, détectés, france, métropolitaine, situaient, pourtour, méditerranéen, virus, nil, occidental, transmet, l'homme, via, moustique, culex, l'espèce, prédominante, l'hexagone, luimême, infecté, oiseaux, contaminés, chevaux, également, testés, positifs, l'automne, virus, transmet, d'homme, homme, cas, linfection, asymptomatique, restants, symptômes, ressemblent, ceux, d'une, grippe, fièvre, douleurs, maux, tête, parfois, accompagnés, d'une, éruption, cutanée, moins, cas, virus, nil, occidental, peut, provoquer, complication, neurologiques, particulier, chez, sujets, immunodéprimés, si, symptômes, parlez, médecin, si, positifs, virus, nil, occidental, déclarezvous, auprès, lars, cest, obligatoire, afin, d'identifier, lieux, possible, contamination, repérer, d'autres, cas</p>

Table 2.1: Comparison of Text Before and After Preprocessing

<sup>3</sup>TF-IDF assesses the relevance of each term in a document relative to its frequency across all documents, producing a matrix where rows represent documents and columns represent unique terms, with values reflecting the significance of each term in its corresponding document.

## 2.5 Conclusion

The methodology chapter outlines a comprehensive approach to preprocess data, integrate web scraping, and prepare baseline models for analyzing zoonosanitary events. By systematically categorizing articles, conducting exploratory data analysis, and implementing preprocessing techniques such as TF-IDF vectorization and feature engineering, the chapter sets a solid foundation for subsequent modeling efforts. The integration of web-scraped data enriches the dataset, enhancing the model's ability to classify zoonosanitary events effectively. Through a combination of text-based and numerical features, the prepared data is poised for training robust machine learning models.

# Chapter 3

## Approach and Results

In this chapter, we'll talk about the models we used. Because our dataset is small, we made different types of models besides the LLM model. We made three classification models: Random Forest, SVM, and Logistic Regression. For the LLM model, because our data is in different languages, we used Multilingual BERT. After that, we looked at how well each model worked and compared their results.

### 3.1 Machine Learning Models Training and Evaluation

#### 3.1.1 Machine Learning Model Training:

- **TF-IDF Features (X\_tfidf):** The tfidf\_matrix contains text features transformed into numerical values. TF-IDF weighs terms based on how common or rare they are across documents, helping to enhance the importance of distinctive terms.
- **Numerical Features (X\_numerical):** It includes char\_count, word\_count, av\_word\_length, polarity, and subjectivity extracted from the text. These features provide additional information about the text such as length, sentiment, and structure, which can be crucial for understanding context and nuances in language.
- **Feature Combination (X\_combined):** The numerical features and TF-IDF features are horizontally stacked (np.hstack()) to create a unified feature matrix. This combination allows the model to leverage both text-based and derived numerical features for classification.

The target variable STATE\_encoded is prepared using LabelEncoder, which encodes categorical string classes into integers. This is necessary for fitting machine learning models which require numerical input.

The dataset is split into training and testing sets using train\_test\_split with a test size of 20%, allowing the model to be trained on 80% of the data and validated on 20%. The random\_state=42 ensures reproducibility of the split

### 3.1.2 Classification Models

In these two approaches, we used TF-IDF Vectorization

**-Random Forest Classifier:** A RandomForestClassifier with 100 trees (n\_estimators=100) is initialized and trained on the training data. Random forests are robust to overfitting and are effective for classification tasks.

**-SVM Classifier:** An SVC (Support Vector Classifier) with a linear kernel is used. SVMs are effective in high-dimensional spaces and are ideal for binary classification tasks.

With Logistic Regression, we tried with several techniques TF-IDF Vectorization, Hashing Vectorization and Word2Vec Features

**-Logistic Regression :** This approach begins by loading a dataset from a CSV file containing several features including text, language, and various text-derived numerical features. We drop many columns from the dataset that may not be necessary for the subsequent analysis, simplifying the data structure to focus primarily on text and its direct features.

## 3.2 Large Language Model (BERT)

This section focuses on utilizing the Multilingual BERT (mBERT) model from the Hugging Face transformers library for text classification tasks across different languages, illustrating two distinct implementation approaches.

### 3.2.1 Environment Setup

For the configuration of the environment we followed the following steps:

- **Datasets Library:** Installation of the datasets library from Hugging Face, which provides a simple way to work with a wide range of datasets tailored for machine learning and natural language processing tasks.  
`!pip install datasets`
- **Virtual Environment Setup:** Creation of a Python virtual environment to manage dependencies effectively without conflicts with other project dependencies.  
`!pip install virtualenv`  
`!virtualenv venv`  
`!source venv/bin/activate`
- **Hugging Face Transformers:** Installation of the transformers library, which provides access to pre-trained models like BERT and utilities for working with them.  
`!pip install transformers[torch]`

- **Accelerate Library:** Updated installation of the accelerate library to simplify and accelerate training on multi-GPU setups.

**!pip install accelerate -U**

### 3.2.2 Multilingual BERT Implementations

#### Standard Multilingual BERT:

This approach initializes the tokenizer and model using `bert-base-multilingual-cased`. This model can handle texts in multiple languages and is pre-trained on a large corpus. `train_test_split` is used to divide the data into training and evaluation sets with 20% of the data reserved for evaluation (`test_size=0.2`). This is a standard practice to ensure that the model is tested on unseen data. `random_state=42` is set for reproducibility.

The `train_df` and `eval_df` DataFrames are converted into `datasets.Dataset` format. This conversion facilitates better integration with the Hugging Face library, enabling efficient data handling and operations. The `AutoTokenizer` and `AutoModelForSequenceClassification` are loaded with the "Bert-base-multilingual-cased" model. This model variant supports multiple languages and is cased, meaning it differentiates between uppercase and lowercase, which can capture more linguistic nuances compared to uncased models.

#### Data Tokenization:

- We used the following settings for the tokenization process: A custom function `tokenize_function` is defined to tokenize the texts. This function applies the tokenizer with specific settings:
  - **Padding and Truncation:** Ensures all tokenized outputs are of the same maximum length (512 tokens), padding shorter texts and truncating longer ones. This uniform length is crucial for batching and processing through the BERT model.
  - **Return Tensors:** Specifies that the output format should be PyTorch tensors ("`pt`"), compatible with the training framework used.
- Labels are included in the output dictionary to facilitate training and evaluation.

#### Training Setup

- We selected the following parameters for the training process: `TrainingArguments` configures the training process:
  - **Output Directory:** Specifies where the model checkpoints and outputs will be saved.
  - **Batch Size:** Sets to 4 per device to manage GPU memory usage effectively.

- Number of Epochs: 10 epochs are chosen to allow the model sufficient exposure to the training data while preventing overfitting.
- Logging Directory: Logs are stored for monitoring the training process, useful for debugging and optimization.
- We initialized The Trainer object with the model, training arguments, training dataset, and tokenizer. This setup simplifies the training process by handling the training loop, optimization, and saving automatically.
- We trained our model using `trainer.train()` starts the training process, iterating through the data for the specified number of epochs, updating model weights to minimize the classification loss.
- After training we evaluated the model using the evaluation dataset which has been tokenized similarly to the training dataset. `trainer.evaluate()` computes the model performance metrics on the evaluation dataset, providing insights such as loss and accuracy.
- we predicted the results using `trainer.predict()` . Predictions and true labels are extracted and compiled into a DataFrame alongside the original text data. This DataFrame facilitates a direct comparison of model predictions against actual labels, crucial for assessing model performance and understanding potential areas for improvement.

we use the new features mentioned on the second chapter such as char count and word count to train our second BERT model to see if they affect our results.

### Custom Implementation with Multilingual BERT:

outlines the process of integrating advanced NLP techniques with deep learning to perform text classification using a custom BERT model that incorporates additional numerical and POS (Part-of-Speech) features. Let's break down the major components and the rationale behind the chosen parameters and methods.

- Model Architecture:
  - The CustomBERTModel incorporates BERT's output with additional numerical and POS features. This approach leverages the contextual understanding of BERT and enriches it with external features that could provide more insights into the text's characteristics.
  - The combined features are passed through a fully connected neural network for classification. The network includes ReLU activations for non-linearity and a final dense layer suited for binary classification.

**Training:**

- Optimization and Loss Function:
  - The AdamW optimizer is used with a learning rate of 5e-5, which is a standard choice for BERT-related tasks due to its effectiveness in handling sparse gradients and adaptive learning rate capabilities.
  - The CrossEntropyLoss function is suitable for binary classification tasks with logits output.
- Training Loop:
  - A custom training function handles the training process, including gradient accumulation, which allows for effectively larger batch sizes than those that can be physically accommodated in memory. This technique helps in stabilizing the training updates.

### 3.3 Comparison ML Models vs. LLM Models

**Baseline Models:** This approach likely uses traditional machine learning models like Random Forest and SVM as baselines. These models, while robust and often performing well with structured features, generally lack the nuanced understanding of text data that comes from contextual embeddings.

**LLM Models (mBERT):** mBERT models leverage pre-trained contextual embeddings that understand language nuances and the syntax of multiple languages, making them superior for text-heavy applications where context is crucial.

#### Multilingual BERT Confusion Matrix

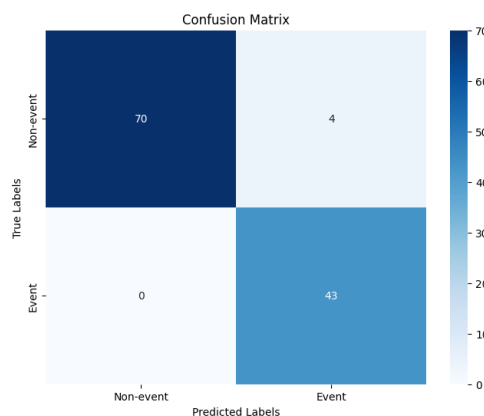


Figure 3.1: Confusion Matrix



**Performance Comparison:**

Model	Precision	Accuracy	Recall	f1-score
Random Forest	0.97	0.97	0.97	0.97
SVM	0.91	0.92	0.93	0.92
logistic regression	0.9	0.9	-	-
Multilingual BERT	0.96	0.97	0.97	0.96
Custom Multilingual BERT	0.96	0.97	0.96	0.96

Table 3.1: Models comparison table

- **Accuracy and Generalization:** LLM models, particularly those trained with comprehensive tokenization and extended features (as shown in the custom mBERT implementation), are expected to outperform traditional models in tasks involving complex language data.
- **Computational Resources:** LLMs require significantly more computational resources and are slower to train compared to baseline models like Random Forest or SVM, which might be more efficient but less effective in capturing linguistic subtleties.

**Conclusion:** In conclusion, the similarity in performance between classification and LLM models, both achieving around 0.96 accuracy, suggests that the dataset size may have played a role in equalizing their effectiveness. Given the relatively small dataset size, both models likely had sufficient data to learn patterns effectively, leading to comparable results. Therefore, while both approaches offer viable solutions for predicting animal disease spread from multilingual articles, future investigations with larger datasets could provide deeper insights into potential performance disparities.

## Chapter 4

# Conclusions and Perspectives

The project on Automated Zoosanitary Surveillance System in Tunisia achieved significant milestones in enhancing the detection of zoonotic diseases through the development of an automated system that utilizes advanced machine learning and natural language processing technologies. This system effectively analyzes data from various sources such as news outlets and social media, enabling early warnings and timely interventions. The integration of a Large Language Model (BERT) has notably improved the system's ability to process and understand multilingual text data, which is crucial for Tunisia's diverse linguistic environment.

One of the major achievements of this project was the implementation of effective web scraping strategies to augment the dataset provided by the National Center for Veterinary Surveillance (CNVZ), ensuring robustness in the system's training and validation phases. The evaluation of various machine learning models, including Random Forest, SVM, and Logistic Regression alongside BERT, demonstrated that while traditional models provided strong baseline performances, BERT excelled in handling complex language data due to its advanced capabilities.

The project faced challenges such as ensuring the quality and diversity of the data collected via web scraping, and managing the complexities of multilingual datasets. These challenges underscored the importance of rigorous data cleaning and preprocessing to enhance model accuracy, as well as the use of sophisticated techniques like tokenization and multilingual models for language management.

Looking forward, the project aims to scale the system to handle more data sources and integrate real-time data processing for quicker response times. Further enhancements to the model are planned, including deeper integration of NLP features and possibly custom modifications to the BERT model to tailor it more closely to the specific needs of zoosanitary surveillance. Efforts will also be directed towards integrating the system with existing national veterinary surveillance operations to enhance its practical impact and utility.

In conclusion, the Automated Zoosanitary Surveillance System represents a significant advancement in using technology to safeguard public and animal health in Tunisia. It

aligns with global trends towards the digital transformation of healthcare and veterinary services and provides a scalable model for similar initiatives worldwide, especially in regions facing similar multilingual and multi-dialectal challenges.

# Bibliography

- [URL, a] LLM description. <https://www.ibm.com/topics/random-forest>.
- [URL, b] Logistic Regression. <https://www.ibm.com/topics/logistic-regression>.
- [URL, c] multilingual BERT. <https://huggingface.co/google-bert/bert-base-multilingual-cased>.
- [URL, d] Nlp techniques . <https://www.techopedia.com/definition/34948/large-language-model-llm>.
- [URL, e] Random Forest. <https://www.ibm.com/topics/natural-language-processing>.
- [URL, f] SVM. <https://www.ibm.com/topics/support-vector-machine>.
- [Edo-Osagie et al., 2020] Edo-Osagie, O., De La Iglesia, B., Lake, I., and Edeghere, O. (2020). A scoping review of the use of twitter for public health research. *Computers in biology and medicine*, 122:103770.
- [Gupta and Katarya, 2020] Gupta, A. and Katarya, R. (2020). Social media based surveillance systems for healthcare using machine learning: a systematic review. *Journal of biomedical informatics*, 108:103500.
- [Liu et al., 2020] Liu, Y., Peng, H., Li, J., Song, Y., and Li, X. (2020). Event detection and evolution in multi-lingual social streams. *Frontiers of Computer Science*, 14:1–15.
- [Zeng et al., 2021] Zeng, D., Cao, Z., and Neill, D. B. (2021). Artificial intelligence-enabled public health surveillance—from local detection to global epidemic monitoring and control. In *Artificial intelligence in medicine*, pages 437–453. Elsevier.
- [Zhao, 2021] Zhao, L. (2021). Event prediction in the big data era: A systematic survey. *ACM Computing Surveys (CSUR)*, 54(5):1–37.