Characterization and detection of semantic textual outliers

Jérémie Pantin

École Doctorale Informatique, Télécommunications et Electronique
Laboratoire d'Informatique de Paris 6
Équipe LFI

Détection et caractérisation sémantique de données textuelles aberrantes

Par Jeremie PANTIN

Thèse de doctorat d'Informatique

Dirigée par Christophe Marsala

Présentée et soutenue publiquement le 11 Septembre 2023

1. Rapporteur	Gregory SMITS	Brest IMT Atlantique
2. Rapporteur	Anne LAURENT	Montpellier LIRMM
3. Examinateur	Bernd AMANN	Paris LIP6 - Sorbonne Université
Directeur de thèse	Christophe MARSALA	Paris LIP6 - Sorbonne Université

École Doctorale Informatique, Télécommunications et Electronique
Laboratoire d'Informatique de Paris 6
Learning, Fuzzy and Intelligent systems

Detection and semantic characterisation of textual outliers

Jeremie PANTIN

Ph.D. thesis in Computer Science

Supervised by Christophe Marsala

Presented and publicly defended on September 11, 2023

1. Rapporteur	Gregory SMITS Brest IMT Atlantique
2. Rapporteur	Anne LAURENT Montpellier LIRMM
3. Examinateur	Bernd AMANN Paris LIP6 - Sorbonne Université
Directeur de thèse	Christophe MARSALA

Jeremie PANTIN

Detection and semantic characterisation of textual outliers

On outlier analysis with text data, September 11, 2023

Rapporteurs: Gregory SMITS et Anne LAURENT

Examinateur: Bernd AMANN

Directeur de thèse: Christophe MARSALA

Sorbonne Université

Learning, Fuzzy and Intelligent systems

Laboratoire d'Informatique de Paris 6

École Doctorale Informatique, Télécommunications et Electronique

4 Place Jussieu

75005 and Paris

In memory of my mother.

Abstract

Recent artificial intelligence advances are bound to the increasing number of data that industry and research fields have access to. Hence, it can be wrong to admit that artificial intelligence is only referring to processing data and observations. A global view of the field refer the ability of the machine to perform and/or imitate cognitive behaviors such as decision making, learning, perceiving and reasoning. Thus, we expect from the machine to perform actions following human reflection and mechanisms, based on given environments or observations.

Such expectation is recently motivated by the growing usage of internet by the population and appearance of large number of data. Machine learning answers to the problem of handling dedicated tasks with a vast variety of data. Such algorithms can be either simple or difficult to handle and data might be similar to deal with. Low dimensional data (2-dimension or 3-dimension) with an intuitive representation (average of *baguette* price by years) are easier to interpret/explain for a human than data with thousands of dimensions. For low dimensional data, the error leads to a significant shift against normal data, but for the case of high dimensional data it is different.

Outlier detection (or *anomaly detection*, or *novelty detection*) is the study of anomalous observations for detecting what is normal and abnormal. Methods that perform such task are algorithms, methods or models that are based on data distributions. Different families of approaches can be found in the literature of outlier detection, and they are mainly independent of ground truth. They perform outlier analysis by detecting the principal behaviors of majority of observations. Thus, data that differ from normal distribution are considered *noise* or *outlier*. We detail the application of outlier detection with text. Despite recent progress in natural language processing, computer still lack profound understanding of human language in absence of information. For instance, the sentence *"A smile is a curve that set everything straight"* has several levels of understanding and a machine can encounter hardship to chose the right level of lecture.

This thesis presents the analysis of high-dimensional outliers, applied to text. Recent advances in anomaly detection and outlier detection are not significantly represented with text data and we propose to highlight the main differences with high-dimensional outliers. We also approach ensemble methods that are nearly nonexistent in the literature for our context. Finally, an application of outlier detection for elevate results on abstractive summarization is conducted.

We propose GenTO, a method that prepares and generates split of data in which anomalies and outliers are inserted. Based on this method, evaluation and benchmark of outlier detection approaches is proposed with documents. Also, learning without supervision often leads models to rely in some hyperparameter. For instance, Local

Outlier Factor relies to the k-nearest neighbors for computing the local density. Thus, choosing the right value for k is crucial. In this regard, we explore the influence of such parameter for text data.

While choosing one model lead to obvious bias against real-world data, ensemble methods allow to mitigate such problem. They are particularly efficient with outlier analysis. Indeed, the selection of several values for one hyperparameter can help to detect strong outliers. Importance is then tackled and can help a human to understand the output of black box model. Thus, the interpretability of outlier detection models is questioned. We find that for numerous dataset, a low number of features can be selected as oracle. The association of complete models and restrained models helps to mitigate the black-box effect of some approaches.

In some cases, outlier detection refers to noise removal or anomaly detection. Some applications can benefit from the characteristic of such task. Mail spam detection and fake news detection are one example, but we propose to use outlier detection approaches for weak signal exploration in marketing project. Thus, we find that the model of the literature help to improve unsupervised abstractive summarization, and also to find weak signals in text.

Résumé

L'intelligence artificielle (IA) a connu une croissance spectaculaire ces dernières années avec l'évolution de l'informatique, du hardware et d'internet. Les récents progrès dans ce domaine sont liés au nombre croissant de données auxquelles l'industrie et la recherche ont accès. Il est difficile d'admettre que l'intelligence artificielle ne concerne que le traitement des données. Une vision globale de l'IA fait référence à la capacité de la machine à réaliser et/ou imiter des comportements cognitifs tels que la prise de décision, l'apprentissage, la perception et le raisonnement. Ainsi, nous attendons de la machine qu'elle effectue des actions en suivant la réflexion et les mécanismes humains, en fonction d'environnements ou d'observations donnés.

Cette attente a récemment été motivée par l'utilisation croissante d'internet par la population, avec l'apparition d'un grand nombre de données. L'apprentissage automatique répond au problème du traitement de tâches spécifiques pour une grande variété de données. Ces algorithmes peuvent être simples ou difficiles à mettre en place, et c'est par ailleurs le même constat qui peut être fait pour les données. Les données de faible dimension (2 ou 3 dimensions) avec une représentation intuitive (ex. moyenne du prix des *baguette* par années) sont plus faciles à interpréter/expliquer pour un humain que les données avec des milliers de dimensions. Pour les données à faible dimension, une donnée aberrante conduit souvent à un décalage conséquent par rapport aux données normales, mais pour le cas des données à haute dimension, c'est différent. Les données à haute dimension ont besoin d'un traitement particulier qui consiste aussi bien à réduire la dimensionalité à un nombre plus convenable, ou à explorer les sous-espaces.

La détection des données aberrantes (ou *détection d'anomalie*, ou *détection de nouveauté*) est l'étude des observations singulières pour détecter ce qui est normal et anormal. Les méthodes qui exécutent cette tâche sont des algorithmes ou des modèles qui sont basés sur l'utilisation des distributions de données. Différentes familles d'approches peuvent être trouvées dans la littérature, elles sont souvent indépendantes de la vérité terrain. Ces appoches effectuent une analyse des valeurs aberrantes en détectant les comportements principaux de la majorité des observations. Ainsi, les données qui diffèrent de la distribution normale sont considérées comme du *bruit* ou des *aberrations*. Nous nous intéressons à l'application de cette tâche au texte. Malgré les progrès récents dans le traitement du langage naturel, les ordinateurs n'ont toujours pas une compréhension profonde du langage humain en l'absence d'informations. Par exemple, la phrase *"Un sourire est une courbe qui redresse tout"* a plusieurs niveaux de compréhension, et une machine peut rencontrer des difficultés pour choisir le bon niveau de lecture.

Cette thèse présente la recherche de valeurs aberrantes ou d'anomalies en présence

de grandes dimensions, appliquée au texte. Les avancées récentes en matière de détection d'anomalies et de détection de valeurs aberrantes ne sont pas représentées de manière significative avec les données textuelles et nous proposons de mettre en évidence les principales différences avec les valeurs aberrantes à haute dimension. Nous abordons également les méthodes d'ensemble qui sont quasiment inexistantes dans la littérature pour notre contexte. Enfin, nous pouvons voir que l'application de la détection de valeurs aberrantes amène des améliorations sur le résumé de texte automatique par abstraction.

Dans nos travaux, nous proposons GenTO, une méthode qui prépare et génère un fractionnement des données dans lequel sont insérées des anomalies et des valeurs aberrantes. Sur la base de cette méthode, nous proposons une évaluation et un benchmark des approches de détection de valeurs aberrantes avec des documents. En outre, l'apprentissage sans supervision conduit souvent les modèles à se fier à certains hyperparamètres. Par exemple, Local Outlier Factor s'appuie sur les k plus proches voisins pour calculer la densité locale. Ainsi, le choix de la bonne valeur pour k est crucial. À cet égard, nous explorons l'influence de ce genre de paramètres pour les données textuelles.

Alors que le choix d'un seul modèle peut entraîner un biais évident par rapport aux données du monde réel, les méthodes d'ensemble permettent d'atténuer ce problème. Elles sont particulièrement efficaces pour l'analyse des valeurs aberrantes. En effet, la sélection de plusieurs valeurs pour un hyperparamètre peut aider à détecter des valeurs aberrantes fortes. L'importance est alors abordée et peut aider un humain à comprendre la sortie d'un modèle boîte noire. Ainsi, l'interprétabilité des modèles de détection de valeurs aberrantes est remise en question. Nous constatons que pour de nombreux jeux de données, un faible nombre d'attributs peut être sélectionné comme oracle. L'association de modèles complets et de modèles restreints permet d'atténuer l'effet boîte noire de certaines approches.

Dans certains cas, la détection des aberrations fait référence à la suppression du bruit ou à la détection des anomalies. Certaines applications peuvent bénéficier de la caractéristique d'une telle tâche. La détection des spams et des fake news en est un exemple, mais nous proposons d'utiliser les approches de détection des aberrations pour l'exploration des signaux faibles dans les résumés automatique par abstraction. Ainsi, nous observons que les modèles de la littérature aident à améliorer les approches de résumé de texte par abstraction, sans supervision. Ceux-ci permettent également de trouver les signaux faibles dans le texte.

Acknowledgement

In the context of this work and over these past years, there are many people to whom I would like to express immense gratitude. This journey has faced numerous challenges, and yet, I fondly remember all the people who landed their hands to me and honored me with their presence.

Formerly, the work presented here originated in 2017 within a startup which explores artificial intelligence in unconventional contexts. With a specialization in NLP, it is in this context that the use of machine learning to identify weak signals in texts took root. During my time in the company, I had the pleasure of meeting brilliant peoples: Thomas, Yohann, Marie, Maryna, Lucie, and Kevin. Thank you all for the discussions, lunch meals, breaks, and after-work moments. And especially, thank you Kevin for being part of my circle and for being the wonderful person you are. I am immensely grateful for all our discussions on our shared passions, the afterwork drinks, the insightful exchanges on research, video game sessions, and much more, as I wouldn't have enough space to write it all. This work could not have come to fruition without your support, thank you.

During these years at the laboratory, I had the opportunity to join the LFI team at the LIP6 laboratory where I met brilliant, incredible, and caring individuals. I deeply thank Thibault, Arthur, Adam, Clara, Vincent, Milan, Adulam, Garance, Yann, Guillaume, Leandro, Ege, Adam, and Aymeric. Thanks for all the lunch and coffee breaks, the lively discussions on everything and nothing, the after-work gatherings at the laboratory and at Baker Street. You have been a warm light, and I am honored to have met you. Daily life is simpler with you. Moreover, special thanks to Thibault for helping me gain perspective on my thesis work and for being a support all these years, with humor and kindness. I cherish all the talks we have had these past years and deeply thank you for sharing your insights and intuitions. I also want to give a special thanks to Adam; our conversations on strange subjects and unconventional domains made everyday life so much more joyful. You helped me a lot with your questions, changing the perspective and vision of subjects. Also, special thanks to Clara; your kindness and rigor have been an example for me, and I am very grateful for the daily support, your curiosity and your wisdom.

I deeply thank the brilliant and exceptional people of the LFI team with whom I had the chance to exchange and work: Sabrina, Jean-Noël, Louis, Bernadette, Marie-Jeanne, and Christophe. I want to thank you for the exchanges, advice, and knowledge you shared throughout these years. I have deep gratitude for Marie-Jeanne; thank you for being present all these years and especially during the early years. Learning scientific rigor and AI knowledge by your side has been a great honor. You are a mentor model in my eyes, and I am infinitely grateful to you for sharing your curiosity, wisdom, and kindness all these years. My most special gratitude is for Christophe;

I have no words to describe these years under your supervision. You have been an exceptional supervisor, from the beginning to the end. I am profoundly grateful for learning curiosity and scientific rigor by your side. I am also grateful for your patience, kindness, and wisdom. I have learned a lot from you, and you will remain a role model for me. This adventure would not have been possible without you. Your qualities make the LFI team a kind, open minded, and passionate research team, making it a perfect environment for learning. Sorry again for these last-minute submissions. By the way, I am writing this part one day before the final submission, and there is little chance you'll be able to read everything. This will be the last urgent reading I dedicate to you.

Also, I want to thank all my friends who have been the support that anyone would dream of having. Thank you Guillaume, Baptiste, Florian, Hugo, Fred, Clement, Sarah, Nicolas, Oussama, Paul, Benjamin, Lucie, Fiona, Thomas, Mathilde, Leopoldine, Mitch, Roland, Isaure, Alexandre, Aurelien, Jeremy, Jean, Caline, Alienor, and all those I couldn't mention. Having you by my side is a real blessing, and this work could not have come to fruition without your support.

Finally, my dearest thanks are for my family and my wife. In no world and no dimension could these past years have been possible without your love, patience, and kindness. To my brother, thank you for being present in all circumstances and for being the ray of sunshine that you are. To my father, thank you for your patience, kindness, and attention. To my wife, no words suffice to define the support, patience, intelligence, curiosity, and kindness you have shown. This work could not have seen the light of day or come to fruition without you.

Contents

Chapter 1

Introduction

Artificial intelligence has grown spectacularly these past years with the evolution of computer science, computer hardware and internet. Recent artificial intelligence advances are bound to the increasing number of data that industry and research fields produce and have access to. Hence, it can be wrong to admit that artificial intelligence is only referring to processing data and observations. A global view of the field can refers to the ability of the machine to perform and/or imitate cognitive behaviors such as decision making, learning, perceiving and reasoning. Thus, we expect from the machine to perform actions following human reflection and mechanisms, based on given environments or observations.

The impetus for such expectations has recently arisen from the widespread use of the internet by the global population and the concomitant surge in data generation. Machine learning offers a solution to the challenges posed by handling diverse tasks across vast datasets. Its ubiquitous applications are evident in various domains, including fraud prevention (such as credit card fraud, system intrusion, and financial fraud), recommendation systems (spanning web navigation, e-commerce, social media, and entertainment websites), computer vision (encompassing facial recognition, object detection, and image segmentation), healthcare (covering disease detection, identification of cancer cells, diagnosis, and molecular exploration), and marketing (including consumer clustering, user analysis, and audience targeting), among numerous others in fields such as agriculture, automobiles, and human resources. Despite its considerable success across diverse applications, machine learning is not without its significant shortcomings on various levels. While some models are naturally understandable for a human[1], they are often outperformed by more complex models that are referred as *black boxes*. Artificial neural networks are one example of black box model.

Machine learning algorithms vary in complexity, ranging from simple to intricate, but dealing with data can pose similar challenges. Low-dimensional data, typically in 2D or 3D with an intuitively interpretable representation (e.g., the average price of a *baguette* by years), is more straightforward for humans to comprehend than high-dimensional data with thousands of dimensions. While this is not a universal rule, it highlights a primary drawback of machine learning, which is intricately tied to the nature of the data itself. When a model underperforms, several factors may

[1] E.g the resulting form of a trained model. For instance, tree-like structures can be simpler to tackle than n-dimensional vectors (with n being high).

contribute to its shortcomings, including a scarcity of data (as neural networks often excel with ample training observations), non-uniformity in the data (e.g., missing features), the presence of feature constraints (such as the impossibility of negative age), or suboptimal algorithm choices. Examining the relationship between machine learning and data reveals a significant aspect—trained models inherently lack true comprehension. For instance, a model may not inherently understand that a person's age cannot be negative; it merely learns from the data it has been trained on. This lack of true understanding becomes evident when considering that age is fundamentally linked to the time elapsed since an individual's birth. In this context, the emergence of negative age values can result in nonsensical outcomes, especially in applications sensitive to such data anomalies (e.g., post-birth complication data with age around zero). For low-dimensional data, errors can lead to a notable shift away from normal data patterns. However, the impact differs in the case of high-dimensional data.

Addressing these challenges often involves the application of deterministic techniques. In contrast to stochastic models, which lack explicit rules and often operate in a probabilistic manner, deterministic approaches provide more structured and predictable outcomes. While stochastic models are frequently juxtaposed with symbolic approaches that emulate human learning processes, symbolic artificial intelligence offers a distinctive strategy for tackling various tasks. However, within the machine learning context, a specialized application addresses the study of abnormal data: *outlier detection* (or *anomaly detection*). Outlier detection involves scrutinizing observations to discern what is considered normal and abnormal. Methods employed for this task rely on data distributions, encompassing a diverse array of algorithms, methods, or models. Various families of approaches, largely independent of ground truth, conduct outlier analysis by identifying the principal behaviors exhibited by the majority of observations. Consequently, data that deviate from the inlier distribution are labeled as "noise" or "outliers." Remarkably, the application of outlier detection in the realm of text is infrequently explored. Despite notable advancements in natural language processing, computers still grapple with a profound understanding of human language due to the lack of comprehensive information. For instance, the sentence "A smile is a curve that sets everything straight" possesses multiple levels of interpretation, presenting a challenge for machines to accurately discern the intended level of meaning.

Motivation

The intricacies of human language present distinctive challenges for machine learning algorithms, particularly in the realm of identifying outliers within text data. Uncovering these outliers can yield valuable insights, such as recognizing emerging trends, pinpointing fraudulent activities, unveiling hidden patterns, or even identifying potential errors in data collection or labeling. Consider online product reviews as an illustrative example. Outlier detection in this context proves instrumental in discerning fake or spam reviews, ensuring that consumers can make well-informed decisions.

Nevertheless, the contextual nature of language, the presence of multiple layers of meaning, and varying degrees of ambiguity contribute to the complexity and demands of the outlier detection task. Effectively identifying outliers hinges on a comprehensive understanding of the normal patterns and behaviors inherent in text data. This understanding is crucial for navigating the intricacies of language and successfully discerning deviations from expected norms.

As AI-driven applications increasingly impact various facets of society, the identification and management of outliers in text data become pivotal for enhancing the interpretability and explainability of artificial intelligence models. This is crucial not only for building trust with users but also for averting potentially biased or harmful decisions rooted in erroneous outliers. Despite significant strides in natural language processing and machine learning, research and applications related to outlier detection in text data remain relatively unexplored. This thesis aims to bridge this gap by focusing on the analysis of high-dimensional outliers within the context of textual data. Our approach involves leveraging recent advances in outlier detection and anomaly detection methodologies, integrating them with innovative techniques tailored for text data. The goal is to develop robust and more interpretable outlier detection approaches. By delving into the challenges and proposing potential solutions for outlier detection in text data, this thesis aspires to contribute to the broader vision of constructing AI systems that exhibit greater human-like understanding, reasoning, and decision-making capabilities.

Research objectives

As explained previously, the principal objective of the thesis is to tackle outlier detection with text data. Achieving such goal requires special attention, and several research questions to investigate have been focused on.

Problem 1. Outlier detection in text data, a formal definition One of the main research objectives is to explore the challenges and opportunities of outlier detection in high-dimensional text data. Textual data present several levels of study (syntax, semantic, ...) and it can be confusing to compare different applications with each other because of this characteristic. Literature on high-dimensional data is rich and presents numerous successful approaches that can be compatible with textual data. We aim to understand how the high dimensionality of textual features impacts the performance of existing outlier detection techniques and whether specialized approaches are required to handle such complexity effectively. Among recent surveys and overviewing works of the outlier detection literature and anomaly detection literature, an attention is often taken regarding textual data but dedicated and comprehensive surveys does not exists for tackling outlier detection with text data. Inherently to such concerns, because textual data are a special kind of high-dimensional data, the related taxonomy for outliers is different from other kind of data. As a consequence, three research questions should be answered:

1. *What is an outlier in the context of textual data ?*
2. *Considering outlier detection with high-dimensional data, what kind of requirements are needed to introduce a definition for textual outliers? Do they share any similarities with common outliers, or are they fundamentally different?*
3. *Are reference methods to detect outliers efficient when applied to text ?*

Problem 2. Evaluation and real-world data, an experimental problem Based on the investigation of high-dimensional outliers in text data, novel outlier detection algorithms should not only consider the unique nature of text data but also demonstrate improved performance compared to traditional approaches. Thus, the issue of conducting experimental study has to be deepen in this context. Browsing reference surveys of our context, it appears that reference works for outlier detection in text are often proposing different experimental settings to evaluate their approaches. As a consequence, several questions blossoms:

1. *Availability of corpora is an occurring challenge in machine learning, is it similar in the textual outlier detection context ? How such corpora should be built and used ?*
2. *Considering the lack of global comparative works, are state-of-the-art approaches exploring and defining the same kind of detection problem ?*

Problem 3. On the lack of diversity for existing methods In our context, lack of comprehensive study of the field is a problem, resulting in a poor diversity of approaches according to state-of-the-art approaches. Popular methods performing outlier detection in text data are mostly tackling the task through text representation with either dimension reduction techniques or text representation improvements. This phenomena raises several questions we aim to answer in this thesis:

1. *Are state-of-the-art approaches achieving a comprehensive or sufficient comparative study with their results ?*
2. *Is there any model enabling to perform global experimental protocols ?*
3. *Are there some kind of outlier detection approaches in the literature not already exploited in our context ? If there are, how do they perform with text corpora ?*

Problem 4. Application impact of textual outlier analysis In some applications, outlier detection results might trigger critical decisions or human interventions. One interesting focus concerns how outlier detection methods can influence decision-making processes and the extent to which their interpretability helps human understanding and trust. The issue of interpretability in machine learning models is critical, particularly in applications where human decision-making is involved. Exploring methods to enhance the interpretability and explainability of outlier detection

models for text data can enable users to understand the rationale behind outlier classifications, leading to two main questions:

1. *What kind of applications could benefit of outlier detection in text ?*
2. *What is the impact of such study for users ?*

Roadmap

This thesis delves into the analysis of outliers in the context of text data and proposes innovative approaches to address the scientific challenges discussed in the previous section. Notably, recent advances in anomaly and outlier detection lack substantial representation in the realm of text data, and our work aims to underscore the key distinctions. Our contribution begins with a comprehensive examination of outlier analysis in textual data, featuring a comparative study and the introduction of the GenTO algorithm. This contribution addresses the initial challenges by presenting a taxonomy for textual outliers and conducting an experimental analysis of high-dimensional methods. Additionally, we explore potential extensions to augment the availability of corpora.

Subsequently, we shift our focus to providing an overview of the field while maintaining a tangible connection to traditional outlier detection and a close alignment with anomaly detection literature. Our experimental study reveals the surprising efficacy of traditional literature in a textual context.

To address the issue of method diversity, we conduct a study and extend ensemble methods, which remain largely unexplored for text data. Our research also investigates recent language models and incorporates interpretability within our context. The exploration of applying outlier detection techniques to ensemble methods demonstrates the potential for extending outlier analysis in text to other applications within the natural language processing field. We particularly delve into abstractive summarization, an intriguing textual application that incorporates a diverse array of techniques. Employing robust text representation through outlier analysis enhances the performance and reliability of such models.

Structure of the thesis

In the pursuit of addressing the challenges associated with detecting outliers in a textual context, as outlined earlier, the initial Chapter 2 serves to introduce the domain. This chapter elucidates the key concepts underpinning the thesis and synthesizes insights from machine learning, data mining, and outlier analysis. With a foundation in these notions, the chapter transitions to the specific challenge of performing outlier detection with text data. Notably, we aim to bridge the gap between text data and other data types for outlier detection, leveraging insights from the burgeoning literature on other data types.

Chapter 3 narrows the focus to the distinctive challenges of performing outlier detection with text data. A comprehensive overview of outlier detection techniques dedicated to textual data is provided. The chapter's contribution lies in unifying outlier detection in text with other data types through the proposition of a generic setup applicable across the literature. We define what constitutes an outlier in the textual context, explore various outlier detection approaches suitable for text, and present an evaluation of these approaches through an experimental study.

Expanding our research in Chapter 4, we delve into ensemble learning and data fusion techniques for outlier detection in text. The concept of outlier ensembles is explored for their potential benefits in enhancing detection performance. Our contribution in this chapter is the introduction of an ensemble autoencoder approach tailored for effective handling of textual outliers. New features for outlier detection in text are introduced, and the interpretability of the proposed methods is discussed. To showcase the practical applicability of our formalism, we apply outlier ensemble techniques to the real-world problem of abstractive summarization of text.

Chapter 5 introduces the application of our outlier analysis techniques to abstractive summarization. We focus on the utilization of neural networks for abstractive summarization and discuss unsupervised text summarization methods. By incorporating our outlier analysis methods, we aim to achieve robust and reliable abstractive summarization results. An evaluation of our improved summarization approach is provided to showcase its efficacy in handling outlier text data.

The concluding chapter (Chapter 6) synthesizes the main contributions of this thesis and provides a comprehensive discussion of the results obtained throughout our research. It highlights the practical implications and potential future directions in outlier detection with text data. This chapter serves as a conclusion to our work, offering insights into the broader scope of applying outlier analysis techniques to various domains. It concludes with a reflection on the contributions and proposes further exploration and future research.

In summary, this work aims to present a comprehensive and coherent analysis of outlier detection with text data, along with its potential applications and extensions. Each chapter contributes to the overarching goal of bridging the gap between outlier detection in text and other data types, providing novel insights and opening new avenues for research.

Chapter 2

Study of outliers: background knowledge

As introduced in the introduction, outlier analysis brings together several tasks that enable knowledge extraction from outlying observations. This chapter is particularly important for the rest of the thesis as it provides a comprehensive overview of required knowledge to become familiar with outlier analysis across various applications and types of data. Within different applications, outlier detection is relevant for either removing noise or improving prediction robustness. Outlier detection is particularly effective for explaining normality (scope of what is expected) and machine learning decisions.

Study of outliers involves expressing through a formalism what normality is, in our context, i.e. machine learning. Obviously, once the notion of normality has been established, it becomes easier to formalise the notion of abnormality. However, once this formalism has been defined, normality can also vary depending on the context, the data and even the type of approach. In our work, we make a distinction between several abnormalities in order to prioritize and differentiate outliers from each other (noise is different from an anomaly, the same goes for a typo and a grammatical error). There are also several methods for analyzing and detecting aberrations. These can be fundamentally different in several key aspects, for example logic rules require different attention than neural networks. Because of the diversity of approaches and types of outliers, the evaluation step (manual or automatic) makes it possible to compare, rank and explain results of experimental protocols on outliers.

Thus, in this chapter we present a view of outlier analysis domain through the study of comprehensive works and surveys of the literature. Common knowledge and main notations of machine learning and classification are presented in Section 2.1. Principal goals and motivations of outlier analysis are recalled in Section 2.2. The Section 2.3 defines what is an outlier and compare its definition against different application level. It also compares the neighboring terms that can be associated to several outlier analysis characteristics. A taxonomy of outliers is proposed in Section 2.4 with further explanation about proximity between anomaly and outlier. Methods that perform outlier detection are based on a particular techniques, they are introduced in Section 2.5. The question of performing benchmarks with protocols and evaluation for approaches of the literature is proposed in Section 2.6.

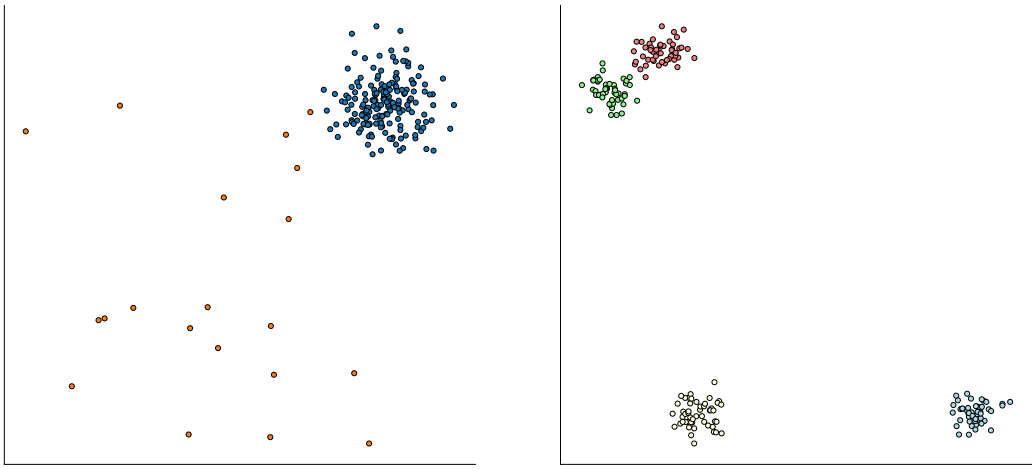


FIGURE 2.1: Binary scenario (left) and multiclass scenario (right).

2.1 Learning from observations

The thesis is primarily devoted to textual data and tackles several aspects from natural language processing. Though this type of data is unique to work with, it is nonetheless data from which it is possible to learn from. In this section, we introduce the different notations and the context of machine learning. Because outlier detection is often performed through binary classification, and more precisely one-class classification (D. M. J. Tax, 2001), we also introduce our notations in such a context. In supervised machine learning, the classification task can be generalized as the *algorithmic categorization of instances* where the principal purpose is to assign a class, or category, to each instance to be classified.

2.1.1 Dataset, features, labels and classes

Let X be a set of N *instances*, $\mathrm{X} = \{\mathrm{x}_1, ..., \mathrm{x}_N\}$. An instance x_i refers to data, observation or point. An instance is described by one or more input variables called *features*. A simple problem takes instances with a small set of features, but more complex problems can involve thousands or millions of features. For all $\mathrm{x}_i \in \mathrm{X}$, with $\mathrm{X} \subseteq \mathbb{R}^{N \times D}$, where D is the number of features, we note $x_{i,j}$ (or x_{ij}) the value of the j-th feature of instance x_i.

$$\mathrm{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,D} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,D} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,D} \end{pmatrix} \tag{2.1}$$

Classification commonly involves machine learning with supervised, semi-supervised

or unsupervised learning. For supervised and semi-supervised learning, a target prediction, label or class, can be associated to each instance. Let $\mathcal{Y}$ be a set of labels associated to X, $(\mathrm{x}_i, y_i) \in \mathrm{X} \times \mathcal{Y}$ is a labelled instance. In the case where labels are available, each x_i is associated with a label $\dagger_i$:

$$\mathcal{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \tag{2.2}$$

The set of labels $\mathcal{Y}$ depends of the application. For binary classification:

$$y_i \in \{-1, +1\} \tag{2.3}$$

In this case we face a problem with two possible outcomes, and represents a task that is often observed. Another type of problem involves classifying instances into one of three or more classes. In such cases, the label y is a value that contains K values, it refers to multiclass classification.

Figure 2.1 displays two scatterplot presenting binary classification and multiclass classification (four exactly). In addition of the classification type, the Figure 2.1 shows two different distributions of classes: balanced and imbalanced. An imbalanced dataset refers to a kind of dataset where the number of observations for one or more class greatly differs from one or more other class. Such situation is not unique and can be addressed with several strategies.

2.1.2 Machine learning

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms, models or agents that enable computer systems to learn and make predictions or decisions without being explicitly programmed. Machine learning refers to the description of agents who can improve their behavior (and/or performance) through diligent study of their own experiences. There exist several kinds of labels which are often related to the application they are associated with. Features from instances and labels can be either quantitative or categorical. The first is often continuous or near-continuous. For categorical variables, it depends of a discrete set of groups (i.e. the gender of a person or the color of a fruit, ...). Categorical variables can also be numeric values that represent a hierarchy, order or any other organized structure. In machine learning, the reference task with quantitative label is named regression and classification with categorical target.

Within supervised learning, many problems and tasks can be written mathematically in the form of optimization (Eisenstein, 2019):

$$\widehat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\ \Upsilon(\mathrm{x}, y; \theta), \tag{2.4}$$

where Υ is a scoring function such as $\Upsilon_\theta : \mathrm{X} \times \mathcal{Y} \longrightarrow \mathbb{R}$. θ is the parameter vector of the function Υ. Following Equation 2.4, the output maximize the scoring process.

Convention and notations can differ in the literature for different applications. The scoring function is also named model. We note that labelling an instance x can be written $\mathcal{Y}(\mathrm{x})$.

With Equation 2.4, a simplified view of the learning process consists of finding parameters θ. This step is usually performed with a training dataset X^{tr} (it is often a subset of X). There exist two other kinds of dataset: the validation dataset X^{vl} often used to compare multiple models built on the same training dataset with different θ, and the testing dataset X^{ts} used to evaluate the final model.

2.1.3 Learning without supervision

We assume that a training set X^{tr} possesses observations x_i that are associated with a label y_i. However, learning without labelled data is possible and refers to unsupervised learning. Unsupervised learning methods, in contrast to supervised learning, focus on understanding data distribution, extracting characteristic features, and uncovering underlying structures. These methods operate without relying on labeled data, allowing them to discover unknown patterns within the data. By not being constrained by the limitations and biases of labeled data, unsupervised learning approaches can estimate important features and offer the advantage of easily accumulating information without manual intervention. However, it is important to note that despite these benefits, unsupervised learning may exhibits lower accuracy and can be more dependent on the chosen model. In cases where the data types are challenging, supervised learning often provides a more straightforward approach, enabling the learning of complex patterns and structures.

Learning from distributions Furthermore, in supervised learning, where labeled data is used, the learning process heavily relies on the correctness and quality of the provided labels. However, labeled data can sometimes be limited in quantity or suffer from biases introduced during the labeling process. These limitations and biases can affect the performance and generalizability of the learned models (Bengio, Courville, et al., 2013). On the other hand, unsupervised learning methods operate without the need for explicit labels. Instead, they focus on understanding the underlying structure and patterns within the data itself. This freedom from labels allows unsupervised learning algorithms to estimate and identify important features directly from the data, without being influenced or biased by predefined labels (Hinton and Salakhutdinov, 2006; Le, 2013). By estimating important features, unsupervised learning can uncover hidden relationships, identify clusters or groups within the data, detect anomalies, or reveal latent factors that may not be apparent in the labeled data[1].

Challenging data When the data types present inherent complexities or pose specific challenges, such as high dimensionality, noise, or ambiguity, supervised learning

[1]This ability to extract valuable features directly from the data contributes to the exploratory nature of unsupervised learning, where it can uncover insights and uncover previously unknown patterns that may be overlooked in supervised learning scenarios.

methods often offer a more straightforward and reliable approach. In challenging data scenarios, having access to labeled data can provide valuable information and guidance to the learning process (Hastie et al., 2001). The presence of explicit labels helps in disentangling complex patterns, reducing uncertainty, and allowing the model to make more accurate predictions (Bishop, 2006). By explicitly knowing the desired output for each input, supervised learning algorithms can directly optimize their performance to minimize errors and align with the provided labels. While unsupervised learning has its own merits and advantages, it may face difficulties in dealing with challenging data types since it lacks the explicit supervision and guidance provided by labeled data.

Considering that Equation 2.4 describes a supervised learning scenario, we write an unsupervised scoring model is usually denoted as:

$$\Upsilon_\theta : \mathrm{X} \longrightarrow \mathbb{R} \tag{2.5}$$

Such model highly depends of its parameters θ. In addition to its parameters, the success of a model depends also of the underlying characteristics of the data and the learning paradigm. Regarding the latter, we observe several approaches that belong to different families such as clustering, latent variable learning or association rules.

2.2 Outlier analysis

Outlier detection is a fast-growing field that concerns numerous domains and applications, although it is not a brand new research topic (Abraham and Box, 1979; Hawkins, 1980). A great number of contributions were achieved by the Statistics community and led to mathematically more precise methods with a simplified view of data representation (Rousseeuw and Hubert, 2011). With the recent trends in data mining, topics such as algorithmic description and interpretability are now getting more attention (C. C. Aggarwal, 2017a). Recent works also focus on definition of methods that are dedicated to much more complex data representations such as text and images. With the emergence of data mining, the field of outlier analysis has grown along with various communities and different application areas (time series, financial systems, information systems, ...) (C. C. Aggarwal and Yu, 2001; Prastawa et al., 2004; Basu and Meckesheimer, 2007; Blazquez-Garcia et al., 2021).

Study of outliers can be motivated by several needs or questions at different steps of a problem. The elimination of outliers is often desired during the data cleaning or preprocessing step. In this case, outliers are not different than noise in the same way as anomalies. Another motivation of outlier analysis lies in helping models to be more robust to noise and rares observations. Alternatively, outlier awareness can be integrated to a model beforehand through dedicated techniques, leading to more robust[2] approaches. In the context of outlier detection and machine learning, robustness refers

[2]We note that usage of the term *robust* or *robustness* is defined differently depending of the task and domain.

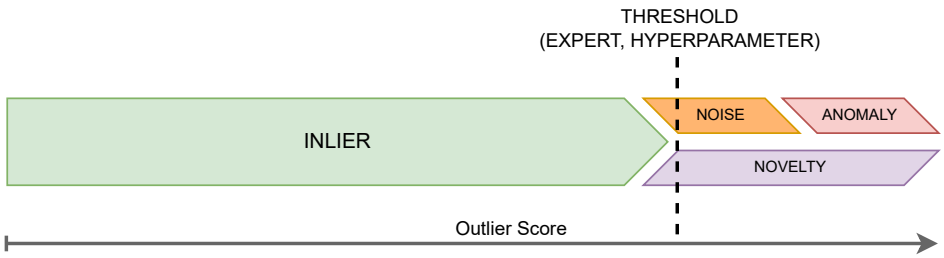


FIGURE 2.2: A one class-classifier learn the normal distribution and provides an outlier score. Depending of the selected approach, or of the expert, the instances are considered outliers from a certain threshold. Furthermore, an outlying instance can be either noise, anomaly or also novel considering the value of the score (high score means abnormal).

to the ability of a method or model to maintain its performance and reliability even when faced with challenging or anomalous data points, including outliers (Hawkins et al., 1984; Rousseeuw and Leroy, 1987; Zimek, Schubert, et al., 2012). A robust outlier detection method should be resilient to the presence of outliers and should not be unduly influenced by them, ensuring that the detection results remain accurate and consistent.

Figure 2.2 (adapted from C. C. Aggarwal, 2017a) describes how a model can considers an outlier score. Although noise and anomaly both refer to an intensity of normality, novelty is a special kind of observation. The latter refers to a new generation of data, whether normal or aberrant. In the context of outlier detection and machine learning, distinguishing between noise, anomaly, and novelty is important. While noise is typically considered as unwanted variation, anomalies can provide valuable insights or indicate important events or patterns in the data. Novelty, on the other hand, represents new and previously unseen data that may require specific handling or treatment (Markou and Singh, 2003; Pimentel et al., 2014). The statement emphasizes that novelty encompasses both normal and aberrant instances, highlighting the idea that encountering previously unseen data, regardless of its nature, is considered novel.

2.3 What is an outlier ?

This section delves into the exploration of outlier definition in various scenarios. Initially, we introduce the commonly adopted definition of an outlier based on its lexical interpretation. Subsequently, we compare this definition with that of anomaly. The motivation behind this comparison stems from the prevailing confusion that considers an anomaly to be synonymous with an outlier. Building upon this initial analysis, we proceed with a comprehensive comparison of the diverse definitions proposed in the existing literature. Finally, we engage in a discussion aiming to derive a conclusive and applicable definition in the context of this research.

2.3.1 Outlier and anomalies: general definitions

Depending on the field, the nomination of an abnormality can be done in several ways, such as *anomaly* or *outlier*. These terms share a semantic relationship with *normal* (or normality) but can not be used without a proper understanding of their differences. The common definition of an outlier is:

Definition 2.3.1 (Outlier). *Something (such as a geologic feature) that is situated away from or classed differently from a main or related body*[3].

The interesting part of this definition is that an outlier can only be expressed with regards to a group of individuals (related body). There is also a conflict between what is relevant and what is less, i.e. *classified differently from a main or related body.* Works of Statistics community like Hawkins (1980) have used the term of outlier in order to describe observation that *wrongly occurs* in a distribution. Another common definition for this domain is:

Definition 2.3.2 (Statistical outlier). *A statistical observation that is markedly different in value from the others of the sample.*

The difference with this definition is the notion of metric and intensity (*markedly*). Such metric aims at computing whether the observation is outside, to a certain degree, of expectations (distance, similarity, ...). Comparing both definitions, an outlier, opposed to *inliers* (normal observations), is an observation that does not follows the same criterion of *normality* than other data. Depending of both application and structure of data, the notion of normality is different. For instance, expecting to find Football and Tennis news in a sport media is normal but Astronomy news is not normal in this media. At another level, the expectation can be done with searching Football news, and finding instead Tennis news.

Identification of outliers can be similar to finding rare items that should not appear, or also looking for anomalies. It is interesting to note that the literature also use the term of *anomaly* when processing this task. The definition of anomaly from the same source as Definition 2.3.2 is:

Definition 2.3.3 (Anomaly). *Something different, abnormal, peculiar, or not easily classified: something anomalous.*

The comparison with the definition of an outlier leads to conclude that finding an anomaly requires, obviously, the description of normality. Thus, the main difference between an anomaly and an outlier is to be the definition of a group of individuals. **Outlier is a term that strongly induce a group of individual, while anomaly does not necessarily.** For example, the rules of a game dictate how the game should be played, but if a player performs a move that takes advantage of a flaw in the rules, it can be considered as an anomaly. Such anomaly will not necessarily depend of a group of observation to be abnormal. Both terms are often used interchangeably in

[3]The definition is extracted from the *Merriam Webster* dictionary.

data mining and Statistics. A brief review of these definitions defines an anomaly as an observation that deviates from an expectation of normality and an outlier as an instance that is observably different.

2.3.2 Towards data mining

In the previous section we observe two different definitions for outlier: usual and statistical. Considering statistical works on outlier, Hawkins (1980) has proposed an intuitive definition:

Definition 2.3.4 (Statistical outlier (Hawkins, 1980)). *An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.*

The notion of *mechanism* is important, it refers to a kind of scenario where outliers occur. Two mechanisms are described:

- Instances come from a distribution where any observation is in any way erroneous.
- Instances come from two distributions: *basic distribution* and *contaminating distribution.*

For generalization purpose, information theory notations can be vastly used for numerous applications involving data distributions. Based on this work, we introduce the following definition for the data generating probability:

Definition 2.3.5 (Data Generating Probability). *Let $\mathcal{X}$ denote the space of possible observations. The data generating probability function $P(\mathbf{x};\boldsymbol{\theta})$ assigns a probability to each observation $\mathbf{x} \in \mathcal{X}$, parameterized by $\boldsymbol{\theta}$. It represents the underlying probability distribution from which the data is generated.*

Considering that the latter mechanism is more often studied, Definition 2.3.4 implies that there exist a probability distribution for inliers and a probability distribution for outliers that we define as:

Definition 2.3.6 (Inlier/Outlier Distribution). *Let* X *be a random variable representing the observations. For any observation* $\mathrm{x} \in \mathcal{X}$*, we define a statistical model* $\mathrm{P}(\mathrm{x};\theta)$ *to assign a probability to* x *according to parameters* θ*. The data generating process represents the causal mechanism from which the data originate. We define* $\mathrm{P}_{in}(\mathrm{x};\theta)$ *as the probability distribution of inliers among* $\mathcal{X}$ *and* $\mathrm{P}_{out}(\mathrm{x};\theta)$ *as the probability distribution of outliers among* $\mathcal{X}$*.*

For a discrete random variable, the probability distribution can be defined using a probability mass function (PMF). The PMF $P(\mathrm{X} = \mathrm{x})$ gives the probability of X taking the value x. For a continuous random variable, the probability distribution can be defined using a probability density function (PDF). The PDF $p(\mathrm{X} = \mathrm{x})$ specifies the relative likelihood of X taking the value x. While Definition 2.3.5 and Definition 2.3.6

are needed for tackling very recent works, they do not represent the only way to formalize outliers. However, they succeed to connect the notations from former and latter reference studies that are furthermore presented.

In a more contemporary context, Hodge and Austin (2004) have put forward the definition initially suggested by Grubbs (1969). The focus of both works lies in exploring the characteristics of outliers to deviate from inliers with respect to their membership attributes:

Definition 2.3.7 (Outlier (Grubbs, 1969)). *An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs.*

In this study, the occurrence of outliers can be attributed to a range of factors, including human error, instrument error, natural variations within populations, fraudulent behavior, and various other causes. It can be stated that the types of outliers are diverse and numerous, with each application potentially presenting unique instances. Based on Definition 2.3.6, a membership function can be employed to evaluate the extent of membership in the context of normality. The membership function provides a way to assess the conformity of each data point to the expected behavior defined by the data generating process. Often, a threshold $\tau \geq 0$ such that a probability under P_{in} is small enough to be considered in the a low probability region (we develop it next with Definition 2.3.10). Such a function plays a crucial role in characterizing the concept of normality within the dataset.

The field of data mining has witnessed significant advancements, leading to notable improvements in the study of data. In statistical contexts, the term "outlier" has often been employed interchangeably with *anomaly*. Noteworthy contributions in the area of anomaly detection (AD), such as Chandola et al. (2009), have provided a specific definition of an anomaly as follows:

Definition 2.3.8 (Anomaly (Chandola et al., 2009)). *Anomalies are patterns in data that do not conform to a well defined notion of normal behavior.*

Hence, a distribution of observations is expected to conform to a state of normality or exhibit normal behavior. Any instance that deviates from this expectation is considered an anomaly. Chandola et al. (2009) assert that the nature of input data plays a crucial role and determines whether instances within a dataset exhibit any relationship. Consequently, various types of anomalies can manifest in this context. Definition 2.3.8 introduces a novel concept that diverges from Definition 2.3.7 by incorporating the notion of a pattern. This definition aligns more closely with data mining applications, where normality can be defined and identified as a specific pattern.

While anomalies and outliers exhibit certain similarities, particularly with their characterization of normal behavior. Traditionally, the definition of an outlier is typically outlined by researchers in the introductory sections of their studies. However, J. Zhang (2013) present a contrasting perspective. They advocate for tailoring the definition of an outlier to each specific application, employing a dedicated taxonomy.

A common observation across various studies is that an outlier is characterized by its substantial deviation from the remaining data points. In a more recent contribution in data mining, C. C. Aggarwal (2017a) proposed the following definition:

Definition 2.3.9 (Outlier (C. C. Aggarwal, 2017a)). *An outlier is a data that is significantly different from the remaining data.*

In Definition 2.3.9, the concept of *similarity* is introduced with usage of term *different*, distinguishing it from other approaches that typically emphasize the notion of distance. Consequently, an outlier is defined as an observation that encompasses both the possibilities of being an *anomaly* or *noise*. In Schubert et al. (2014), there is a formal distinction between noise and anomaly, considering the locality of outliers. As per C. C. Aggarwal (2017a), *an "anomaly" refers to a special kind of outlier that is of interest to an analyst.*

Recently, Ruff, Kauffmann, et al. (2021) have proposed a definition of an anomaly that is similar to Definition 2.3.4: an outlier is similar to an anomaly but is considered rare/unique:

Definition 2.3.10 (Anomaly (Ruff, Kauffmann, et al., 2021)). *An anomaly is an observation that deviates considerably from some concept of normality.*

In addition, they propose a formal conceptualization relying on the principles of Probability Theory and Information Retrieval: a concept of normality is a distribution $\mathbb{P}^+$, on a specific data space $\mathcal{X}$, it represents the ground-truth law of normality for a task or application. An anomaly is an observation that deviates considerably from such law. A distinction is proposed between the three terms *anomaly*, *novelty* and *outlier* which refer to observations from low probability region of $\mathbb{P}^+$. Thus, an anomaly is a point from a distinct distribution other than $\mathbb{P}^+$ (generated from another process than inliers). An outlier is a rare observation from the low probability region of $\mathbb{P}^+$, and novelty is an observation of some new region of $\mathbb{P}^+$. Differently from our Definition 2.3.6, Ruff, Kauffmann, et al. (2021) notes an anomaly as follows:

$$\mathcal{A} = \{x \in \mathcal{X} | p^+(x) \leq \tau\}, \tau \geq 0 \tag{2.6}$$

which assumes that $\mathbb{P}^+$ has a corresponding PDF $p^+(x)$ and τ a threshold such that the probability of $\mathcal{A}$ under $\mathbb{P}^+$ is sufficiently small. Such formalism is conveniently fitting a wide range of applications, either being an anomaly or an outlier.

2.3.3 Discussion

The definition of an outlier has traditionally been clear within statistical scenarios. However, with recent advances in complex data such as images and text, new definitions are needed. Terms like *anomaly* and *novelty* are often used to refer to observations similar to outliers. A recent definition proposed by Ruff, Kauffmann, et al. (2021) unifies the notions of anomaly and outlier, which can be compared with the definition presented by C. C. Aggarwal (2017a). This comparison reveals a clear

	Outlier	Anomaly	Statistic	Data Mining	Information T.	Distance	Normality	A/O Disambiguity	Noise Hierarchy	Novelty Identification
Hawkins (1980)	✓		✓			✓				
Hodge and Austin (2004)	✓			✓		✓				✓
Chandola et al. (2009)		✓		✓			✓		✓	
J. Zhang (2013)	✓			✓			✓			
C. C. Aggarwal (2017a)	✓		✓	✓		✓	✓	✓	✓	✓
Ruff, Kauffmann, et al. (2021)		✓	✓		✓		✓	✓	✓	✓

TABLE 2.1: Reviewing table of the definition of an anomaly and outlier according to the literature. The table is separated in four categories: work subject, reference formalism, principal criteria of the definition and additional highlighted features. A/O disambiguity refers to the distinct definition of the difference between anomaly and outlier.

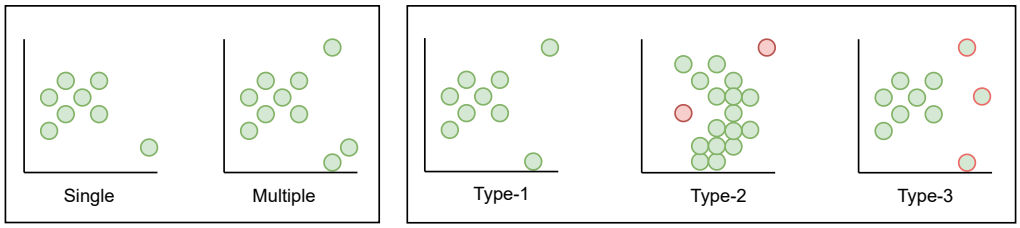
similarity between anomaly and outlier. An outlier is characterized by its rarity and uniqueness, making it a special case of an anomaly.

Through this research, it becomes evident that an outlier can be viewed as an anomaly or as an observation that deviates from the normality. The concept of distance, originally inherent in the etymological definition of an outlier, has evolved into a notion of difference. While the statistical community primarily associated the term outlier with static instances, its meaning has evolved in the data mining community, aligning more closely with the definition of an anomaly. Consequently, it is not uncommon to find works that use these two terms interchangeably. An outlier is characterized by its rarity as an occurrence that deviates from the rest of the group, while an anomaly is defined by its differentiation from other instances within the same group.

The distinction between anomaly and outlier lies in the context in which they appear. An anomaly cannot be explained through the analysis of the base distribution. It occurs when the assumptions of normality are correct, but the appearance of such an observation is deemed impossible. On the other hand, an outlier is an event that can be explained according to the base distribution, regardless of whether it is an anomaly or not. Furthermore, an anomaly is not necessarily restricted to the rest of the observations, unlike an outlier.

2.4 Different kinds of outliers

In Section 2.3 we have described what an outlier can refers to. Also, several terms like anomaly or noise are often associated to an outlier, in practice. While anomaly and outlier are different, as introduced in Section 2.3.3, it appears that in practice



(A) Hawkins (1980) (B) Hodge and Austin (2004)

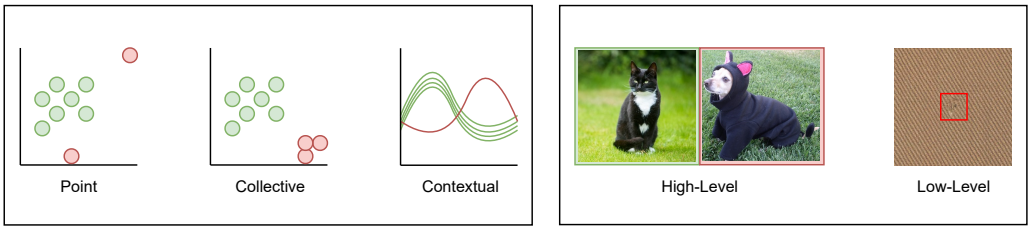
FIGURE 2.3: Taxonomy of outliers for Hawkins (1980) (left) and Hodge and Austin (2004) (right). Hawkins proposes unsupervised methods on static distribution through hypothesis. Hodge and Austin separate outliers in three types where type-2 is aware of outliers and type-3 is aware of what is normal.

they can be interchangeably employed. This section carries the purpose of detailing the different kinds of anomalies and outliers.

In Section 2.3, a compatible definition of an outlier can be found in early works from Statistics. We propose to study in Section 2.4.1 the taxonomy from Hawkins (1980) as a reference for early works. While it provides an easy-to-apply taxonomy, appearance of numerous methods in machine learning and data mining has motivated the definition of a more complete and dedicated taxonomy. Thus, we study in Section 2.4.2 the proposed outlier taxonomy of Hodge and Austin (2004) which is dedicated to more recent needs. From this point, we note several works that aim to introduce a proper taxonomy for outliers and anomalies. We propose in Section 2.4.3 a study of different kinds of anomalies in the recent years. The Section 2.4.4 propose an analysis of the recent taxonomy of outliers in recent works. Several similarities can be illustrated with both taxonomy of anomalies and outliers, we propose a comprehensive comparison in Section 2.4.5.

2.4.1 Outliers: a statistical consideration

One of the main concern of outlier analysis according to Hawkins (1980) is a scenario where a distribution of independent data has only one outlier. He names it *single outlier* and follow his study with *multiple outliers* (Figure 2.3a). Considering this context, such taxonomy takes place where outlier detection is applied with univariate or parametric methods on controlled scenarios (static or artificial distributions). The principal drawback of such taxonomy arises when there is not a known underlying distribution of instances. In this case, non-parametric and/or multivariate methods are required. Another kind of outlier can be introduced when discrete distributions are considered (Ben-Gal, 2005). Thus, the required conditions of the presence of an outlier are unknown and attributes hypothesis is needed. This statement is reminiscent of the condition discussed in Section 2.3.3 for differentiating an anomaly from an outlier. With such kind of scenario, an anomaly is different of an outlier. An outlier is an outlying instance that deviates considerably from most of the distribution.



(A) Chandola et al. (2009) (B) Ruff, Kauffmann, et al. (2021)

FIGURE 2.4: Corresponding contributions from Chandola et al. (2009) (left) and Ruff, Kauffmann, et al. (2021) (right). The first contribution defines the three well-established kinds of anomaly (point, collective and contextual), while the second extends the former taxonomy with two new types.

2.4.2 Outliers with data mining methods

According to Hodge and Austin (2004) the choice of a method must be motivated by the characteristics of the data. They introduce an intuitive categorization of these methods by grouping them according to the learning approach: Type-1, Type-2 and Type-3 with respectively unsupervised, supervised and semi-supervised techniques. The Type-1 approach assumes that data distribution is static and outliers are located in the most remote parts of the distribution. Intuitively, the hypothesis is that anomalies are naturally separated from "normal" data. Type-2 approaches consider supervised classification where a distribution of normal data can be subdivided into distinct classes. The last type of approaches, Type-3, refers to novelty detection or methods that focus on learning inliers only.

Figure 2.3b illustrates these two approaches for outlier identification. The most common situations are those where information about outliers are missing. While Hodge and Austin consider a first kind of outlier that gather all unsupervised methods, they introduce an applicable extension for data mining with Type-2 and Type-3.

2.4.3 Kinds of anomalies

Data mining has quickly converged on the task of anomaly detection instead of outlier detection. The growth of these related tasks led to the definition of a common formalism which is still actively applied. It is intimately related to the popularity of common data sets such as fraud, spam, ... Similarly to Hodge and Austin (2004), contributions of anomaly detection are highly dependent of available data and applications. Afterwards, Chandola et al. (2009) have identified three kinds of anomalies:

1. *Point anomalies* are observations that fall outside the boundaries of normal regions. It is the simplest type of anomaly and it is the focus of majority of research on anomaly detection.

2. *Contextual anomalies* rely on the context that is induced by the structure of the data. Contextual anomalies are then dependent on the problem formulation and can be defined with contextual attributes and/or behavioral attributes.

3. *Collective anomalies* represent observations that belong to a sequence or spatial data (ex. vehicular traffic). Point anomalies can be found in any data set, but for collective anomalies, relationships between instances are required.

Furthermore, the difference between contextual and collective anomalies is the availability of context attributes in the instances. Following this taxonomy, five types of anomaly are considered by Ruff, Kauffmann, et al. (2021): *point anomaly*, *conditional (contextual) anomaly*, *group (collective) anomaly*, *low-level sensory anomaly* and *high-level semantic anomaly*:

1. Point anomaly is an individual anomalous observation. It is the most commonly studied type in the literature.

2. Conditional, or contextual, anomaly is an anomalous observation considering a context such as time or space.

3. Group, or collective, anomaly is a set of related observations that together are anomalous. The notion of relationship often implies that group anomalies tend to be also contextual.

4. Low and high refers to the degree of hierarchy between the features. Low-level sensory anomaly belong to observations associated with normal features with few variations (e.g. artifacts in a picture or texture defects).

5. High-level semantic anomaly describe a kind of anomaly that is dependent of specific and dedicated representation.

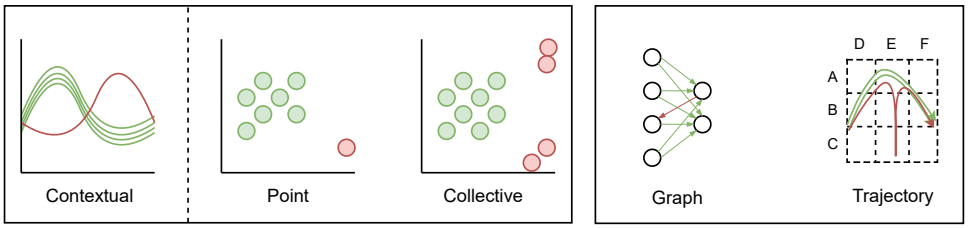
Figure 2.3b illustrates the three principal kinds of anomaly that can be found in literature. They are also connected to outlier taxonomy which can be observed on various references (J. Zhang, 2013; C. C. Aggarwal and Sathe, 2015; Fouché et al., 2020). We detail furthermore in Section 2.4.4 those similarities. Additionally we note that both the high-level anomaly and the low-level anomaly (Figure 2.4b) have a close relationship with the contextual anomaly. One possibility is to consider them both as subtypes of contextual anomalies.

2.4.4 Taxonomy of outliers

For J. Zhang (2013), an outlier is applied to a set of instances and its definition depends on it. First, he describes two kinds of outliers:

1. *Point outliers* which is the simplest to analyse and the most studied. It corresponds to an observation that deviates significantly from the rest of the data. Such kind of observation is considered an outlier regarding its unique characteristic rather than with remaining inliers.

2. *Collective outliers* collective outlier which is a subset of data that deviates significantly from the rest of the instances: an element of this set can be normal, but considered an outlier according to its membership to a group of anomalies.



(A) Principal outlier taxonomy (B) Application outliers

FIGURE 2.5: While J. Zhang (2013) has first introduced a taxonomy that integrates single/multiple outliers (left) with application outliers (right), it lacks the definition of a contextual outlier (left part). C. C. Aggarwal (2017a) identifies three principal kinds of outliers in addition of application outliers, identifying contextual outliers as a kind of interest. Green instances are inliers and red instances outliers.

An additional kind of outliers is introduced: *application outliers* (Figure 2.5b). In this category, one can find *vector outliers*, *sequence outliers*, *trajectory outliers* or *graph outliers*, for example. The Figure 2.5b illustrates two examples of such kind of outliers: oriented graph (outlier is the linkage that goes backward) and trajectory (outlier is the trajectory with measurement error).

Similarly to Chandola et al. (2009), three kinds of outliers are proposed by C. C. Aggarwal (2017a):

1. The first and most studied type of anomaly is the *independent anomaly*. It can refers to a point anomaly or noise.

2. When data carries dependencies, contextual and/or collective anomalies can arise. For a contextual anomaly, an observation is declared to be an outlier because of its relationship to the remaining data items.

3. Collective anomaly refers to a set of data declared outliers. Such type of anomaly highly occurs in dependency-oriented data, such as sequences for textual data. There exists multiple other way to define such type of anomaly, and is related to the application.

In this work, data governs analysis of outlying observations. Special attention is given to each type of data and an introduction to many applications is also provided.

2.4.5 Conclusion

The taxonomy proposed by Hodge and Austin (2004) is further examined in the work of C. C. Aggarwal (2017a), explicitly addressing the differentiation between anomaly, outlier and noise. Depending on the degree of differentiation of an observation according to its distribution, an outlier can be considered either an anomaly or noise. This distinction helps to clarify the differentiation between undesirable outliers and outliers that warrant consideration. Furthermore, recent work by Ruff, Kauffmann, et al. (2021) introduces two novel types of anomaly based on the notion of intensity: *low-level sensory anomaly* and *high-level semantic anomaly*. The former deals with

	Single	Multiple	Independent	Contextual	Collective	Applications	Special context	Noise and intensity	Feature level
Hawkins (1980)	✓	✓	✓						
Hodge and Austin (2004)	✓	✓				✓			
Chandola et al. (2009)			✓	✓	✓		✓		
J. Zhang (2013)			✓		✓	✓			
C. C. Aggarwal (2017a)	✓	✓	✓	✓	✓		✓	✓	
Ruff, Kauffmann, et al. (2021)			✓	✓	✓		✓	✓	✓

TABLE 2.2: Taxonomy of anomaly and outlier, as introduced by the different references of the literature. Applications refer to applications related anomalies/outliers. Special context denotes the definition of contextual anomalies/outliers that requires special attention in several applications. Noise and intensity is the definition of an intensity of anomaly/outlier that differentiate noise and anomaly/outlier. Feature level is the introduction of special kind of contextual anomaly in which the features are difficult to handle.

observations that appear normal but possess slight variations that set them apart. The latter defines anomalies dependent on data representation and its dependencies.

Hawkins (1980) proposed a taxonomy for outliers, which has seen significant improvements over time. The taxonomy identifies two main situations: one involving independent and singular outliers, and the other involving collective outliers that occur together. The second situation can manifest in diverse forms depending on the application context, leading to the distinction between *collective outliers* and *contextual outliers*. While collective outliers are often contextual, occurence of multiple outliers which are abnormal together highly differ from multiple outliers which share common features, such as time or semantics, making them collectively abnormal. The integration of the *notion of intensity* for the definition of this taxonomy is essential. This approach offers an interesting perspective for understanding and characterizing outliers across various applications.

Figure 2.5 presents the combined taxonomy of collective and contextual outliers, as proposed by J. Zhang (2013). However, it is crucial to acknowledge that in certain applications, these two types can have distinct meanings. It reinforces the significance of the notion of intensity in outlier analysis. Table 2.2 illustrates how the literature overviewed the different kinds of anomalies and outliers. We denotes two principal things: taxonomy for both anomaly and outlier are similar and recent works aim to detail contextual anomalies/outliers. Also, we can observe that if contextual are not an identified kind, the application justifies the presence of anomalies/outliers. It is highly related to recent works that aim to specify contextual properties in purpose of more visibility.

Overall, the proposed taxonomy and the integration of the notion of intensity

offer valuable insights into understanding and classifying outliers, contributing to the advancement of outlier detection research. In conclusion, a taxonomy of three outliers can fit a wide number of situations but for other scenarios, the level at which the context is study is required.

2.5 Outlier detection approaches

Although the definition of an outlier can take several forms and methods for detecting them are numerous. They can be grouped into several families that are presented in this section. Before that, we introduce the one-class classification (OCC) task which is nowadays the most common task to perform outlier detection. Then, we present a comparison of the previously mentioned works that perform a comprehensive analysis of the task. Families of methods are then introduced: statistical approaches, proximity-based approaches, matrix factorization problem, high-dimensional approaches, outlier ensemble and neural approaches. For each of these families, we present the approaches studied in this thesis. Finally, we conclude the section with a discussion.

2.5.1 Unsupervised one-class learning

We previously introduced the task of outlier detection that aims to find out-of-distribution observations. Recent progress in data mining has seen its application to numerous fields such as computer vision, time series, natural language processing, etc ... Performing unsupervised one-class classification is similar to Equation 2.5. The output of a one-class classifier can be either a score or a label. For a label, we have:

$$\begin{aligned} f :&\mathbb{R}^D \longrightarrow \{-1, +1\} \\ &\mathrm{x} \longmapsto f(\mathrm{x}) \end{aligned} \tag{2.7}$$

with usually -1 if x is an outlier. In the case of a method that return a score, we have:

$$\begin{aligned} s :&\mathbb{R}^D \longrightarrow \mathbb{R} \\ &\mathrm{x} \longmapsto s(\mathrm{x}) \end{aligned} \tag{2.8}$$

The characteristic of using s is that, the higher the score, the greater the possibility that the instance is an inlier. On the other hand, the instance has more chance to be an outlier if the score is negatively low. In the case of point outlier an instance is considered outlying with respect to other data: when the score exceeds a threshold θ.

$$f(\mathrm{x}) = \begin{cases} +1 & \text{if} \quad s(\mathrm{x}) > \theta \\ -1 & \text{if} \quad s(\mathrm{x}) \leq \theta \end{cases} \tag{2.9}$$

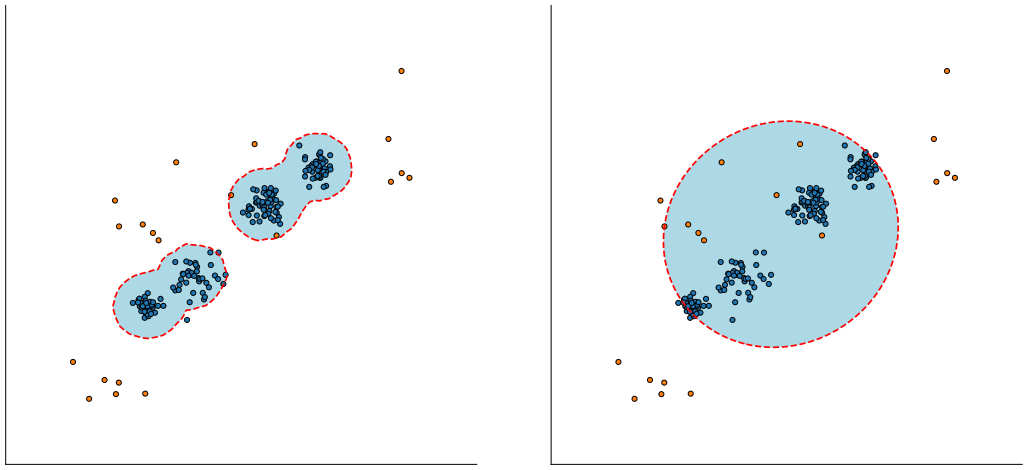


FIGURE 2.6: Example of two models with a set of multiclass data with local noise. Blue area is the inlier distribution predicted by the model and red contour its boundary decision.

Similarly to recent works (Zhao, Nasrullah, and Li, 2019; Fouché et al., 2020; Ruff, Kauffmann, et al., 2021; Manolache et al., 2021) we chose to refer to outliers as the positive class, for all this thesis. Thus, if a scoring model $\Upsilon(x) \in [0, 1]$ (see Equation 2.5) perform an outlier detection on x, the higher the score is, more is the probability that x is an outlier.

The efficiency of a method highly depends of its ability to generalize the normal distribution I (see Definition 2.3.6). Figure 2.6 displays two different models that perform one-class classification: proximity-based with k-nearest neighbors (left) and one-class support vector machine (right). Both of these methods differently separate the space according to training data. The k-nearest neighbors method succeeds to draw a more accurate decision boundary than the one-class support vector machine. This difference can be explained with two arguments: hyper parameters are not set accordingly for the support vector machine or there exists situations where one method excels and not the other. Preparing multiclass dataset for one-class classification induces several problems. While they have access to the same data, both methods handle such scenario differently from each other, not all methods can fit all situations. The noise for a class is obviously not the same for another. Figure 2.6 demonstrates that the choice of an approach against another one is an important step of outlier analysis.

In the context of outlier detection, the term "contamination" refers to the presence of outliers or abnormal observations within a dataset. It represents the degree to which the dataset is affected by the presence of such outliers. A contaminated dataset contains a non-negligible proportion of outliers compared to the overall data. Contamination can arise due to various reasons, including errors in data collection, measurement inaccuracies, anomalies, or rare events (see Section 2.3). These outliers can significantly deviate from the expected or normal behavior of the majority of the data. The level of contamination can vary from mild, where only a small fraction

of outliers is present, to severe, where a substantial portion of the data consists of outliers.

Handling contamination is crucial because outliers can distort statistical analysis, modeling, and decision-making processes. Contamination can impact the performance of outlier detection algorithms, as they need to accurately identify and distinguish outliers from the normal data instances. It also poses challenges in defining appropriate thresholds or criteria for determining what constitutes an outlier in the presence of contamination. Addressing contamination typically involves using robust outlier detection techniques that are designed to handle contaminated datasets. In this thesis, we note ν the contamination rate of a dataset X such as $\nu \in [0, 1]$, but is in practice define as $\nu \in [0, 0.3]$. The upper bound of ν is not intended to be high because outliers are intended to be in minority.

2.5.2 Overview of the literature

A comprehensive view of the field is made by C. C. Aggarwal (2017a). It results in covering almost all types of data that can be studied in data mining. In this section we introduce the reference works that have influenced recent studies.

Chandola et al. (2009) propose an identification of different types of output related to anomaly detection techniques. There exists a first kind of techniques that assign an anomaly score and another kind that output a label. The first type associates either a score of "abnormality" or a score based on "normality" to each instance. One example consists in sorting instances according to such score. The difference between them is that the former gives an analyst the opportunity to use, for example, domain-specific rules. In the case of the latter, it is only possible to tweak the input parameters. For outlier detection, there exists a wide number of approaches in the literature. Chandola et al. (2009) propose the following classification of approaches: *classification-based methods*, *nearest neighbor-based methods*, *clustering-based methods*, *statistical methods* and *information theoretic methods*.

J. Zhang (2013) proposes a comprehensive study of approaches. His study mainly focuses on applied outlier detection. Outlier detection methods for *Low Dimensional Data* are one of the earliest work in outlier detection. J. Zhang (2013) proposes a classification into four categories based on the techniques: *statistical*, *distance-based*, *density-based* and *clustering-based*. It differs from Chandola et al. (2009) with usage of categories of methods instead of task-related categories. The benefits lie in the granularity of method characteristics.

1. Statistical methods rely on distribution or probability models to fit the given data set. They cover *parametric methods*, like Gaussian model-based and regression model-based methods, and *non-parametric methods*, like histograms and kernel density function where there is no assumptions about the statistical distribution of data.

2. Distance-based methods rely on a distance metric (Euclidean, ...) in order to find outliers. It does not assume any underlying data distributions but scale better to multi-dimensional space and more complex data structures.

3. Density-based methods are a family of methods that involves to investigate not only local density of the instance but also local densities of its nearest neighbors. Methods such as *Local Outlier Factor* (LOF) and *Connectivity-based Outlier Factor* (COF).

4. Finally, clustering-based methods that deal with outliers depending of their characteristics. These methods implicitly define outliers as the background noise of clusters and some of them are built with mechanisms to reduce the negative effects of outliers. *DBSCAN* or *CLIQUE* are example of such kind of methods.

Outlier detection methods for *High Dimensional Data* are approaches that can handle dozens, hundreds or/and even millions of dimensions. In such kind of data, the curse of dimensionality appears and one of the main issues is dealing with sparse data. This situation often makes methods like distance-based inefficient because data tend to be equidistant to each other. J. Zhang (2013) proposes three categories: *Methods for Detecting Outliers in High-dimensional Data*, *Outlying Subspace Detection for High-dimensional Data* and *Clustering Algorithms for High-dimensional Data*.

1. Methods for detecting outliers in high-dimensional data address such instances through two main categories. The first one consists in performing a dimension reduction in order to apply low dimensional methods and/or feature selection. The second one aims at developing dedicated mechanisms, although more challenging to elaborate.

2. Outlying subspace detection for high-dimensional data is a technique that allow to investigate a subspace where the observation is exceptional or divergent from the rest of the population. We find dedicated methods in this category that study subspace characteristics, but also genetic methods (J. Zhang and H. Wang, 2006; J. Zhang, Q. Gao, et al., 2006; Sathe and C. C. Aggarwal, 2016).

3. Clustering algorithms for high-dimensional data are clustering methods that have been elaborate for such type of data. They are methods that focus on grouping data depending of the attributes and the of their characteristics. Often, finding a group of similar data imply to identify data that do not align with the rest of the distribution.

While static data, high-dimensional data, time series and numerical data were often presented, C. C. Aggarwal (2017a) proposes to describe how the task can be approached with the others. Thus, categorical, text, mixed attribute, discrete sequences, spatial, graphs and networks data are tackled. Although former methods are presented, most recent algorithms are also addressed in this contribution. Then, fundamental knowledges are approached and most studied scenarios are covered.

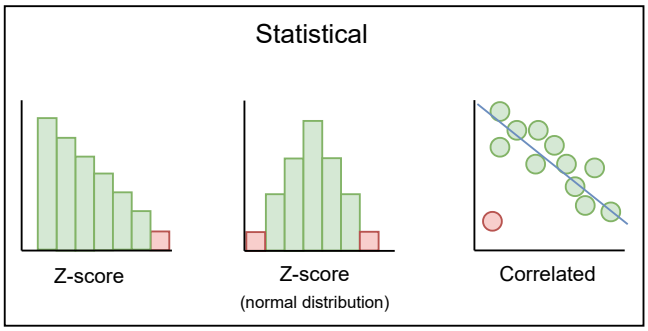


FIGURE 2.7: Statistical methods for *Z-score* outlier analysis on two histograms including normal distribution (left), and a third scenario with linearly correlated data (right). The first *Z-score* scenario is more suited for extreme value analysis, which not necessarily needs threshold.

2.5.3 Statistical methods

Early study were interested to compare an observation against a population. Statistical works were already achieving such analysis on various distribution and scenarios. These methods aim to find outliers according to the Definition 2.3.2. We propose to describe two popular approaches in such context: the *Z-score* based approach and the Mahalanobis distance based approach.

Outlier analysis with *Z-score*

One of the most popular approach for outlier detection, and also one of the most simple, is computation of *Z-scores* on univariate points (Rousseeuw and Hubert, 2011; V. Aggarwal et al., 2019; Chikodili et al., 2021) :

$$z_i = \frac{x_i - \mu}{\sigma} \tag{2.10}$$

Z-score has equivalent names including *Z-value*, *normal scores* or *standardized variables*. *Z-score* is computed with the mean of the population μ and the standard deviation σ. Such score aims to estimate how a value is relevant based on the other values of the dataset. Data mining tasks often take advantages of distance metrics for comparing observations with each other. Equation 2.10 can be extended with absolute value of numerator when applied in this context. The score will then relates how the current instance corresponds from the population mean of the standard deviation.

The Figure 2.7 displays three scenarios in which *Z-score* can be applied. Despite such method succeeds on artificial distribution, it is different for real-world data. Several drawbacks can be observed. In practice, complete intelligence about a dataset is rare. Another problem is that the *Z-value* assumes a normal distribution for the projected population. For data that carry contextual properties over their attributes, results of *Z-score* are expected to be poor. Finally, one last problem lies in the amount of data to get a significative *Z-score*. We note that sparse data are a kind

of observation where statistical methods struggle to be robust. Sparse data can be a challenge for statistical methods because such data are not fully observed and may have missing or uninformative variation per variable.

Extreme-value analysis with Mahalanobis distance

An intuitive outlier analysis aims to study extreme values when associated with distances or other metrics. The Mahalanobis distance has been widely used in outlier detection because it allows to measure a distance between a point x and a distribution X of dimension $N \times D$. Based on C. C. Aggarwal (2017a), we note $\overline{\mathrm{x}}$ the D-dimensional mean vector of X and Σ its $D \times D$ covariance matrix. The covariance matrix is processed with computation of each element $\mathrm{Cov}(\mathrm{X}_i, \mathrm{X}_j)$, which represents the covariance between variable X_i and X_j. Considering that we note x an D-dimensional data, we define the Mahalanobis distance d_M as follows:

$$d_M(\mathrm{x}, \overline{\mathrm{x}}, \Sigma) = \sqrt{(\mathrm{x} - \overline{\mathrm{x}})\Sigma^{-1}(\mathrm{x} - \overline{\mathrm{x}})^\top} \tag{2.11}$$

In contrast of *Z-score* previously approached, Mahalanobis distance is robust against the number of dimension thanks to the use of Σ. This characteristic is also important regarding the extreme value analysis categorization of this method. Indeed it can not be fully referred as an Extreme Value Analysis (EVA) because it normalizes the data through inter-attribute correlations. Another advantage of Mahalanobis is that it is parameter-free. Although EVA is efficient with aberrant data, they lack stability if the distribution is unknown and if the values are imbalanced as EVA focuses around the bounds.

2.5.4 Proximity-based approaches

For proximity-based approaches, we observe two principal categories that are popular in the literature. We present distance-based approaches and then present density-based approaches. The latter is often compared to the former.

Proximity induced that an object can be near or far of another object. It is obvious that methods that attempt to assert this property rely on distance metrics.

Definition 2.5.1 (Distance metric definition). *A distance d is a measure:*

$$d : \mathrm{X} \times \mathrm{X} \to \mathbb{R}$$

Such that for any $\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3 \in \mathrm{X}$*:*

1. $d(\mathrm{x}_1, \mathrm{x}_2) = 0 \iff \mathrm{x}_1 = \mathrm{x}_2$ — *identity of indiscernibles*
2. $d(\mathrm{x}_1, \mathrm{x}_2) = d(\mathrm{x}_2, \mathrm{x}_1)$ — *symmetry*
3. $d(\mathrm{x}_1, \mathrm{x}_3) \leq d(\mathrm{x}_1, \mathrm{x}_2) + d(\mathrm{x}_2, \mathrm{x}_3)$ — *triangle inequality*

Distance-based approaches

Proximity-based approaches are among the most popular methods that can be found in numerous tasks. The basic idea is that an observation can be compared depending of its distance with its (nearest) neighbors.

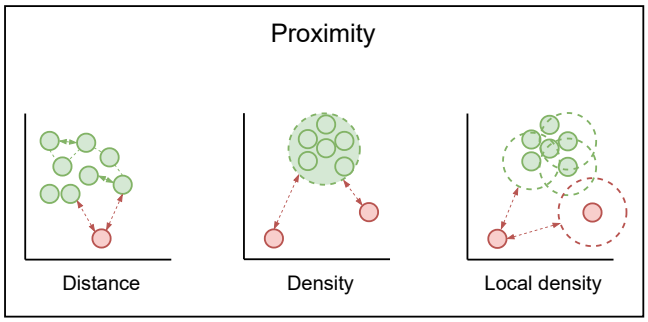


FIGURE 2.8: Three kinds of proximity-based approaches. The difference between them lies in the distance notion and from where and what it is processed. The intuitive method is the pure distance-based method (left) which compare an observation with its neighbors through a distance metric. Another kind of approach is based on the density (middle) such that if instances are near from each other, they create a dense area. The last types presented integrate the locality concept (right) which compare an observation with the local density of its neighbors.

Definition 2.5.2 (Distance-based outlier). *An outlier is a data that is significantly distant from other observations.*

Ramaswamy et al. (2000) have proposed to use the study of k-Nearest Neighbors (kNN). kNN is a distance-based approach that has numerous extensions. For one-class classification, the targeted result is a score or a label. Because outlier analysis is often unsupervised, a special version is often used.

Definition 2.5.3 (k^{th}-Nearest Neighbor based outlier score). *For* $\mathrm{x_i} \in \mathrm{X}$*, the distance between* $\mathrm{x_i}$ *and its* k^{th} *nearest neighbor within* $\mathrm{X}\backslash\{\mathrm{x_i}\}$ *is its score of outlierness.*

In this context, we define $distance_k(\mathrm{x})$ the function that finds the k nearest neighbors with their corresponding distance.

Definition 2.5.4 ($distance_k$). *Let* $d(\mathrm{x_1}, \mathrm{x_2})$ *the distance between* $\mathrm{x_1} \in \mathrm{X}$ *and* $\mathrm{x_2} \in \mathrm{X}$ *which is computed through euclidean distance, cosine distance or any other metric. Also,* $d_k(\mathrm{x_1})$ *is the distance function such that in the context of the* kNN*, for any* $k \in \mathbb{N}$ *we have:*

1. *for at least* k *observations* $\mathrm{x_2}' \in \mathrm{X}\backslash\{\mathrm{x_1}\}$*,* $d(\mathrm{x_1}, \mathrm{x_2}') \leq d(\mathrm{x_1}, \mathrm{x_2})$
2. *for at most* $k-1$ *observations* $\mathrm{x_2}' \in \mathrm{X}\backslash\{\mathrm{x_1}\}$*,* $d(\mathrm{x_1}, \mathrm{x_2}') < d(\mathrm{x_1}, \mathrm{x_2})$

Although the Definition 2.5.4 details how a distance metric can be used for an observation, the kNN returns the k-nearest neighbors of a point:

$$kNN(a) = \{\mathrm{x} \in \mathrm{X}\backslash\{a\} | d(a, \mathrm{x}) \leq distance_k(a)\} \tag{2.12}$$

Based on Equation 2.12, the cardinal $|kNN(\mathrm{x}_1)|$ can be greater than k if several observations are similarly distant from x_1. Considering Equation 2.12, the k^{th}-Nearest Neighbor (Definition 2.5.3) is deduced as follows:

$$k^{th}\text{-}NN(\mathrm{x}_1) = max\{kNN(\mathrm{x}_1)\} \tag{2.13}$$

In the Definition 2.5.3 we introduces a score related to the k^{th} nearest neighbor, but another approach focuses on the first nearest neighbor. A common attention is needed regarding the hyperparameter k that needs to be wisely chosen. There exist extensions of the Definition 2.5.3 that propose a differently defined score.

Definition 2.5.5 (Weighted k-Nearest Neighbor). *For $\mathrm{x_i} \in \mathrm{X}$, the outlier score is the average distance between $\mathrm{x_i}$ and its k-nearest neighbors within $\mathrm{X}\backslash\{\mathrm{x_i}\}$.*

All these approaches needs only one hyperparameter k and one distance metric. The performance of such method can be negatively impacted if the value of k is set too high or too low. If k is set too high, the method may be overly influenced by the majority of neighboring points, potentially leading to a loss of sensitivity in detecting outliers. Conversely, if k is set too low, the method may become too sensitive to noise or small fluctuations in the data, resulting in a higher likelihood of false positives.

The extension proposed by Definition 2.5.5 tackles this problem. It introduces the concept of weight distance, where the influence of each neighboring point is weighted based on their average distance. By incorporating the average distance, the decision-making process for selecting a suitable value of k is mitigated to some extent. The inclusion of the average distance provides a more nuanced and adaptive approach, allowing the method to adapt to varying densities and local structures within the data (Ramaswamy et al., 2000). This partially resolves the issue of strong dependency on the choice of parameters, enabling more robust outlier detection.

Density-based approaches

A popular type of proximity-based approaches is density-based methods. We have seen that distance-based methods estimate an outlier score with distance between observations. In contrast, density-based approaches locate an instance against principal areas where most of observation are gathering.

Definition 2.5.6 (Density-based outlier). *An outlier is located outside the dense area formed by the inliers.*

Distance-based methods, while naturally suitable for boundary-based tasks, can also be applied to address local density-related problems. One widely used approach for this purpose is the Local Outlier Factor (LOF) algorithm, which assesses the outlierness of an observation based on its proximity to the local density. The local density of a data point is computed by counting the number of data points within its neighborhood. A higher count indicates a higher local density, implying that the data point is surrounded by a dense cluster of neighboring points. Conversely, a lower count

suggests a lower local density, indicating that the data point resides in a sparser region. Similar to DBSCAN (Ester et al., 1996), LOF relies on two important concepts: reachability distance and core distance. According to Breunig et al. (2000) and Definition 2.5.4 , for $x_1 \in X$ and $x_2 \in X \backslash \{x_1\}$ we define the reachability distance of x with respect to x_2:

$$reach\text{-}dist_k(x, a) = max\{distance_k(a), d(x, a)\} \tag{2.14}$$

Density-based algorithms such as DBSCAN consider that an instance x is a *core* instance if at least *MintPts*[4] are reached within a range distance. Based on those parameters, a last characteristic of such kind of methods is the cluster volume. The cluster volume is determined by the collection of all core instances and their reachable neighboring instances. From here we refer to $kNN(x)$ as $N_k(x)$. For LOF, the *local reachability density* (*lrd*) of an observation x is performed as follow:

$$lrd_k(x) = 1/\left(\frac{\sum_{a \in N_k(x)} reach\text{-}dist_k(x, a)}{|N_k(x)|} \right) \tag{2.15}$$

Along with Definition 2.5.6, k refers to the minimum number of instances a dense area needs to reach. The Equation 2.15 showcases a local approach of such definition. Similarly to Definition 2.5.5, it is often more stable and interesting to compare an observation against other instances. The local outlier factor of x is defined as:

$$LOF_k(x) = \frac{\sum_{a \in N_k(x)} \frac{lrd_k(a)}{lrd_k(x)}}{|N_k(x)|} = \frac{\sum_{a \in N_k(x)} lrd_k(a)}{|N_k(x)|.lrd_k(x)} \tag{2.16}$$

The computation of LOF, as described in Equation 2.16, is influenced by several factors that density-based methods attempt to address. One advantage of the approach is its ability to identify outliers in remote areas with limited samples. However, this characteristic also presents a challenge as the outlier threshold varies depending on the data distribution. To mitigate this issue, several contributions have proposed solutions.

One notable work by Schubert et al. (2014) provides a comprehensive study of shared properties related to locality and distances, specifically focusing on LOF-based approaches. Based on this work, a simplified version of LOF, referred to as *Simplified-LOF*, is introduced, as represented by Equation 2.14. Additionally, extensions such as Local Outlier Probability (LoOP), proposed by Kriegel, Kröger, et al. (2009a), are included in their study. These efforts aim to improve the robustness and applicability of LOF to different scenarios and data types.

LOF offers a valuable approach for outlier detection, offering various possibilities among different applications and data types. To address some of the challenges associated with density-based methods, recent works have made significant contributions by

[4] Or minimum number of data points required to form a dense region

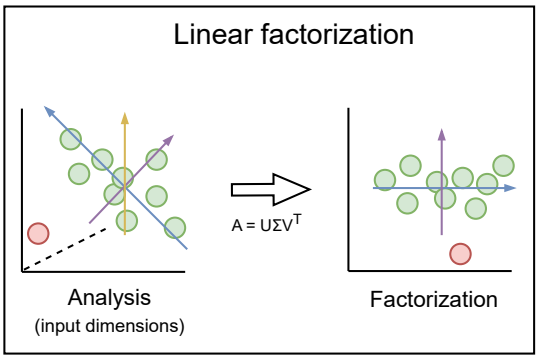


FIGURE 2.9: Linear decomposition of 10×3 raw matrix A in $10 \times K$ projected matrix A' with $K = 2$. If the linear model is PCA, the three eigenvectors (left), or principal components, describe how the raw data can be linearly projected. The factorization using $K = 2$ with blue and violet principal components is displayed on the right. Such scenario can be applied with different method such as Singular Vector Decomposition.

investigating the properties of locality and distances, introducing simplified versions of LOF, and exploring extensions such as LoOP. These advancements contribute to the ongoing development and refinement of LOF-based outlier detection techniques, and its usage in more recent works.

2.5.5 Matrix factorization problem

Matrix factorization problem is tackled in various domains and is not restricted to outlier analysis. Approaches that are proposed under this kind of problem are also referred as linear models. It involves decomposing a given matrix or tensor into a product of lower-dimensional matrices or tensors. This decomposition aims to capture the latent structure or underlying factors present in the data. Latent features refer to representations or dimensions in a model that capture underlying patterns or meaningful information information in the data (Cortes and V. Vapnik, 1995; Bengio, Courville, et al., 2013; LeCun et al., 2015). Thus, data distribution plays a crucial role in factorization matrix problems because it influences the nature and characteristics of the latent factors that are being extracted. Different data distributions can lead to distinct patterns and structures in the data, which can impact the effectiveness of factorization methods.

Matrix factorization aims to find a decomposition of the original matrix to reduce dimensionality while retaining as much information as possible. As an illustration, the goal is to approximate a given matrix A using a composition of two matrices, W and Z:

$$A \approx \tilde{A} = WZ^\top \tag{2.17}$$

The original matrix A of dimensions $\mathrm{N} \times \mathrm{D}$ is approximated as a decomposition of two matrices. Matrix W of dimensions $\mathrm{K} \times \mathrm{D}$, with $\mathrm{K} < \mathrm{D}$, captures the latent factors (or patterns) from the original data. Each row of W represents a low-dimensional

representation of a feature of A. The matrix Z of dimensions N $\times$ K, captures the weights (or importance) assigned to each latent factors (rows of W). The purpose of such decomposition is that any matrix A is decomposed into multiple matrices for keeping as much information while reducing dimensionality. The objective of this matrix factorization is to find optimal values for W and Z that minimize the reconstruction error between the original matrix A and its approximation $\tilde{A}$. This process allows for dimensionality reduction while preserving important patterns or relationships in the data (D. Lee and Seung, 2000; Koren et al., 2009).

Thus, such methods can also be referred to dimension reduction approaches because they map D-dimensional features data to K-dimensional latent factors. For proximity-based approaches, outliers can be found in peculiar area of space in which they are arrange differently from inliers. In contrast, linear models aim to find lower-dimensional subspaces where outliers have a high chance to be found. For C. C. Aggarwal (2017a), such method can be viewed as an orthogonal version of proximity-based methods which try to describe data horizontally rather than vertically. Horizontal view of data refer to seeing such observation on rows or data values rather than on vertical view which refers to columns or dimensions. The assumption for analysing outliers with matrix factorization approaches is as follows:

Definition 2.5.7 (Outlier with linear models). *An outlier is an observation that presents dimension values that are poorly or differently correlated with features of the rest of data.*

For outlier analysis, Shyu et al. (2003) have proposed an approach using Principal Component Analysis (PCA) (Hastie et al., 2001) with the Mahalanobis distance (Equation 2.11). Thus, they achieve outlier analysis with their approached called Principal Component Classifier (PCC). The mentioned method use PCA for analysis of principal components, which are eigenvectors of the covariance matrix Σ. Considering that a data set A of D dimensions and N observations is mean-centered, we note the covariance matrix Σ:

$$\Sigma = \frac{A^\top A}{N} \tag{2.18}$$

Principal components are a key notion of PCA, we note $K \in \{1 \dots D-1\}$ the approximated largest number of eigenvectors. Thus, PCA follows three properties:

1. principal component are uncorrelated
2. the first principal component has the highest variance
3. variance of transformed instance along each eigenvector is the corresponding eigenvalue

Any data can be transformed through orthonormal eigenvectors matrix called P. We note A' the transformed data matrix:

$$A' = AP \tag{2.19}$$

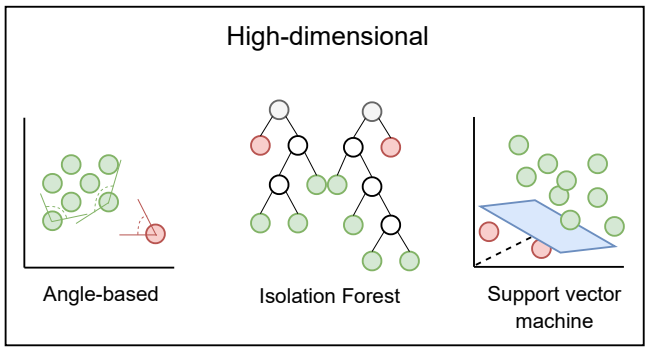


FIGURE 2.10: Presentation of three popular approaches that tackle high-dimensional data. The intuitive angle-based method finds outliers through short angles (left). For IF (middle), the outlier score is processed with the path length among isolation trees. SVM estimates a $D-1$ dimension hyperplane that separate the observations (right).

Based on Shyu et al. (2003) and C. C. Aggarwal (2017a) and Equation 2.11, with e an eigenvector of eigenvalue (variance) λ, the estimated outlier score is computed as follows:

$$PCC(\mathrm{x}) = \sum_{j=1}^{D} \frac{(\mathrm{x} - \overline{\mathrm{x}}).e_j}{\lambda_j} \tag{2.20}$$

The Equation 2.20 can be assimilated as the sum of weighted projected distances to the eigenvector hyperplanes. In addition, Shyu et al. (2003) propose to filter outlying data through Mahalabonis distance before applying PCA on the sampled distribution. For the rest of the thesis, we refer to PCC following the Equation 2.20.

While PCC can perform as both low and high dimensional approach for outlier analysis, it arises two principal drawbacks: noise sensitivity and regularization issues. In Shyu et al. (2003) setup, extreme values are filtered through Mahalabonis distance and PCA can then focus on low correlated observations. Depending of the data and the real contamination (ν), PCA may poorly performs with respect to the optimal hyperplane. It is also sensitive to particular features or sparse matrices. PCA is also dependent of the number (N) of observations: if there are not a sufficient amount of instances in X, the covariance matrix can be hard to estimate.

2.5.6 High-dimensional approaches

High-dimensional data sets are characterized by having a large number of features, which can imply unique challenges for traditional data analysis methods. The complexity of high-dimensional data arises from several factors. One of these factors is the presence of complex correlation structures, where variables may exhibit intricate relationships with each other. Additionally, high-dimensional data can exhibit various characteristics, such as sparsity (where most of the variables have zero or very few non-zero values), contextual correlation (where the correlation between variables depends on specific contexts or subsets of the data), or even lack of correlation. For addressing the challenges mandated by high-dimensional data, numerous methods and

algorithms have been developed. These methods aim to uncover meaningful patterns, identify outliers, or perform other tasks in the context of high-dimensional data analysis. Some popular high-dimensional approaches include distance-based approaches with subspace rotation and cosine distance. These methods focus on finding similarities or dissimilarities between high-dimensional data points based on their orientations in different subspaces. Another class of methods uses tree-based approaches, such as Isolation Forest (IF), which leverages the concept of isolating anomalies in a decision tree-like structure.

High-dimensional approaches are methods that focus on multivariate distribution with complex correlation. Nature of data can be sparse, contextually correlated, uncorrelated, ... Methods tackling such kind of data are numerous and may eventually been applied on full dimensionality or partial dimensionality of raw distribution. Among those approaches, we find distance-based approaches with subspace rotation and cosine distance, or also tree-based approaches such as Isolation Forest (IF), to mention a few. Thus, we propose to study methods that are designed for high-dimensional purpose first. Subspace-based methods are popular and approaches such as Rotation-based Outlier Detection (ROD), proposed by Almardeny et al. (2020), or Subspace Outlier Detection (SOD) of Kriegel, Kröger, et al. (2009b) have found success with high-dimensional data. Furthermore, the definition of high-dimensional outliers can be difficult to formalize hence the number of application is high.

Definition 2.5.8 (High-dimensional outlier). *An outlier in high-dimensional data refers to an observation that deviates from the correlation patterns observed in the majority of the data. Moreover, outliers can also manifest in subspaces where inliers are not adequately represented.*

For high-dimensional outlier detection methods, we first propose the study of methods based on angles and rotations. Specifically, the cosine distance emerges as a popular metric widely applied to high-dimensional data. We prefer to categorize this approach as high-dimensional rather than distance-based (Section 2.5.4) because the emergence of this kind of method was motivated by the high-dimensional property of the data. Additionally, we describe the isolation forest method, which can be considered an ensemble approach for outlier detection due to its utilization of several decision trees. Given its success in dealing with high-dimensional data, we categorize it as a high-dimensional approach. Furthermore, we present the One-Class Support Vector Machine (OCSVM), which leverages the kernel trick to identify outliers. OCSVM can also be categorized as a linear model in terms of its underlying principles, we propose to categorize it in high-dimensional methods in regard to how SVM can tackle a wide amount of scnenarios. While various approaches exists, these three kind of approaches allow to exhibit different and unique techniques that address outliers in high-dimensional data.

Angle-based methods

Angle-based approaches present interesting properties against high-dimensional data. Indeed, the approach is more suitable with this type of data because of its use of angles instead of distance metrics. In Section 2.5.4, distance-based approaches have been mentioned. They find success for numerous applications and types of data. While common distance metrics can be used for high-dimensional data, they get poor results with exploding number of dimension. They also lack success with sparse matrices. Distances like euclidean distance consider all dimensions the same way. One drawback of such property is the sensitivity to irrelevant dimensions: in high-dimensional data, some dimensions may be irrelevant or contain noise. Euclidean distance treats all dimensions equally, which means that irrelevant or noisy dimensions can significantly impact the distance calculations and potentially introduce inaccuracies in outlier detection. All of these problems are included ad referred as *curse of dimensionality*. Angle-based methods offer an alternative approach that can mitigate the effects of the curse of dimensionality. Instead of relying solely on distances between data points, angle-based methods focus on the relationships between vectors or subspaces formed by the data. By considering the angles between vectors or subspaces, these methods can capture the intrinsic structure of high-dimensional data more effectively.

Cosine distance is often a key part of such kind of methods due to several reasons The cosine distance is the complement of the cosine similarity with matrices of positive features, and is purely not a distance metric. The reason is that the cosine distance does not follows the triangle inequality (Definition 2.5.1). Considering two feature vectors $\mathrm{x}_1, \mathrm{x}_2 \in \mathrm{X}$, we note the cosine similarity is defined as follows:

$$s_{cos}(\mathrm{x}_1, \mathrm{x}_2) = \frac{\mathrm{x}_1 \cdot \mathrm{x}_2}{||\mathrm{x}_1|| \cdot ||\mathrm{x}_2||} \tag{2.21}$$

Here, $||\cdot||$ represents the L_2-norm, and $s_{cos}(\mathrm{x}_1, \mathrm{x}_2) \in [-1, 1]$ where -1 means that a is opposite to b, 1 that they are the same and 0 that they are orthogonal. Hence, any other values indicate relative similarity or dissimilarity. Based on Equation 2.21, the cosine distance is performed as follows:

$$d_{cos}(\mathrm{a}, \mathrm{b}) = 1 - s_{cos}(\mathrm{a}, \mathrm{b}) \tag{2.22}$$

With non-negative features, $d_{cos}(\mathrm{x}_1, \mathrm{x}_2) \in [0, 1]$ and is often used as distance metric for high-dimensional and sparse data. Proximity-based methods can benefit from Equation 2.22.

Cosine distance is insensitive to the magnitude or length of vectors. It only considers the orientation of the vectors, not their absolute values. This property can be seen in the cosine similarity Equation 2.21, where the similarity is determined by the angle between the vectors, not their magnitudes. In cosine distance calculation, vectors are implicitly normalized by dividing them by their magnitudes. This normalization step ensures that the distance calculation is not biased by varying vector lengths. The normalized vectors can be represented as $\mathrm{x}' = \frac{\mathrm{x}}{||\mathrm{x}||}$ with x'.

A popular angle-based approach for outlier analysis is Angle-Based Outlier Detection (ABOD) (Kriegel, S hubert, et al., 2008). The idea of this algorithm is to measure the coverage of angle for an observation (spectrum). An observation with a large coverage is most likely to be an inlier, and one with a low coverage an outlier. For example, the Figure 2.10 shows that inliers have high angles to cover most of nearing observations while the outliers need a low one. ABOD uses an Angle-Based Outlier Factor (ABOF) that, given a point $\mathrm{x} \in \mathrm{X}$ and a norm $||\cdot|| : \mathbb{R}^D \to \mathbb{R}^+$, consider the scalar product denoted by $< \cdot, \cdot >: \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$.

Based on Kriegel, S hubert, et al. (2008), ABOF is valued as:

$$ABOF(\mathrm{x}) = \mathrm{Cov}(\mathrm{x}_1, \mathrm{x}_2) \cdot \left(\frac{< (\mathrm{x}_1 - \mathrm{x}), (\mathrm{x}_2 - \mathrm{x}) >}{||\mathrm{x}_1 - \mathrm{x}||^2 \cdot ||x_2 - x||^2} \right) \tag{2.23}$$

for any $\mathrm{x}_1, \mathrm{x}_2 \in \mathrm{X}$.

Based on Equation 2.23, ABOD performs the ABOF on every pair of points, which has a high computational cost. For tackling this problem, another approach proposed by the same authors consists in approximating results with usage of sampling. This method is defined as *FastABOD* (Kriegel, S hubert, et al., 2008) and take less time to compute than original ABOD thanks to sampling. Regarding this latter enhancement, ABOD can also be categorize as a probabilistic approach. ABOD can also be considered an extreme-value analysis method, and is therefore sensitive to contextual outliers. While angle-based approaches can tackle the problem of curse of dimensionality, they are not an all-in-one paradigm. We also note that they are often free parameter methods.

Isolation forest

Isolation forest (F. T. Liu et al., 2008) is an anomaly detection approach that is close to *random forests*. It is widely used in anomaly detection, outlier detection, fraud detection, and many others. However, its categorization can be complicated with its characteristics to build tree ensembles or subspace sampling. The first refers to ensemble outliers, which is detailed in the next section, and the second to subspace methods.

The isolation forest algorithm build several isolation trees which are typically proper binary decision trees (each node in the three has exactly zero or two daughter) with at most $\mathcal{N}$ leaf nodes. Each of these leaf nodes represent exactly one instance.

Definition 2.5.9 (Isolation tree). *Given T a node of the tree, it can be either an external node with no child or an internal node with one test and exactly two daughter nodes T_l and T_r. For a feature j and a split value p, the test $j < p$ separates data points into T_l and T_r.*

The first step of the algorithm is to build isolation trees with recursively dividing X into subsets with randomly selected features j and split value p. It is achieved until:

1. a depth limit is reached

2. $|\mathrm{X}| = 1$

3. all observations in the considered subset have the same values

The path length of a point x is noted $h(\mathrm{x})$. It is measured by the number of edges x traverses an iTree from the root node until the traversal is terminated at an external node. Evaluation is done by means of a scoring function based on the average path length $avg(h(\mathrm{x}))$ among isolation trees. Precisely, the path is measured by the number of edges that traverses isolation trees from the root node to the corresponding node (external) of observation x. Given H_n the n-th harmonic number, estimated by $\ln(n) + 0.57721$ (Euler's constant), the isolation forest score for a given $\mathrm{x} \in \mathrm{X}$ and the number of examples N in the training set is performed as:

$$IF(\mathrm{x}) = 2^{-\frac{avg(h(\mathrm{x}))}{2H_{N-1} - \frac{2(N-1)}{N}}} \tag{2.24}$$

Considering that IF is monotonic to $h(\mathrm{x})$, in Equation 2.24 and based on F. T. Liu et al. (2008), the score follows the assessment: i) if $IF(\mathrm{x})$ is very close to 1, then x is an outlier, ii) if $IF(\mathrm{x})$ is much smaller than 0.5, then x is a normal instance and iii) if all instance $\mathrm{x} \in \mathrm{X}$ return $IF(\mathrm{x}) \approx 0.5$ then the entire sample does not have distinctive outlier.

With isolation forest, outlying points are scored during the test phase using the split condition (p) computed in training phase. Strictly speaking, observation that reach an external node quickly are considered outliers. It can be view as out-of-sample observation that does not fit regular behavior of the majority of sample instances. The subsampling characteristic of IF is similar to the axis-parallel subspaces method proposed by Kriegel, Kröger, et al. (2009b).

One-class support vector machine

Support Vector Machine (SVM) are method with a rich literature, and are subject to a high number of extensions. Also, this kind of method is defined by supervised learning in which $y_i \in \{-1, 1\}$. Outlier detection is often depicted as an unsupervised problem regarding the lack of labelled data set. One-Class Support Vector Machine (OCSVM) is a type of SVM which does not requires any target label. OCSVM assumes that all observations belong to the inlier class (see Section 2.5.1). Thus, the origin of a kernel-based representation belongs to outliers (C. C. Aggarwal, 2017a).

Given the Φ unknown non-linear function that projects the raw data to a space with higher dimension and its corresponding coefficients vector w, the hyperplane that separates the inliers from outliers is represented as follows:

$$\mathrm{w} \cdot \Phi(\mathrm{x}) - \rho = 0 \tag{2.25}$$

where $\rho \in \mathbb{R}$ is a bias variable that determine the position of the hyperplane that separates inliers from outliers. For w, it represents the weight vector in the feature space and is a parameter that determines the orientation and direction of the hyperplane. We present the optimization problem extended from this equation in which

the computed value is positive for majority of X. This property has been mentioned previously and is the result of the hypothesis that all observations belong to inliers. From Equation 2.25 the margin regularizer term $\frac{||\mathrm{w}||^2}{2}$ is added and separate ρ from the objective function for closing the origin with inliers. The objective function is then performed as follows:

$$\text{Minimize } \frac{||\mathrm{w}||^2}{2} + \frac{C}{N}\sum_{i=1}^{N}\max\{\rho - \mathrm{w}\cdot\Phi(\mathrm{x_i}), 0\} - \rho \tag{2.26}$$

In Equation 2.26, the *slack penalty* $max\{\rho - \mathrm{w}\cdot\Phi(\mathrm{x_i}), 0\}$ aims to handle the scenario in which result of Equation 2.25 is negative. Also, the constant $C > 1$ estimates the trade-off between maximizing the margin and the training errors.

Instead of solving w, the optimization approach is often preferred using the *kernel trick*. It allows to perform the high-dimensional projection and substitute $max\{\rho - \mathrm{w}\cdot\Phi(\mathrm{x_i}), 0\}$ with *slack variables* $\xi_1 \ldots \xi_N$ in Equation 2.26:

$$\text{Minimize } \frac{||\mathrm{w}||^2}{2} + \frac{C}{N}\sum_{i=1}^{N}\xi_i - \rho \tag{2.27}$$

For Schölkopf, Williamson, et al. (2000), introduction of the parameter $\nu = \frac{1}{C}$ now characterize the solution which sets an upper bound on the fraction of outliers (contamination) and a lower bound on the number of training examples used as support vector. This extension is referred as ν–SVM or ν-SVC in the literature, and is the version we are using in this thesis.

The dual formulation has N variables $\alpha = [\alpha_1 \ldots \alpha_N]^T$ which correspond as Lagrangian parameters. Every α_i weight the decision function and few of them are actually non-zero value. Thus, for a $kernel function$ $K(\mathrm{x}, \mathrm{x_i}) = \Phi(\mathrm{x})^T\Phi(\mathrm{x_i})$, we score OCSVM as follows:

$$\Upsilon(\mathrm{x}) = \sum_{i=1}^{N}\alpha_i \cdot K(\mathrm{x}, \mathrm{x_i}) - \rho \tag{2.28}$$

While being successful, OCSVM with ν extension has several drawbacks such as the choice of the kernel function which often is either linear, polynomial, sigmoid or Radial Basis Function (RBF). Hidden parameter such as C can also make it difficult to use kernel function. The method also needs N to be high enough to efficiently estimates the hyperplane. Despite the mentioned drawbacks, the approach has been widely used for numerous applications and is still competitive in recent benchmarks of the literature.

2.5.7 Outlier ensemble

Ensemble methods are popular in numerous domain, and is not strictly limited with outlier analysis. The main idea behind these methods is that combination of several models, also called *base detectors*, and their outputs is more robust than a single

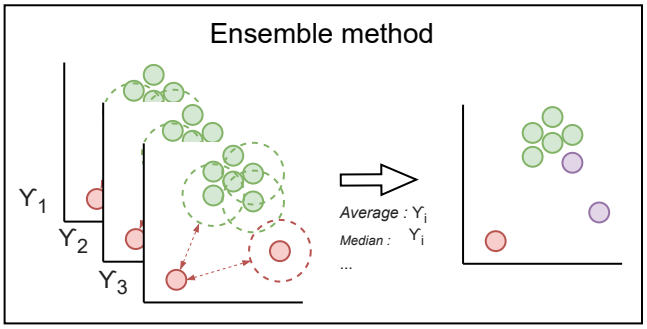


FIGURE 2.11: Ensemble method performing the fusion technique (average, median, etc ...) with X and local density model ($k \in [1, 3]$). Observations that were not estimated as outliers with every models are represented in purple.

model. Base detectors are not limited to be identical, they can be different. Although the possibility to combine multiple base detectors is intuitive, the design of such approaches needs special attention regarding normalization of outputs. Another attention is also needed in the choice of the base detector. Independently of the previous statements, sampling the raw data set X is also important. Sampling can then be applied either with data sampling or with features sampling.

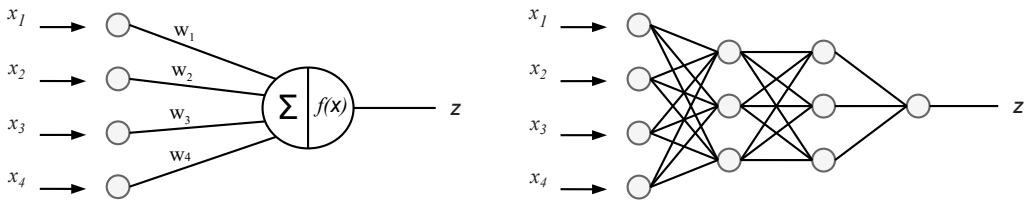
There exist three basic and popular policies for performing outlier combination: Average, Maximum and Median. The average is popular among classification literature and offers to outlier detection more robust and stable results. Such method succeed to trivially generalize meaningful behaviors of inliers and reduce the bias effect raised with unbalanced distribution. Median and maximum functions are more complicated to generalize because they rely on the success of a base detector. For the average, if all base detector perform poorly, the result is more often to be biased. Bias can be reduced thanks to the distribution selection or the feature bagging techniques.

Feature bagging, also known as random subspace method, is an ensemble learning technique (Breiman, 1996). It involves creating multiple subsets of features from the original dataset and training individual models on each subset. The predictions from these models are then combined to make the final prediction. The main idea behind feature bagging is to introduce diversity among the models by using different subsets of features. This helps to reduce overfitting and improve the overall performance and generalization ability of the ensemble.

C. C. Aggarwal and Sathe (2015) have proposed approaches that rely on base detector pooling. They are referred as Average-of-Maximum (AOM) or Maximum-of-Average and integrate feature bagging for mitigating bias-variance tradeoff.

2.5.8 Neural networks

Based on the natural structure of the human brain, artificial neural networks (ANN) simulate the synaptic system. For human beings, the learning process is performed by increasing importance of the connections between neurons (also called cells or units).



(A) Perceptron (B) Multilayer perceptron

FIGURE 2.12: The simplest perceptron has a single layer (left) and uses a linear activation function to make predictions based on weighted inputs. The multilayer perceptron (right) is built with multiple layers of interconnected neurons.

Artificial neural networks have similar behavior: neurons are connected through weighted links. Thus, a neuron receives its input from other neurons based on connections to other neurons, or input values. Nowadays, there exist numerous types of neural networks. Some of them have exactly one layer of connected units to the input, while some have *hidden* layer(s) of neurons between the input and the output. Multiple layers can also be considered for connecting one layer to the next one, multiple times.

We present in Figure 2.12 the most basic artificial neural network: the perceptron (Rosenblatt, 1958). The single layer perceptron, noted Υ as referring to Equation 2.5, processes the weighted sum of the input x such as:

$$\Upsilon(\mathrm{x}) = \sum_{i=1}^{D} w_i \cdot x_i \tag{2.29}$$

On top of this basic computation, a differentiable *activation function* is usually used (Bishop, 2006). Multilayer perceptron is the sequential addition of several perceptron connected with each other. Considering that setting, more complex architecture can be built. Two types of neural network framework are explored for outlier detection: one-class neural networks and replicators.

One-class classification

In this section, we introduce a simple approach based on a one-class perceptron with mean squared error (MSE) loss for outlier detection, inspired by C. C. Aggarwal (2017a). As the perceptron is a simple model, we aim to present the key concepts for performing one-class classification with neural networks. The key assumption behind one-class neural networks is that the output of the perceptron, denoted as Υ, should ideally be zero, despite the non-zero weights w_i ($i \in [0, D]$) associated with it (see Equation 2.5). In this model, all training observations are assumed to be inliers, and thus, the prediction $\Upsilon(\mathrm{x})$ is expected to be 0. However, if $\Upsilon(\mathrm{x})$ deviates from 0, it indicates that the instance is an outlier and does not conform to the underlying inlier model. Consequently, the objective of a one-class neural network is to optimize the model in such a way that it maximizes the score for outliers while minimizing

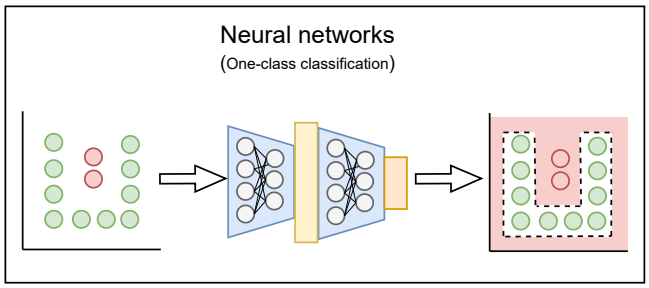


FIGURE 2.13: One-class classification performed with neural networks are often projecting the input space in a new feature space that aims to characterize the normal data behavior.

the score for inlier data. Thus, a decision boundary that maximizes the separation between inliers and outliers is expected.

To achieve such optimization, we use the backpropagation algorithm which trains a *feedforward* ANN by updating its weights. This algorithm aims to solve the optimization problem for finding a local minimum of a differentiable function. In this context, we use the Mean Squared Error (MSE). This function estimates the loss by calculating the average squared error between the predicted value and the actual value. To illustrate a one-class perceptron, we consider the squared error function. Such a function is compatible with the characteristics of the outlier detection task previously explained. Let W be the set of all weights values, the squared error of the instance values x is computed as follows:

$$\Upsilon(\mathrm{x})^2 = (W \cdot \mathrm{x})^2 \tag{2.30}$$

The gradient descent is performed for each training instance with the weight update $W - \eta\nabla\Upsilon(\mathrm{x})^2$ in which $\eta > 0$ is the learning rate and ∇ the corresponding activation rule of the neural network. Considering the squared error at the iteration p, the gradient descent update of the weights is performed as follows:

$$W_{p+1} = W_p - \mathrm{x}\eta\Upsilon(\mathrm{x}) \tag{2.31}$$

All instances of X are given to the perceptron and W is updated until convergence is reached. At this point, the one-class perceptron output is is valued as:

$$\Upsilon(\mathrm{x}) = (W \cdot \mathrm{x})^2 \tag{2.32}$$

For the case of the multilayer perceptron, W is not anymore a vector but a matrix that corresponds to each layers weights values. The multilayer case is more challenging because the sum of the squares errors of each output is performed. Normalization of the weight is performed with the additional constraint $W^\top W = I$ that ensures mutual orthogonality.

The perceptron is the basic version of actual neural networks. With recent advances in neural networks there exist numerous approaches for tackling the optimization problem of one-class neural networks. Recently, Ruff, Vandermeulen, et al. (2018) have proposed an application of the Support Vector Data Description (SVDD), introduced by D. M. Tax and Duin (2004), as a kernel function that minimizes the volume of a hypersphere. This approach is called Deep SVDD and the optimization problem is tackled as follows:

$$\underset{R,\mathcal{W}}{\text{Minimize}} \quad R^2 + \frac{1}{\nu N}\sum_{i=1}^{N}\max\{0, ||\phi(\mathrm{x_i};\mathcal{W}) - c||^2 - R^2\} + \frac{\lambda}{2}\sum_{l=1}^{L}||\mathrm{W}^l||_F^2 \tag{2.33}$$

where $\tilde{\mathcal{X}} \subseteq \mathbb{R}^p$ is the output space of dimension p and $\phi(\cdot; W) : \mathcal{X} \to \tilde{\mathcal{X}}$ a neural network with $L \in \mathbb{N}$ hidden layers. We set the weight W^l of layer $l \in \{1, \ldots, L\}$ considering $\mathcal{W} = \{W^1, \ldots, W^L\}$. For the kernel SVDD, minimizing R^2 minimizes the volume of the hypersphere characterized by the radius $R > 0$ of center $c \in \tilde{\mathcal{X}}$. In this equation $||\cdot||_F$ is the Frobenius norm and there are two hyperparameters: λ and ν. The ν-parameter (Section 2.5.1) is related to the expected contamination of the data sample and $\lambda > 0$ a weight decay regularizer on the network parameters $\mathcal{W}$. The authors propose a simplified objective function:

$$\underset{\mathcal{W}}{\text{Minimize}} \quad \frac{1}{N}\sum_{i=1}^{N}||\phi(\mathrm{x_i};\mathcal{W}) - c||^2 + \frac{\lambda}{2}\sum_{l=1}^{L}||W^l||_F^2 \tag{2.34}$$

Then, the outlier score is performed as follows:

$$\Upsilon(\mathrm{x}) = ||\phi(\mathrm{x};\mathcal{W}^*) - c||^2 \tag{2.35}$$

where $\mathcal{W}^*$ are the network parameters of a trained model. Deep SVDD is a one-class neural network that proposes a boundary separation of the data that can be illustrated in Figure 2.13.

Reconstruction problem

In the previous section we have approached the one-class neural network framework with a basic illustration and a state of the art technique. There exists another approach for estimating an outlier score with neural network that is referred as *replicator* neural networks. Instead of minimizing a projection error through all attributes, the model aims at predicting the input x_i with the reconstructed $\tilde{x_i}$. In this setting, the final output dimension is identical to the input, as illustrated in Figure 2.14. Basically, let x be an instance and $\tilde{x_j}$ the predicted value corresponding to x, the reconstruction error is estimated by

$$\sum_{j=1}^{D}(x_j - \tilde{x_j})^2, \tag{2.36}$$

This error should be minimized in the training step. Autoencoders are the most popular approach for reconstructor neural networks and can handle high-dimensional

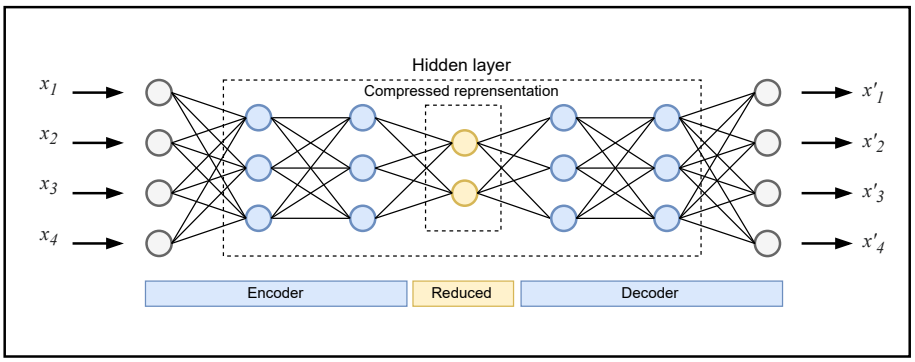


FIGURE 2.14: Autoencoder is a neural network architecture that is built with the encoder/decoder framework. The main idea is that an encoder (left part) aims to perform dimensionality reduction for keeping meaningful features and weights (the bottleneck of the architecture). The decoder attempts to reconstruct the original input (right part).

data in a similar behavior than PCA or matrix factorization (Section 2.5.5).

These past years, different types of autoencoders have been proposed in the literature. It is the case of the Variational Autoencoders (Kingma and Welling, 2013), referred as VAE. It differs from autoencoders in the output of the encoder one: it is a probability distribution instead of a single output. Thus, the VAE aims to describe the samples of the dataset at the latent space level. The loss function is also different than the basic autoencoder, the Kullback-Leibler divergence is used. We observe several benefits with this kind of autoencoder: the trained model is a generative model that relies on the latent variables and can be used as a one-class estimator.

Based on such kind of neural networks, we can highlight two examples of outlier detection algorithms in recent contributions: a Generative Adversarial Network (GAN) by Y. Liu, Li, et al. (2019) and a reconstruction-based autoencoder by Lai et al. (2020). The former is based on the GAN neural network approach that is built with a *generative* network that generates observations and a *discriminative* network that evaluates them. For instance, the case of Single Objective Generative Adversarial Active Learning (SO-GAAL) (Y. Liu, Li, et al., 2019), the generator proposes outliers while the discriminator estimates a rough boundary. The principal issue lies in the trained model which can be more or less successful to estimate what an outlier really is. For mitigating this issue, Y. Liu, Li, et al. (2019) have proposed a Multiple-Objective Generative Adversarial Active Learning (MO-GAAL) approach with sub-generators that handle different subspaces. The idea of using multiple generators helps to mitigate the bias problem mentioned in Section 2.5.7.

On the other hand, Lai et al. (2020) have proposed a Robust Subspace Recovery (RSR) AutoEncoder that aims to robustly and nonlinearly reduce the dimension of the original data (Lerman and Maunu, 2018). It is an autoencoder with a RSR layer which map the inliers around their original locations and the outliers far from their original locations. The Figure 2.15 depicts such method. With RSR layer they also propose a

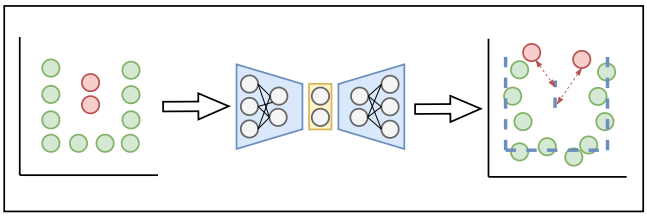


FIGURE 2.15: Often, the reconstruction error of autoencoders is performed for outlier analysis. The model presented here (middle) is an autoencoder, which has three main parts: encoder (left in blue) and decoder (right in blue) with an hidden layer (yellow). We can see through the decoding step, outliers can not be as well reconstructed than inliers.

reconstruction loss function that combines the usual loss described in Equation 2.36 with the RSR loss. For a training data matrix $A \in \mathbb{R}^{N \times D}$ and any instance $\mathrm{x_i}$ of A, the RSR loss is processed as follows:

$$\begin{aligned} L_{RSR}(A) &= \lambda_1 L_{RSR_1}(A) + \lambda_2 L_{RSR_2}(A) \\ &:= \lambda_1 \sum_{i=1}^{N} \|\mathrm{x_i} - A^\top A \mathrm{x_i}\|_2 + \lambda_2 \|AA^\top - \mathrm{I}_d\|_F^2 \end{aligned} \tag{2.37}$$

with $\lambda_1, \lambda 2 > 0$ two hyperparameters, I_d the $d \times d$ identity matrix. The first term of Equation 2.37 is a term that increases robustness while the second term optimizes orthogonality. During the training step, the RSR layer attempts to find an orthogonal projector whose subspace robustly approximates the dataset.

A brief conclusion for reconstruction problem is that recent advances in neural networks have brought promising approaches that can be used in several applications. This characteristic lies in the heavy use of the hidden state of encoder-decoder architecture. The bottleneck formed by the output of the encoder acts as a dimensionality compressor that codes informative subspaces. Several approaches for learning reconstructor have been highlighted previously and they are near from each other when we compare how the loss optimization is performed. For instance, the RSR layer can be applied in a GAN approach for learning a more robust generator.

2.5.9 Discussion

Methods for outlier detection have increased greatly over the years. Although dedicated methods are very popular to perform the task, there is also the possibility to use techniques from other fields. It is also possible to see a clear evolution in the interest of the community towards data types that were not represented initially. First, artificial data have been used by statisticians, then real world data have been introduced by the data mining community. Recently, the study of outliers has been able to address high dimensional data issues. Advances in the various fields of artificial intelligence have allowed the community to focus on more complex data.

		True class	
		Positive (outlier)	Negative (inlier)
Predicted class	Positive (outlier)	True positive T_p *(Correct)*	False positive F_p *(Incorrect)*
	Negative (inlier)	False negative F_n *(Incorrect)*	True negative T_n *(Correct)*

TABLE 2.3: Confusion matrix applied to outlier detection task. We consider the positive (1) as outlier and the negative one (0) as inlier.

For these last data, we observe a trend to either reduce the dimension of high-dimensional data or creating dedicated methods. The dimensionality reduction allows to then apply common methods like distance-based approaches or parametric approaches. Also, we note that several techniques are not limited to one kind of approach. It is the case of the matrix factorization techniques that can be used among most of mentioned methods. Independently of the outlier detection task, neural networks share similar optimization problem that matrix factorization techniques.

From statistical methods to outlier ensemble methods, outlier detection approaches often rely on the parameter ν which can be used as hyperparameter. This parameter can be estimated based on the input distribution, or also on the model. While it can be difficult to find, few works are approaching the ambiguity of setting it. For instance, in the $\nu-$SVM model this parameter can be an inhibitor for the boundary of the projected subspace. On the other hand, we also find this parameter in the data preparation and the evaluation. Finally, we can observe that most of the approaches are following two kind of policy for discovering outliers: *progressive* and *regressive*. Progressive approaches separate the data based on the representative sample and evict/penalize points that do not fit such assumption. Regressive method like LOF are expressive for outlier and less informant for inliers and thus explores outliers, not inliers.

2.6 Evaluation

We have previously presented several methods that perform outlier detection through different paradigms. Performing a machine learning task is separated in several steps. We have introduced that we tackle outlier detection and have presented algorithm and models that perform this task. Once the problem is defined and the model selected/built, an expert has to split the original dataset. Often, data are separated between train split, test split and validate split. This step allows to evaluate a model on a prepared data split that the model has not been trained on.

The idea of comparing several approaches is intuitive and special preparation is required. In this section we introduce how methods are performed in order to be benchmarked against the literature. Firstly, we describe how the evaluation step is

tackled in the literature. Secondly, we present the common benchmark properties for performing outlier analysis. Lastly we present the evaluation metrics that are used in this thesis.

2.6.1 How to evaluate outlierness ?

The question of the evaluation of outlier detection approaches has often been made in relation to the field of application. In our context, we refer to evaluation of a classifier or of a system in information retrieval. For Hawkins (1980), the problem of evaluation was intimately linked to the case study since the evaluation hypotheses were made at the same time as the construction of the observations. However, this method allows for qualitative evaluation in an unsupervised setting.

Subsequently, the evaluation methods of the approaches were often derived from modelling techniques. Each application strongly influences the type of automatic evaluation that is carried out. For example, in the work of Hodge and Austin (2004), Chandola et al. (2009), and J. Zhang (2013) it is possible to find the use of the confusion matrix with the F1-measure or the accuracy when predicting an outlier. Another popular method in data mining was the mean-squared radius. Nevertheless, the evaluation step was still largely dependent on the application domain and the type of modeling.

The formalism of evaluation methods is taken up by C. C. Aggarwal (2017a) by proposing a comparison of recent trends in data mining. He depicts a progressive evolution of the field towards an automatic evaluation methodology common to many methods. Thus, a measure of external validity is raised by the joint use of precision and recall when predicting an instance or an outlier. The latter makes it possible to evaluate the output of an anomaly score according to a threshold. Ruff, Kauffmann, et al. (2021) complements the evaluation step with an analysis of several scenarios in which it is preferable to focus on such metrics. In the absence of expert feedbacks, precision and recall are preferred for evaluation of an unbalanced problem like OD.

The current version of evaluation step is recent and keep growing thanks to advances in data mining. Proposition of a metric based on precision and recall shows a willingness to compare each contributions within numerous applications. The problem of comparing outlier detection methods remains, however, dependent on the fields of application and the clarity of the protocols. In this respect, Ruff, Kauffmann, et al. (2021) wisely raises the question of the interpretability of the output of the methods. They also highlight the importance of this step in a field where protocols tend to be different.

2.6.2 Benchmarking and preparation

Special attention is required for the preparation of the experimental setup. While resources like ODDS[5] or UCI[6] give access to various data set, it is difficult to find real-world data that are designed for outlier detection. Literature of data mining for outlier analysis has proposed to prepare data set from classification and clustering domains to fit the needs of the task. Thus, for C. C. Aggarwal (2017a) and Ruff, Kauffmann, et al. (2021) the benchmark and the data are keys requirements for outlier analysis.

There exist several strategies for preparing data set from other tasks. With binary or multiclass data set, one class is selected as inlier and other classes are outliers. Depending of the application, several classes can be selected as inliers. Another strategy consists in having data set with human labelled on outliers, inliers or both. This scenario is the ideal one. Lastly, generation of synthetic outliers allows to get full control on the benchmark properties.

Above strategies have their advantages and disadvantages but they often consider a supervised evaluation. The first strategy implies some risks in the way that noise and conflictual inliers can not be properly identified beforehand. While the second strategy avoids this drawback thanks to a manual preparation of the data, labels and annotations can be incomplete. Finally, the last mentioned strategy is often used in statistical setup but unfrequent in data mining. Outliers in data mining can be difficult to reproduce or to generate. For all of these reasons, the first scenario is often preferred in data mining.

2.6.3 Evaluation metrics

When performing outlier detection on observations of a dataset, we observe four outcomes that can be drawn in the *confusion matrix*. For example, a model that attempts to classify cats and dogs can find cat among cat pictures, find dogs among dogs pictures, confuse dogs as cat or confuse cats as dogs. For OCC, we define outlier as the positive class and inlier as the negative class. The Figure 2.3 describes a confusion matrix applied to OCC.

Based on the confusion matrix, we find evaluation metrics that measures different characteristics of a model can be drawn. The first metric is the *Precision* and displays the proportion of correct positive identifications:

$$Precision = \frac{TP}{TP + FP} \tag{2.38}$$

The *Recall* is often used to complement *Precision*. It displays the proportion of real positive examples among the predicted positive ones:

[5]ODDS is a dedicated resource gathering outlier detection dataset and can be found following this link: http://odds.cs.stonybrook.edu/.

[6]It is a machine learning repository gathering multiple data set for numerous tasks. Numerous of the mentioned data set in the literature are using it. The resource can be found in http://odds.cs.stonybrook.edu/.

$$Recall = \frac{TP}{TP + FN} \tag{2.39}$$

The *Recall* can also be named True Positive Rate (TPR). Usage of both metrics is popular in numerous contributions in classification, information retrieval, ...

It is different for outlier detection because a score is almost always computed. Based on such score, ranking can be more convenient for evaluation. For displaying performances of an approach with automatic metric, a threshold τ is chosen.

The Receiver Operating Characteristic (ROC) curve is one of the most popular evaluation metric for outlier detection. It plots TPR with False Positive Rate (FPR) at different classification thresholds τ. If Equation 2.39 is equivalent to TPR, the FPR is computed as follows:

$$FPR = \frac{FP}{FP + TN} \tag{2.40}$$

If the ROC curve draws TPR against FPR, the usually used aggregated evaluation metric is the Area Under the ROC curve (AUROC). AUROC provides a metric from $[0, 1]$ where 1 is the perfect classifier, 0.5 a random classifier and 0 a degenerate model. AUROC can be overly optimistic with high imbalanced test sets.

Another metric is often used for outlier detection in order to complement the AUROC. This metric is called Area Under the Precision-Recall curve (AUPRC). Similarly to AUROC, it draws a curve based on Precision against Recall for a classification threshold τ. The Average Precision (AP) is more robust for computing AUPRC. In every scenario, the interpretation of both metrics has to be carefully done. Depending of the application, one curve can be more informative than another. With no preference, both are recommended.

2.7 Conclusion

As we have shown in this chapter, different concepts are associated with the study of outliers and it is not surprising for this task that is used in many fields. A comparison of the literature has allowed us to build a concrete vision of how it evolves through recent days. Formally, the definition of an outlier has been developed over time, as evidenced by Definition 2.3.1, Definition 2.3.7 and Definition 2.3.9. However, the emergence of data mining methods has further influenced the definition of outliers. Additionally, terms such as anomaly and novelty have been used interchangeably with outliers.

Based on this observation, it becomes evident that a significant portion of the research community is motivated by the practical applications of studying outliers. Tasks like spam detection have emerged, leveraging advancements in outlier detection and anomaly detection techniques. Despite their common usage, it is important to note that these terms have distinct characteristics that impact their execution. For instance, spam detection can be based on metadata or raw data, where outlier detection approaches may not necessarily be utilized. It is therefore important to note

that the study of outliers extends beyond its name, a notion that we develop in the Chapter 3.

Following this analysis of the state of the art, we have detailed the main approaches that perform outlier detection with unsupervised learning. These approaches can be used in a very large number of applications, we focus particularly in text in Chapter 3. Some approaches are not treated at all in the context of textual data, we address this in the Chapter 4. To this day, the study of outliers is mainly associated with the field of data mining and thus benefits from popular automatic evaluations. However, we observe that this evaluation step is dependent on the availability of labels and data sets. However, the clear advantage of this task is the presence of methods that allow the removal of noise or the pre-processing in absence of labels. We explore this use case in Chapter 5.

Chapter 3

Outlier analysis for text

The Chapter 2 introduces outlier analysis and the task of outlier detection. Thus, we know that outlier detection is originally performed through statistical domains before appearing in data mining research works. Such evolution has successfully been made for high-dimensional data but several types of data lack reasonable number of contributions and interest. Textual data are poorly represented in outlier detection task to such an extent that anomaly detection and novelty detection are preferred. This observation is related to the characteristic of text that follows different rules depending of the generation model (news article, email, technical report, fantasy book, ...). For instance spam detection benefits from outlier detection techniques but is not necessarily associated to such domain.

In this thesis we aim at studying outlier analysis for textual data through several applications and recent trends. Because there is few contributions that focus on text, we mainly propose an analysis of compatible techniques from anomaly detection, novelty detection, spam detection and plagiarism detection. In Section 2.1 we have introduced background knowledge and notations for unsupervised learning with data.

We describe in Section 3.1 the problems and motivations of applying outlier analysis to text. Section 3.2, proposes an introduction to natural language processing for data mining (text mining). The Section 3.3 focuses on the definition of a textual outlier. Some difference can be observed depending of the study of raw text: syntax and semantic. An outlier taxonomy is proposed in Section 3.3.4, adapted from Section 2.4 and taking into consideration Section 3.3. In Section 2.5 we focus on the issue of outlier detection techniques that are best suited for one or multiple types of data. The same observation can be made with text, and we highlight the principal approaches of the literature in Section 3.4. Section 3.5.2 presents the general evaluation protocol of the literature and our algorithm GenTO which prepares data for outlier detection benchmarking. Based on GenTO, we present an experimental study of the best performing and most popular approaches of the literature for outlier detection on text in Section 3.6. We conduct an extensive analysis of experimental results in Section 3.7 and conclude in Section 3.8.

3.1 Problems and motivations

Outlier detection is a task that concerns numerous domains and applications. The task is not recent, Abraham and Box (1979) and several of contributions that introduces it are already available in the literature (Hawkins, 1980; Hodge and Austin, 2004; Chandola et al., 2009; J. Zhang, 2013; C. C. Aggarwal, 2017a; Ruff, Kauffmann, et al., 2021). Formerly, the task was performed through statistical domain (Beckman and Cook, 1983) with static data and parametric methods. The recent definition of outlier is not the same than in the past and implies a different way of performing outlier analysis. Most popular methods have emerged within high-dimensional and parametric context. Although these approaches produce significant results in many types of data, its application to text lack clarity at different levels: definition of a textual outlier, a dedicated taxonomy and how to properly evaluate reference methods.

3.1.1 Problems

In Chapter 2 we have observed that there are sufficient number of works that offer a survey of outlier detection task (Hawkins, 1980; Hodge and Austin, 2004; Chandola et al., 2009; J. Zhang, 2013; C. C. Aggarwal, 2017a; Ruff, Kauffmann, et al., 2021) but they rarely investigate the scenario in which data are purely textual. They have proposed and detailed numerous characteristic such as taxonomy of outliers, taxonomy of methods, scenarios, applications, evaluation and more recently interpretability. Interest for textual outlier detection has recently been ignited with neural networks (Gorokhov et al., 2017; Ruff, Zemlyanskiy, et al., 2019; Lai et al., 2020) and matrix decomposition methods (Allan et al., 2008; Kannan et al., 2017). While these contributions focus to perform outlier detection task, referred as anomaly detection, they do not investigate the nature of OD within text. In this issue, we also note two related problems to the current state of the literature: recent approaches are not comparing themselves with recent contributions and they do not investigate outliers built with their experimental setup.

3.1.2 Motivations

The main motivation of this chapter lies in exploring how outlier analysis is performed with text and also how to properly conduct the task. In Figure 3.1 we present an illustrative example involving three newspapers. Although the main topic is Sport/Tennis, we observe the presence of an article from Politics in the corpus, indicating a misclassification. Such scenario is common, and the task of spam detection has similarities with this problem. However, the existing literature on Textual Outlier Detection (TOD) can be confusing to navigate due to variations in terminology and the lack of dedicated datasets. In addition, most of the recent contributions create a new experimental setup each time with old and new datasets. Because there is not yet dedicated datasets for textual outlier detection, the proposal of an algorithm for outlier generation seems necessary.

Topic: Sport/Tennis. Inlier documents:
x_1: Naomi Osaka came to Flushing Meadows to entertain and did not disappoint on Monday, overcoming a slow start to beat Czech Marie Bouzkova 6-4 6-1 and get her U.S. Open title defence under way in front of a roaring capacity crowd.
x_2: World number one Novak Djokovic had to work hard for a three-set victory over Hungarian Marton Fucsovics at the Paris Masters on Tuesday in his first match since losing the U.S. Open final in September.

Topic: Politics. Outlier document:
x_3: The Olympic Games have started and yet, all countries are [...] Japanese tennis star Naomi Osaka on Friday lit the Olympic cauldron to mark the formal start of Tokyo 2020, in an opening ceremony shorn of glitz and overshadowed by a pandemic but celebrated as a moment of global hope.

FIGURE 3.1: Example of three articles from Reuters and Eurosport when searching for news about tennis and politic topics (for readability, only sentences of interest are presented). Inliers documents address Tennis results of players while the outlier topic is the open ceremony of Olympic Games that appears to mention a tennis player.

Approaching the task of TOD is difficult for many reasons: poor number of extensive comparison with state of the art approaches, domain specific problems (email, news papers, scientific articles, ...) that are similar but associated to different tasks naming (spam detection, plagiarism detection, anomaly detection, novelty detection, ...), lack of dedicated datasets and divergent experimental protocols from the literature. Our contribution to formalize outlier detection in text can be structured around three challenges: lack of comprehensive works tackling textual outliers which categorize and reference recent advances, a definition of an outlier in this context and an experimental study on approaches of the literature. We propose a short introduction to each of these points.

Extensive study of related works

Our first contribution is the study of OD for text data, with an extensive analysis of related methods. While TOD approaches are rare in the literature, we also study popular methods of novelty detection, anomaly detection, plagiarism detection, spam detection and one-class classification. Furthermore, we explore the similarities and differences between these methods and outlier detection in text, identifying commonalities in terms of methodologies, feature extraction techniques and evaluation metrics. This comparison allows us to draw connections between different domains and leverage insights from related areas to inform the development of effective outlier detection techniques for text data. Conducting this extensive study aims to provide researchers and practitioners in the field of textual outlier detection a comprehensive overview of existing methods and their suitability for different kind of text data. This analysis serves as a foundation for our subsequent contributions, enabling us to build upon the existing knowledge and identify gaps that need to be addressed.

Taxonomy and generation approach of outliers

To facilitate a comprehensive understanding of outliers in the context of text data, we propose a taxonomy that categorizes different kinds of outliers. This taxonomy is based on the established formalism from past surveys and works of outlier detection. This identification not only provides a structured categorization of outliers but also establishes a connection to well-known types of outliers discussed in the existing literature. It takes into account factors such as semantic coherence, topic relevance, and contextual relationships within the text. Following this taxonomy, we propose an approach that generates point/independent outliers and contextual/conditional outliers: GenTO. Our algorithm is generic and can be easily applied to any corpus with a hierarchy of topics and therefore applied in future works.

Comparative experimental setup

The type of outlier is not clearly identified in the evaluation step of the literature, leading to an incomplete read of the results. In addition of this issue, we define ν the contamination rate of a corpus (weight of outliers number in a corpus). In various work, such parameter can varies in order to give an idea of the robustness of one approach against real-world data. A comparative work that highlights these characteristics, based on GenTO, is also proposed in order to evaluate state of the art approaches at a same level. By conducting a comprehensive evaluation using the GenTO algorithm as a baseline, we can provide a fair and informative comparison of state-of-the-art techniques, facilitating a deeper understanding of their effectiveness in this domain.

3.1.3 Applications

We have seen in the literature that there are many applications for outlier detection, it is also the case for textual data. It may seem difficult to list all the possible and existing applications, but it is interesting to highlight the most common applications. There are four popular applications: spam detection, plagiarism detection, "topic drift" detection and first story (novelty) detection. These applications can also be seen in many works from the NLP field, such as translation or text classification.

Spam detection

Email spam refers to the appearance of messages that differ in malicious or marketing characteristics, in terms of content, from expected messages in an email box (Karim et al., 2019; Spirin and Han, 2012). This application has seen a strong interest over the years, mainly motivated by the increasing integration of email into the daily lives of users. There are also a large number of techniques in the literature that perform this task, including the use of anomaly detection methods. These include density-based techniques (You et al., 2020; Idris et al., 2014), semantic similarity (Laorden et al., 2014) and systems (Idris et al., 2014). It is interesting to note that other types of data than e-mails are also subject to the same problems. For example, the spam

detection task can be tackled with outlier detection methods and applied to twitter data (T. Wu et al., 2018). Popular data sets are Spambase, SMS Spam Collection and Enron Email.

Plagiarism detection

Plagiarism is a problem that has been studied for a long time by scholars and academics. Although there are many forms of plagiarism, the data mining community is particularly interested in plagiarism of documents (whole or partial) and web content (Parker and Hamblen, 1989; Chang et al., 2021). By studying a document and its composition, outlier detection methods can target parts that differ from the rest (D. Guthrie, L. Guthrie, et al., 2007). This application is called intrinsic plagiarism detection. There is also the scenario where the comparison is done by document, and in this case also the use of outlier detection techniques is effective. Among the techniques used are the probabilistic study of a distribution (Stein and Eissen, 2007), parametric methods on document representations (Muhr et al., 2009) and clustering (Potthast et al., 2011).

Semantic outlier detection

We categorize the topic drift (rare topic) detection task as the detection of documents that do not belong to the main topic of a corpus (Ruff, Kauffmann, et al., 2021; C. C. Aggarwal, 2017b; Mahapatra et al., 2012). It is one of the most studied tasks in the context of textual outlier detection. Precisely, the task consists in characterising the main topic or topics of a corpus in order to find documents that greatly differs from it. Among the outlier detection techniques used are neural networks (Lai et al., 2020; Ruff, Zemlyanskiy, et al., 2019; Hu and Khan, 2021), density-based methods (Kannan et al., 2017), Support Vector Machine (SVM) (Manevitz and Yousef, 2001), proximity-based (C. C. Aggarwal, 2017b) and fuzzy clustering (Lazhar, 2019). Most popular data sets are 20-Newsgroups and Reuters-21578.

First story detection

The task of novelty detection in text consists of identifying a new document that differs from the rest of the corpus. It is often considered as a technique to increase the efficiency of a neural network technique (Kryściński et al., 2018) or statistical approaches. Applications can also be found in streaming or big data where it is important to detect a change of feature or topic (F. Wang et al., 2018; Shanmugam et al., 2020). In this context, novelty detection shares similarities with the concept drift task (Lu et al., 2018; Bhattarai et al., 2020), which is concerned with the latter problem.

3.2 Representation models for text

In the previous chapter we have observed that study of outliers is related to data mining field. When data mining is applied to text, we refer to the sub domain of text mining. Notations and popular techniques of dat mining can be addressed to text but text mining also integrates natural language processing (NLP) methods. Natural language processing is a set of approaches and structures that make human language accessible to computers. Its application are numerous in real world through dialog assistants, translation and other examples. It takes benefit from artificial intelligence and machine learning literature and shares several characteristics with them.

In text mining, we represent text through vectors, graphs, etc ... One of the most popular and easiest way to represent text is the tokenization with bag-of-words. In this representation, raw text is transformed in a sequence of *tokens* (or *terms*). A token can be a symbol, a word, a pair of words, a sentence and many others. Through this representation, a document represents a sentence, a quote, a paragraph, a section, an email, a tweet, a blog, etc...

Textual data are special data derived from natural language. In order to apply algorithms on them, it is necessary to choose or develop a representation. These techniques often consist of transforming the raw textual data into vectors. Several kind of these representations are presented in this section. First, the most common and easier representation to apprehend is bag-of-words, presented in Section 3.2.1. An extension of bag-of-words, the term frequency inverse document frequency representation, is detailed in Section 3.2.2. In Section 3.2.3 we present the linear decomposition approach. We detail in Section 3.2.4 the recent language models and in Section 3.2.5 the very recent large language models. Finally, a discussion is proposed in Section 3.2.6 that outline the numerous challenges of text representation.

3.2.1 Bag-of-words

Bag-of-Words (BOW) consists to build a dictionary of terms, here words, which are independent of each other.Each dimension x_j of a document x is the count number of the j-th word in the corpus X.In another way, each dimension can either be represented by the *term frequency* tf or the *document frequency* df.The Table 3.1 displays an application of a term frequency bag-of-words representation with two documents. For the document x_2 we can observe that the majority of the dimensions of its representation x'_2 are set to 0. Such phenomena is expected to worsen as we add more documents, resulting in increasing the size of vocabulary $\mathcal{V}$. We note that in this example with BOW representations, a term is a word be can be an n-gram or any other feature type. Transforming a text with this method gives a high-dimensional and sparse vector. Because we intuitively transform raw text based on a vocabulary, for short and moderate documents most of the dimensions are set to 0. This phenomena is also referred as *curse of dimensionality*.

In practice, several techniques exist for tackling this problem but they present cons and pros on top of their solution. One the most known techniques is *stop words*

Vocabulary ($\mathcal{V}$)	x'_1	x'_2
a	1	2
little	1	1
cat	0	1
hope	1	0
eating	0	1
however	1	0
desperate	1	0
is	1	1
never	1	0
withtout	1	0
worth	1	0
fish	0	1

(A) BOW representation

Document	Raw text
x_1	a little hope, however desperate, is never without worth
x_2	a little cat is eating a fish

(B) Raw documents (corpus)

TABLE 3.1: Example of a BOW model (left) with two documents x_1 and x_2 (right). For each word (term) of $\mathcal{V}$ the frequency is counted for documents, one by one. Thus, each dimension of the BOW model represent the term frequency in a document. This example displays how a document from raw space $\mathcal{X}$ is represented in space $\mathcal{X}'$.

filtering which consists of removing most recurrent words of a language. For the Table 3.1 the term with the higher frequency is *"a"* and is expected to appear in numerous documents. Examples of stop words foe english language include *"the", "and", "is", "are" and "a"*. These words are often filtered as they do not contribute much to the overall understanding of differentiation of documents. It also contribute to reduce the dimensionality of BOW representation and potentially improve the effectiveness of classification models.

Another optimization can be found in the tokenization process which break down a text document into individual tokens. The goal of tokenization is to segment the text into meaningful units that can be further analyzed. Such technique can involve removing punctuation marks, splitting on white spaces and handling special cases like contractions or hyphenated words. In Table 3.1 punctuation has been removed. In addition, other preprocessing steps may also be applied to text data before creating BOW representation. These can include lowercasing all raw texts for ensuring case-insensitive matching, stemming or lemmatization to reduce tokens to their base form (e.g., *"eating"* to *"eat"*), filtering special characters and handling numerical/special symbols. For any application and task, it is important to note that removal of stop words or choosing any other preprocessing step on raw text is not always necessary or beneficial. A special attention is needed regarding these optimization.

3.2.2 Term frequency inverse document frequency

The most popular extension of bag-of-words is Term Frequency Inverse Document Frequency (TFIDF) approach where the frequency of each term is discounted by its frequency in the corpus. The term's weight processed by TFIDF is computed as

follows:

$$x_{i,j} = \mathrm{tf_{i,j}} \cdot \log\left(\frac{N}{\mathrm{df_i}}\right) \tag{3.1}$$

The recurrent problems of BOW methods are sparsity, dimensionality and semantics. Although its popularity in Information Retrieval literature, TFIDF is the target of some issues such that lack of term distribution knowledge and relevance/quality criterion. There are other methods in machine learning which allow to represent the text with other properties. One type of approach is to reduce the BOW representation of the text into a latent space in order to extract strong features such as semantics. Such approaches include Latent Semantic Analysis/Indexing (either LSA or LSI) (Deerwester et al., 1990; Landauer et al., 1998), Principal Component Analysis (PCA) or Autoencoder (AE).

3.2.3 Linear decomposition

PCA and AE have been explained in Section 2.5, we complete matrix factorization methods (Section 2.5.5) with LSA (or LSI). LSA explores the best subspace approximation of the original document space based on term co-occurence. This characteristic is related to semantic relationship within terms connection. Singular Vector Decomposition (SVD) (Stewart, 1993) is the technique used by LSA for identification of relationship patterns between terms. Given a term matrix (BOW matrix) $\mathrm{X}' \in \mathbb{R}^{N \times D}$ and an approximation rank r, LSA decompose X' as follows:

$$\mathrm{X}' = \mathbf{U}\Sigma\mathbf{V}^\top \tag{3.2}$$

with $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ are the singular values of X'. $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices that are respectively called left and right singular vectors, and Σ is a diagonal matrix. The **Truncated** singular-vector decomposition is a popular technique in machine learning and natural language processing. The approximation objective is preformed as follows:

$$\min_{\mathbf{U},\Sigma,\mathbf{V}} ||\mathrm{X}' - \mathbf{U}\Sigma\mathbf{V}^\top||_F \tag{3.3}$$

in which $\mathbf{U} \in \mathbb{R}^{V \times r}$ considering $\mathbf{U}\mathbf{U}^\mathrm{T} = \mathbb{I}$ and $\mathbf{V}^\mathrm{T} \in \mathbb{R}^{N \times r}$ considering $\mathbf{V}\mathbf{V}^\mathrm{T} = \mathbb{I}$. In Equation 3.3, $||\cdot||_F$ is the Frobenius norm with $||\mathrm{X}'||_F = \sqrt{\sum_{i,j} \mathrm{X}'^2_{i,j}}$. In truncated singular-vector decomposition, the hyperparameter r truncates the r largest singular values with their corresponding singular vectors ($\mathbf{U}$ and $\mathbf{V}$). Based on Equation 3.3, the approximated matrix has a minimal error and all documents from X are now represented as dense vectors of continuous numbers. More importantly, the embedded matrix now refer to a **semantic space** and the features are now called latent features/variables. In this chapter, we study how such representations can address outlier detection task.

3.2.4 Language models

Language models (Bengio, Ducharme, et al., 2000) refer to models that are designed to understand and generate human language. They are statistical or probabilistic models that capture the patterns and structure of language. Language models can range from simple n-gram models that calculate the probability of word sequences based on their frequency in a given corpus, to more complex models like recurrent neural networks (RNNs) (Sutskever, Martens, et al., 2011) or transformer models that learn to predict the likelihood of a sequence of words given the context.

Language models are a kind of approach that focuses on the representation of text according to its natural distributed representation at several levels. These methods compute the probability of a word sequence, not term only, for addressing fluency and vocabulary issues. Neural network methods are among the most effective methods in a wide range of applications. Neural Language Models (NLM) such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) or BERT (Devlin et al., 2019) can be cited. These methods can be used to represent different levels of text, such as letters, words, sentences, paragraphs or documents. Language models are prone to be used in generative tasks because of their ability to handle sequences. Although these methods are highly efficient, they are complex to develop and require a substantial computing environment.

Word embedding

Word2Vec and GloVe represent text as a dense vector that capture the semantic and syntactic relationship between words. Such kind of approach is also referred as word embedding. For Word2Vec, neural networks are used with either Continuous Bag-of-Words or Skip-gram (D. Guthrie, Allison, et al., 2006; Mikolov et al., 2013) architectures. This approach predict the context of a token given its neighboring tokens, or vice versa. On the other hand, GloVe is a combination of two model: global matrix factorization, which refers to Equation 2.17 in Section 2.5.5 and Section 3.2.3, and local context window methods (Bengio, Ducharme, et al., 2000; Collobert and Weston, 2008). It leverages global co-occurence statitics of words within a corpus.

Word embeddings offer several advantages for representing text. First, they can capture semantic similarities between words enabling to find semantic relationship. Second, word embeddings provide compact and dense representations, which are computationally efficient and can be easily used as input for downstream tasks. Additionally, the embeddings can be pretrained on large corpora and then fine-tuned on domain-specific data, leveraging the general language knowledge learned from the pretraining phase.

However, it's important to note a few limitations of word embeddings. They may struggle with representing rare words or words that are not present in the training corpus. Out-of-vocabulary words might be represented by unknown tokens or have suboptimal embeddings. Additionally, word embeddings lack explicit representations

of word order and sentence structure. While they can capture word-level semantics, they may not fully capture the nuances of longer phrases or sentences.

Contextualized word embedding

To address word embedding limitations, contextualized word embeddings have been developed. Unlike traditional word embeddings that assign fixed vectors to words, contextualized word embeddings generate different embeddings for the same word based on its context. For Embeddings from Language Models (ELMo) (Peters et al., 2018), a bidirectional Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005) architecture is used for generating word representations. Then, ELMo can considers the left and right context of a word simultaneously, allowing it to capture dependencies and nuances that depend on a specific context.

Bidirectional Encoder Representations from Transformers (BERT) is a model that introduced the transformer architecture for contextualized word embedding. Unlike ELMo, it employs a self-attention mechanism (Vaswani et al., 2017) to capture global dependencies and contextual information across the entire input sequence. In recent years, numerous extension such as XLNET (Z. Yang et al., 2019), RoBERTA (Y. Liu, Ott, et al., 2019) and ALBERT (Lan et al., 2019) have made improvements in training objectives and architectures of BERT to enhance the representation quality. Another kind of contextualized word embedding are OpenAI's Generative Pretrained Transformers (GPT) (Radford, Narasimhan, et al., 2018) and GPT-2 (Radford, J. Wu, et al., 2019). They differ from BERT-based models in their architecture and training objective. GPT is a generative model that also employs transformer architecture using an autoregressive approach during training. It learns to predict the next word in a sequence based on the preceding words. It captures the dependencies between words in a left-to-right manner so GPT excels at generating coherent and contextually appropriate text.

Contextualized word embeddings offer several benefits, they can capture word sense disambiguation, where the same word may have different meanings based on its context. This is particularly useful in distinguishing rare or unusual word usages. Contextualized embeddings also help in capturing syntactic and semantic relationships between words, allowing for a more nuanced representation of the text. Despite their advantages, contextualized word embeddings come with some challenges. Pretraining and fine-tuning these models can be computationally expensive and require large amounts of training data. The contextualized nature of these embeddings means that the representation of a word may vary based on its context, making it challenging to compare and interpret embeddings directly. Additionally, contextualized word embeddings are limited to the vocabulary and language patterns present in the pretrained data, which can impact their generalizability to specific domains or specialized texts.

3.2.5 Large language models

Large Language Models (LLM) are trained on massive amounts of text and can generate coherent and contextually relevant text. First, they are pretrained on diverse and extensive corpus, enabling them to learn rich representation. They capture both shallow and deep semantic relationships between words, contextual dependencies, and syntactic patterns. The size of these models, often comprising billions of parameters, allows them to encode vast amounts of knowledge and linguistic nuances. They can capture not only the meaning of individual words but also the overall coherence and structure of the text.

One notable feature of large language models is their ability to perform "zero-shot" or "few-shot" learning. These models can generalize to new and unseen tasks by leveraging their pretraining on diverse data. For example, they can be fine-tuned on specific outlier detection tasks with limited labeled data, achieving competitive performance without extensive task-specific training. However, it is worth noting that large language models also come with certain limitations. They can exhibit biases present in the training data, and the generated text may not always be accurate or aligned with specific domains or expert knowledge. The computational resources required to train and deploy large language models can be substantial, and real-time inference may pose challenges in certain scenarios.

The key difference between language models and large language models lies in the scale and capacity of the models. While language models can vary in complexity and size, large language models specifically refer to models that are designed to handle vast amounts of data and exhibit exceptional performance due to their large size. These models typically require extensive computational resources for training and deployment. It's worth noting that the terms "language models" and "large language models" are not always mutually exclusive. Large language models can be seen as a subset of language models that possess specific characteristics related to their size and capacity.

Example of LLM models include GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023) or Text-to-Text Transfer Transformer (T5) (Raffel et al., 2020). GPT-3 and GPT-4 are generative model known for their large size and impressive language understanding and generating capabilities while T5 follows a text-to-text transfer learning approach. The key idea of T5 is to pretrain on numerous task, then fine-tune for specific downstream tasks.

3.2.6 Discussion

Count-based and BOW-based models provide a simple and interpretable representation, allowing for efficient processing of large volumes of text data. However, they often overlook important semantic and contextual information, limiting their ability to capture complex outlier patterns accurately. Word embeddings build upon count-based models succeed in capturing semantic relationships between words, and have

Part-of-Speech	Description
Noun	Concrete or abstract entity highly represented in a language
Pronoun	Word or group of words that substitutes a noun
Adjective	A modifier that change the perception of a word
Verb	Verbs are words that present action, mood and basic aspects
Adverb	Modifier of an adjective or verb that makes language more precise
Preposition	Defines relationship between noun/pronoun and another word
Conjunction	Connector between words or group of words
Interjection	Express strong statements and feelings

TABLE 3.2: Table of all possible POS tags that are principally used in latin language. We can add to these eight POS article/determiner that marks (in)definiteness.

been widely adopted in various NLP tasks. They offer finer-grained representations and exhibit overall better performance than BOW models.

Contextualized word embeddings take representation a step further by considering the contextual information of words within a sentence or document. These models leverage large-scale pretraining and capture complex linguistic phenomena, leading to more accurate understanding of a language. Transformer-based models, such as BERT, have demonstrated exceptional performance in understanding and generating text, making them valuable for numerous tasks that require comprehensive representation and analysis.

It is important to note that the choice of representation model should be guided by the specific requirements of the task, the characteristics of the dataset and the available computational resources. Choosing a representation model needs to carefully consider the trade-offs between simplicity, interpretability and the ability to capture intricate semantic relationships and contextual information.

3.3 Outliers in text

Despite a strong interest in outlier detection in recent years, some types of data have had few contributions. Nowadays, this is the case for the study of outliers with textual data where few surveys and introduction works exist. Although this kind of applications is more and more represented, the formalism as well as the formal definition remains to be found. In Section 2.3, a definition of an outlier has been presented. The Section 2.2 presents how anomalies and outliers can find different definitions based on the application. However, it is interesting to ask whether the latter definition remains true with textual data. Considering the Definition 2.3.9 applied to text, several issues arise.

Definition 3.3.1 (Naive textual outlier). *A textual outlier is a document that is significantly different from the remaining documents.*

However, there is still a problem with this proposition since it does not assimilates characteristics of a textual data. As a reminder, a document is composed of terms that rigorously follow the rules of the language. Precisely, a spoken language admits

phones that defines the successive sounds, *phonemes* that are groups of phones that affect terms recognized by a practitioner, and *morphemes* that refers to the minimal meaning of a word. In written language, we prefer terms introduced in Section 3.2. Thus, words are grouped into *phrases* (verb phrases, noun phrases, adjective phrases, ...). In addition of those rules, there are two principal structure levels: *semantic* and *syntax*. While more precision regarding semantic are presented in this section, the syntax is less represented in outlier analysis literature and often refer to different tasks. Indeed, outlier analysis with syntax often implies a different goal where syntax is more important than semantic information, such as plagiarism detection. Taking the most commonly used representation of a term, namely words, documents are almost all different from each other. There are a huge number of words (vocabulary) in each language, and there are many different ways of representing a document.

Problem 3.3.1. *The documents of a corpus are structurally different from each other. Ocurence of multiple similar documents are considered duplicates.*

On a second level, terms can carry semantic meaning and/or information(s). Thus, a document manages to represent a set of information. One data mining task consists in searching and gathering such informations in the text for classification of the documents of a corpus. Then, the documents are associated with themes (sport, music, politics, etc.) which are called categories or topics. At this level, although the documents in a corpus are different in form, they may carry common informations. In this context, the definition of an outlier is:

Definition 3.3.2 (Topic outlier). *A textual outlier is a document associated to a topic that is significantly different from topics of the remaining documents.*

If the initial definition is problematic in its application to text because of the Problem 3.3.1 , this can be solved by using a higher level of representation such semantic. Notations for studying outlier detection with text are detailed Section 3.3.1 Then, we propose a focus of both syntax and semantic levels for outlier analysis in Section 3.3.2 and Section 3.3.3. Finally, we present our taxonomy for textual outliers in Section 3.3.4.

3.3.1 Notations

When referring to text data, the term *document* is often preferred. A collection of documents is called *corpus*. In the following, we use the same notations introduced in Section 2.1. Starting from here, a corpus X of N documents is written: $\mathrm{X} = \{\mathrm{x}_1, ..., \mathrm{x}_N\}$. The difference between a document and observations mentioned in Chapter 2 are the dimensions. A document corresponds to a sequence of symbols and punctuation, which rigorously follows language rules such as grammar. With latin languages, symbols refer to *letters*, and groups of letters to *words* or *entities*. The length of a document can be $\emptyset$ or any number of symbols.

Several types of document can be involved and it is common to observe many sources regarding the literature. Given a corpus, *a textual outlier is a document that*

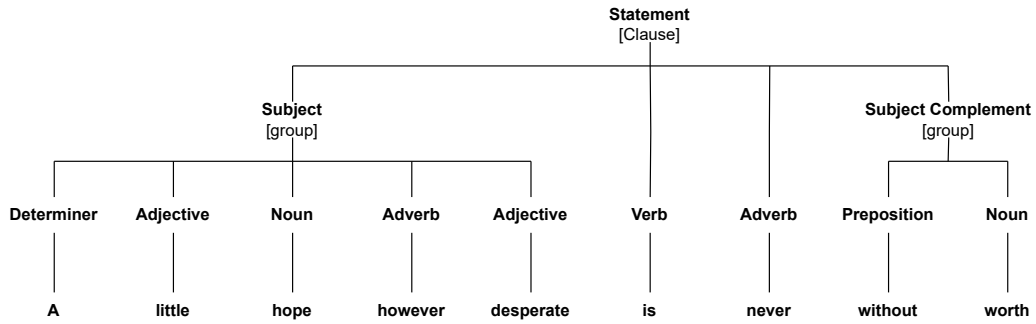


FIGURE 3.2: Tree parsing of part-of-speech processed with nltk and perceptron model. It is a simplified view without any additional information such as *3rd person* for verb.

is significantly different from the remaining documents. Following this definition we can apply it to news, for instance, where a *textual outlier is a document associated to a topic that is significantly different from topics of the remaining documents.* Let a sequence of M words $x_1, ..., x_M$ with a fixed vocabulary $\mathcal{V}$ of size $|\mathcal{V}| = V$. Each word is represented as $x_i \in \mathcal{V}$ for $i \in \{1, ..., M\}$ when the representation model is a kind of Bag-of-Words (BOW). Thus, x $\in$ X. Outlier detection is associated with One-Class Classification (OCC) where most of the time, the output is a score such as $s : \mathrm{X} \mapsto \mathbb{R}$. An outlier detection model tries to find the optimal number of divergent data from X while minimizing as much as possible false positives (Section 2.6).

3.3.2 Syntax level

Syntax is the part of linguistic that studies how morphemes and words are combined in sentences and, more globally, in the language (Manning and Schütze, 1999; Manning, Raghavan, et al., 2008; Nadkarni et al., 2011). Syntax dictates the grammar of one language with definition of rules on the position of lexemes (abstractive form of a word). In computational linguistic, syntax is often represented in a structural hierarchy derived from linguistic tools. Through grammar and linguistic theory, text can be processed automatic and unsupervised acquisition. One of the most popular approach is assignment of Part-of-Speech (POS) to each element of a document. Thus, each word is tagged as a grammatical property described in Table 3.2. One of the most used representation for grammatical structure is the Context-Free Grammars (CFGs) presented by Chomsky (1956). Such representation can be structurally displayed with trees or graphs for parsing purposes.

We propose an example of CFG in Figure 3.2 and we can observe that CFG can be used for grammatical parsing in natural language processing with a tree-like structure. The POS tagging step is one of the most critical part of grammatical parsing task. Recent success of machine learning methods for computational linguistic has seen numerous models applied for this step. POS is not limited to text, spoken language can be associated to similar structure and contribute to improve efficiency of multiple machine learning contributions. The example of the Figure 3.2 shows that pattern mining can be easily performed on top of POS tags.

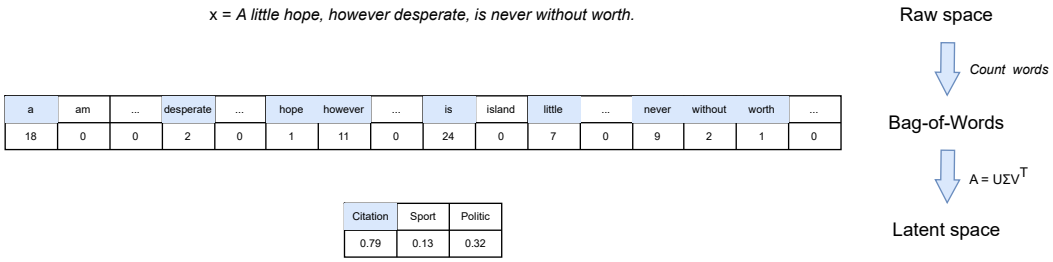


FIGURE 3.3: Semantic analysis of a document based on a corpus and natural language processing methods. The raw document is transformed in a BOW matrix before performing Latent Semantic Analysis with rank approximation of 3.

CFGs is an easy-to-use and efficient method for applying information retrieval, or any task, with a grammatical level. In the case of outlier detection, this level of features for text data is associated to plagiarism, authorship verification (Boukhaled and Ganascia, 2014) or similar tasks that require morphologic comparison. With this kind of application, the Definition 3.3.1 is preferred and the Problem 3.3.1 is not relevant. Instead, another kind of issue for exploring outliers is blossoming:

Problem 3.3.2. *Considering the case of documents that are written by a human, an outlier is close from a typo, for instance. We can imagine a typo in the final version of a book. Clear difference between an outlier and a typo can be difficult to prove for all data and depend on the application.*

Such problem is not limited to typos but also to the data and its inherent structure. The case of spam detection in mail requires to handle the special structure hierarchy of a mail (sender, object, head, body, signature, ...) (Karim et al., 2019) . With Definition 2.3.8 of an anomaly, the normal behavior of normal document rigorously follows the grammar and the different rules of the language. In this scenario, anomalies first handle such errors as an abnormal behavior while the definition of an outlier not necessarily. If the corpus has multiple documents that have typos but still follow similar patterns, they are considered normal. One example lies in poetry in which a French alexandrine is a syllabic poetic metre of exactly twelve syllables, not less or more.

We note that this section open up several questions. If we keep in mind that an outlier is linked to a population of observations (corpus), apparition of outliers can be independent of the normal or of the expected behavior.

3.3.3 Semantic level

When outlier detection is performed on textual data, relationships and dependencies should be taken into account. One of the growing interest in the community is the semantic property of text. The semantic analysis of text can differs from an application to an other but for outlier detection, it allows to extract topics and their weight in a data set. In linguistic, semantic is the exploration of meaning. Distinctly different

from syntax, semantic values the meaning to be communicated. Thus, meaning representation can be addressed through multiple techniques and theories of numerous background knowledges (cognitive science, logic, statistics, ...).

Semantic representation of text can take several forms and use different kind of methods. We can cite the use of syntactic trees such as Figure 3.2 in logic for inferring semantic interpretations. Another approach to semantic representation is based on psychology and cognitive semantic ignited in seventies and eighties. Works such as Talmy (2000) introduce and detail multiple representation through conceptualization, categorization and knowledge acquisition. In recent data mining literature, the statistical representation is often preferred with projection of bag-of-words matrices into latent spaces (see Section 3.2) or of the use of neural network language models. The Figure 3.3 shows a popular pipeline for performing semantic analysis of corpus.

Regarding outlier detection, the task is often associated with text classification where dataset are available. Those dataset have documents labelled with topics or concepts. In text classification, the class of one document is corresponding to different level of knowledge depending of the source of data. For instance, news articles are associated with a section or a subject of discourse such as *sport* or *politics*. In opposition of syntactic analysis, semantic requires a formal level for studying documents. This level allows to differentiate an approach that explores relationships between words and an approach that handle external knowledges (ontologies, for instance). With this context, a semantic outlier for news articles is different of semantic outlier for tweets. Considering that documents carry information about one or more topics, we assume that a hierarchy of topics from the corpus exists. This kind of hierarchy is needed in order to properly evaluate contextual outliers. In classification task, hierarchical approaches are not rare (Toutanova et al., 2001) and structure with two or three levels are common. It is then possible to easily identify the outliers that may appear depending on the application.

Topics (categories) are organized as a tree-like structure where they can be associated to at most one parent. In this setup, documents can only appear at leaves level, and can not therefore be parent of a topic. Let H a hierarchy that has $\mathrm{l} \in \mathbb{N}$ levels and $y \in \mathcal{Y}$ topics. Each elements $\mathrm{h} \in \mathrm{H}$ admit a set of children that leads to the level $\mathrm{l}+1$ that is either a topic, a leaf or $\{\emptyset\}$. If the raw space of representation is the text itself, we denote $\mathcal{X}$ such space. In text mining, text is often transformed in another latent space $\widetilde{\mathcal{X}}$ where hidden semantic structures can be easier to find. While the attributes of the instances of $\mathcal{X}$ would be intuitively words and numbers, in $\widetilde{\mathcal{X}}$ the latent attributes can represent topics.

While the Definition 3.3.2 proposes a difference with Definition 2.3.9, it is somehow limited to text classification. A better spelling of a semantic outlier is:

Definition 3.3.3 (Semantic outlier). *A semantic outlier is a document carrying a different subject, topic or meaning that remaining documents.*

We have seen that depending of the approach (symbolic, statistical, ...), meaning representation is prone to capture specific features. For outlier analysis, such representations can help to explore relationship between documents.

3.3.4 Taxonomy

In Section 2.4, several types of outliers have been proposed in the literature: Point outlier, Conditional/Contextual outlier and Collective/Group outlier. A similar taxonomy can be applied to text. In addition of point outliers, text is a type of data that is naturally contextual. Thus, multiple types of outlier often coexists among documents of one corpus. The definition of a topic seen in the previous section is assimilated to the subject that a document can address. Depending on the type of document, there may be several topics within the same topic (e.g. a sports topic that talks about football or tennis). Thus, taking into account this hierarchy, a type of contextual outlier appears which would be normal but considered outlier when associated with a small group of the corpus.

Collective outliers are difficult to formalize because of the contextual nature of text. For illustration: we imagine a legal document that mentions a football player and wrongly occurring in a sport corpus. Point outliers are observation with a topic that does not share any relationship with another topic. Precisely, outliers topic and inliers topic have different parents in the hierarchical structure of categories. Let a labelled document of a corpus $(\mathrm{x}, y) \in \mathrm{X} \times \mathcal{Y}$ and ζ be the inlier category, and its corresponding subset $\mathrm{X}_\zeta \subseteq \mathrm{X}$. We define $\mathcal{O}$ the subset of all outliers such as $\mathcal{O} \subset \mathrm{X}$. We have:

$$\mathcal{O} = \mathrm{X} \backslash \mathrm{X}_\zeta \tag{3.4}$$

Regarding $\mathcal{O}$, we can make the distinction with two different constraints. First, an observation $\mathrm{x_i}$ is considered to be an outlier if its parent topic is different of inlier parent topics such as:

$$\mathcal{O}_p(\zeta) = \{\mathrm{parent}(\zeta) \neq \mathrm{parent}(\mathrm{y}) | (\mathrm{o}, \mathrm{y}) \in \mathcal{O} \times \mathrm{Y}\} \tag{3.5}$$

The second constraint is corresponding to documents that do not lie in X_ζ but share the same parent topic as ζ. These observations are identified as another kind of outlier: contextual outliers. We write:

$$\mathcal{O}_c(\zeta) = \{\mathrm{parent}(\zeta) = \mathrm{parent}(\mathrm{y}) | (\mathrm{o}, \mathrm{y}) \in \mathcal{O} \times \mathrm{Y}, \mathcal{O} \backslash \mathcal{O}_p\} \tag{3.6}$$

3.4 Outlier detection approaches for text

In this section we propose a survey of the literature of outlier detection with text. In multiple cases, applications mentioned in Section 3.1.3 have methods compatible with outlier detection. Through one-class classification or semantic representation, these methods can be successfully used. We present methods of the literature that can perform outlier detection on text, starting by dimensionality reduction. The mentioned methods of Section 2.5 that are present in the literature are also introduced. Thus, distance-based approaches, density-based approaches, kernel approaches and

neural networks-based approaches are respectively studied. We complete the section with a discussion.

3.4.1 Dimensionality reduction

Dimensionality reduction addresses one of the problems of studying text for data mining: high dimensionality. The latter often prevents classical machine learning approaches from finding success for this type of data. LSA is one of the most popular dimension reduction methods for text. It computes a latent space from an occurrence matrix by performing low rank approximation. Although the rank parameter has to be chosen beforehand, the vectors of the obtained matrix allow to describe the text in a low dimensional space using Singular Value Decomposition (SVD). In this case, a chosen number of largest eigenvectors are kept.

PCA and LSA

C. C. Aggarwal (2017b) first presents the possibility of using PCA for textual data. Since BOW representations fail to detect *synonymy* and *polysemy*, because several terms have the same meaning. The aim of using PCA is to see all these words as noise and to keep only a single common version of them. The same principle is described by C. C. Aggarwal (2017b) where the latent representation of LSA can be judiciously coupled with the use of a distance-based method, for example. C. C. Aggarwal (2017b) also describes the use of a probabilistic LSA method, pLSA, and LDA. These methods reduce the dimension of a BOW matrix while treating noise and proposing a probability of membership of a text to a topic. In this context, C. C. Aggarwal (2017b) suggests that any method performing the topic modelling task can be used for outlier detection.

Non-negative matrix factorization

Other low rank approximation methods such as Non-negative Matrix Factorisation (NMF) can be used for outlier detection with text (Allan et al., 2008; Kannan et al., 2017). NMF assumes that the data and the components are non-negative. According to Berry et al. (2007), there are three general classes of NMF algorithm: multiplicative update, gradient/coordinate descent and alternating least squares. Allan et al. (2008) propose a multiplicative update NMF, based on a mean squared error objective function. Although the use of this method succeeds to get conclusive results, the approach seems to encounter difficulties when the volume of documents increases. The use of Block Coordinate Descent (BCD) is proposed by Kannan et al. (2017). In this work the experimental protocol and the formal definition of studied outliers are missing: their experimental protocol focuses on the notion of a weak topic among several strong topics. This issues the problem of a large representation of inliers against few outliers from a divergent topic. An interesting addition to this protocol may be the contamination with various topics.

Discussion

The advantages of the use of dimension reduction methods are notable for their contributions. All these methods aim to reduce the noise that persists in sparse and high-dimensional data. Depending on the adopted strategy, the semantic inherent problem with BOW representation is addressed. This is the case for LSA, pLSA and NMF but not for PCA which is mainly concerned with noise. Another advantage of these methods is the possibility of using different kinds of methods on top of the reduced representation, such as distance-based. However, these methods encounter difficulties when the size of the BOW matrix increases significantly (millions of documents) and they fail to correctly separate topics and associated terms when there exists a huge variety of possible associations (Allan et al., 2008). Their scalability may also be lacking in this context, but in the case of NMF there is also the choice of the objective function for the approximation.

3.4.2 Distance-based approaches

Distance-based methods are popular in a wide variety of fields, including outlier detection. Often, these methods are interested in the k nearest neighbors of an observation according to a distance metric such as the Euclidean distance or the Cosine distance. The natural hypothesis for performing outlier detection in this context is: a normal data is close to its neighbors while an outlier is far from them.

K-Nearest Neighbors

The use of K-Nearest Neighbors (KNN) is common and can be seen in many works for text (Kannan et al., 2017; Mohotti and Nayak, 2020; Koppel and Seidman, 2013; Ramaswamy et al., 2000). Koppel and Seidman (2013) proposes the use of KNN from a second-order similarity metric in plagiarism detection. Although a similarity metric differs from a distance metric in their range of values ($[0, 1]$ for distance and $[-1, 1]$ for similarity), KNN is used here as an aggregation function. Few contributions exist for outlier detection in text with distance-based approaches, the aim of Kannan et al. (2017) and Mohotti and Nayak (2020) is to compare their proposition with a similar distance-based approach. Kannan et al. (2017) concludes that KNN is highly sensitive to the distance metric and fails to perform well.

Relative distance scoring

F. Wang et al. (2018) propose three models that compute the distance from one document to another, from a document to a cluster, and from a document to the rest of the corpus. For the two first models, a threshold is processed and KNN is performed following the detailed distance. Although the outlier detection task focuses on a given corpus, it is often possible to apply novelty scoring methods to strengthen the approaches.

Discussion

Few textual outlier detection methods purely use distance-based approaches alone. The main motivation is the difficulty of distance metrics to correctly estimate a numerical semantic value for a text. Another drawback of distance-based methods is that they do not take into account the relative or local position for a fixed distribution. Density-based methods are interested in these notions of locality, and we propose to study them in the next section. The computation time of distances can also be a problem in a context with many documents. However, this type of method has interesting properties regarding the interpretation of the results thanks to a natural conceptualisation. Finally, this kind of algorithm can easily distinguish between noise and outlier, especially when coupled with a topic vector.

3.4.3 Density-based approaches

The main characteristic of distance-based approaches lies in the distance metric. This notion of distance corresponds to information linking documents together, but some problems persist. This is the case for positioning a document in a corpus. With this characteristic, we can determine whether an observation is close or not to a position where the other observations are gathered. The natural hypothesis which follows that of the Section 3.4.2 is: an outlier is located outside the dense area formed by the inliers.

Parametric methods

From a statistical point of view, the analysis of outliers is carried out by assuming the properties of a distribution to train a probabilistic density function. The use of mixture models in this context is often made for one of their properties which seek to represent the presence of a subset of documents among the entire corpus. Srivastava and Zane-Ulman (2005) propose an approach based on Gaussian Mixture Model (GMM) and PCA in order to reduce the BOW matrix. They also propose another method based on an Expectation Maximization algorithm. Although Srivastava and Zane-Ulman (2005) claims good results, the approaches do not rely on semantic parameters and is processed with a small corpus. Very recently, Ait-Saada and Nadif (2023) have proposed a novel approach based on GMM and word embedding, performing anomaly detection on short text data in french language. This kind of method perfectly match with the challenge of handling a low amount of documents and short texts.

DBSCAN

DBSCAN estimates density by counting the number of points in a fixed radius and considers two points to be connected only if they are in each other's neighbourhood. Unlike the distance-based approaches seen in Section 3.4.2, DBSCAN handles the relative position of an observation. Naturally, observations that reside outside the density area have a chance to be outliers. Works such as Tran Manh Thang and Juntae Kim (2011) and Celik et al. (2011) focus on the ability of DBSCAN to detect

outliers. Although the algorithm is very successful in a variety of domains and types of data, the textual outlier detection task has few contributions.

Local Outlier Factor

This method has quickly become fundamental and is very successful in many situations, including text. Walkowiak et al. (2020) uses LOF to distinguish documents incorrectly associated with clusters. These approach is also used with cosine distance as a comparison for the evaluation step (Lai et al., 2020). The success of this approach has led to many extensions that can also be applied to text. However, the insufficient number of works with textual data can be regretted.

Discussion

The density-based methods bring a notion of position with respect to the overall, or partial, distribution which is an undeniable asset for detecting outliers. Among these methods, the use of locality for the text manages to capture data that are moderately distant from each other (Walkowiak et al., 2020). However, the use of a distance metric is still hard to apply on text. For this reason, it is wise to use a semantic or dimension reduction method to refine the results (Walkowiak et al., 2020; Srivastava and Zane-Ulman, 2005). These methods have other limitations, especially when natural dimensions are not reduced. The distinction between low and high noise and outlier becomes less clear. It is difficult for this kind of methods to distinguish:

- anomalous documents that use similar words than inliers;
- outliers that target a topic closely related to other documents.

3.4.4 Kernel approaches

A Support-Vector Machine (SVM) constructs a hyperplane in a space that can be high dimensional. It defines a separation boundary using a hyperplane and there is a wide variety of applications and extensions of their usage, both linear and non-linear, in the literature. However, we are particularly interested in SVMs that take advantage of kernel techniques with the popular One-Class Support-Vector Machine (OCSVM) approach.

One-Class Support-Vector Machine

OCSVM is an approach that emerged early after the introduction of SVMs. Manevitz and Yousef (2001) have proposed the use of OCSVM in the text. They have studied the effectiveness of this method on four text representations with linear, polynomial, radial and sigmoid kernels. Shravan Kumar and V. Ravi (2017) recently proposed to use OCSVM coupled with a semantic representation of text using LSA. They add the reduction of dimensionality step comparing to Manevitz and Yousef (2001).

Discussion

The ability of OCSVM to handle high-dimensional data, such as BOW, without using a heavy pre-processing step is undeniably a great strength. When coupled with LSA, OCSVM is also successful in performing outlier detection task. Although the approach succeeds in correctly creating a decision boundary within documents of a corpus. When a large number of documents occurs, OCSVM encounters difficulties in limiting the number of candidate outliers. The choice of kernels and the parameters are also a problem when the documents do not come from the same creation process. To overcome these problems, the sub-sampling of the corpus as well as the use of several models is a solution to consider (C. C. Aggarwal, 2017a).

3.4.5 Neural Networks-based approaches

Recently approaches using neural networks have increased in popularity and efficiency. This is largely due to the scalability of the methods as well as the ability to handle high dimensional data. The outlier detection task is no exception to this rule and there are a large number of neural network approaches applied to outlier detection. However, textual data do not have the same appeal as for other types of data. In the following, methods for text are presented and they have been separated into two types: reconstruction and one-class classification.

Reconstruction

Reconstruction methods learn the characteristics of the observations in order to reproduce them regarding a distribution. Concerning outlier detection, an observation is considered outlier if the model does not manage to reconstruct this same instance properly. The model aims to minimise the reconstruction error from the decoder that work on the latent space initially obtained after encoding. Mei et al. (2018) propose a novelty detection method that uses an autoencoder to perform reconstruction from a semantic representation. Although the results show that the method lack stability over experiments, it manages to correctly avoid false positives. The approach proposed by Lai et al. (2020) uses a robust subspace recovery layer that seeks to extract significant subspaces where outliers would be difficult to locate. This technique facilitates the reconstruction stage of the decoder. The advantage of this kind of method is that it can be generalised to many types of data.

One-class classification

The outlier detection task can be assimilated to one-class classification, whose approaches seek to learn to characterise the observations of a distribution. Gorokhov et al. (2017) use a Convolutional Neural Network (CNN) with an RBF activation function and a logarithmic loss function. This approach is similar to an SVM except that the CNN is preferred to handle the BOW representation. In Section 3.2, language models that perform text representation have been presented. These models are built with neural networks and a small number of literature approaches exploit

them. Ruff, Zemlyanskiy, et al. (2019) propose to use these language models with a Context Vector Data Description (CVDD) that learns several semantic contexts via self-attention. More recently, Manolache et al. (2021) have introduced an approach based on ELECTRA (Clark, Luong, et al., 2020) which enforce two independent signals: one at token level and one at sequence level. Once the original model is trained, they processed E^3*Outlier* framework (S. Wang et al., 2019) for processing the anomaly score. This approach considerably outperform CVDD and have a better grasp of token level attention.

Discussion

Reconstruction models are highly dependent on the constitution of a corpus that presents an exhaustive set of examples. Although these approaches are robust to outliers, it is essential that the inliers are correctly gathered. For one-class classification methods, the contributions are similar to those seen with SVMs. Indeed, they follow a similar process in order to separate documents, although the use of multiple representations seems to correctly address the sub-sampling problem.

3.4.6 Discussion and problems

In recent years, the interest on outlier detection for text has grown and several contributions can be observed. We have seen in this section that various type of techniques and approaches have been proposed. Unfortunately there are still some kind of approaches that are poorly represented like Isolation Forest (IF) (F. T. Liu et al., 2008) and outlier ensembles combination (C. C. Aggarwal and Sathe, 2015). Few contributions propose to use such kind of approach with text for many reason such as comparable method, difficulty to address curse of dimensionality or also interpretability issues. Benchmark of various high-dimensional method is nonetheless possible at the moment where the representation of text has been reduced.

An observable problem through recent and older contributions is the protocol of experiments. From building an experimental dataset to evaluation of different methods, the complete process is often different from one contribution to an other. Independent outliers are almost always benchmarked through Reuters-21578 and 20 Newsgroups, with sometimes contextual outliers being part of benchmark but not differentiated of independent outliers. It is the case for Lai et al. (2020)'s protocol that integrates contextual outliers at the same time as independent. Without proper knowledge of strength and weakness against some kind of outliers, results may be poorly understood. In addition of the problem of understanding results, without a common protocol and common experimental setup, state of the art methods need to be reproducible each time.

Regarding text representation, various kind of approach are represented but we observe two different types of methodology. The first one consists to train the representation model from $\mathcal{X}$ with all available corpus and the other one from the prepared data. It can be noted that most of the time, splitting step of train and test data

Approach	Kind of approach	Model	Application	Text	Reference
Dimensionality reduction	Matrix factorization	NMF	Anomaly detection		(Allan et al., 2008)
		TONMF	Outlier detection	✓	(Kannan et al., 2017)
		l_2LSA	Outlier detection	✓	(Kannan et al., 2017)
		LSA	Outlier detection	✓	(C. C. Aggarwal, 2017b)
		pLSA	Outlier detection	✓	(C. C. Aggarwal, 2017b)
Distance-based	Nearest neighbors	KNNO	Outlier detection	✓	(Mohotti and Nayak, 2020)
		KNN	Outlier detection	✓	(Kannan et al., 2017)
		KNN	Plagiarism detection	✓	(Koppel and Seidman, 2013)
	Distance scoring	P2P	Novelty detection	✓	(F. Wang et al., 2018)
Density-based	Parametric methods	EMGMM	Anomaly detection	✓	(Srivastava and Zane-Ulman, 2005)
	Clustering	DBSCAN-MP	Anomaly detection		(Tran Manh Thang and Juntae Kim, 2011)
		DBSCAN	Anomaly detection		(Celik et al., 2011)
	Local density	LOF	Text classification	✓	(Walkowiak et al., 2020)
		LOF	Anomaly detection	✓	(Lai et al., 2020)
Kernel-based	SVM	OCSVM	One-class classification	✓	(Manevitz and Yousef, 2001)
		LSI-OCSVM	One-class classification	✓	(Shravan Kumar and V. Ravi, 2017)
Neural Networks-based	Reconstruction	AECB	Novelty detection	✓	(Mei et al., 2018)
		RSRAE	Anomaly detection	✓	(Lai et al., 2020)
	One-class classification	RBF-CNN	Anomaly detection	✓	(Gorokhov et al., 2017)
		CVDD	Anomaly detection	✓	(Ruff, Zemlyanskiy, et al., 2019)
		DATE	Anomaly detection	✓	(Manolache et al., 2021)

TABLE 3.3: Overview of the literature on compatible methods from anomaly detection, outlier detection, plagiarism detection, novelty detection and one-class classification. The approaches are categorized as seen in this Section, with their own sub categories. We provide the information if the approach has been originally introduced for text data.

is not processed. The main reason is that approaches are unsupervised and evaluation of trained model on a test split refers to a different task: novelty detection. While training the model of representation on prepared data without splitting train and test split has obvious advantages, weights of nearly unseen token are insignificant and out-of-vocabulary (OOV) tokens do not arise, the results are quite biased by the experimental setup.

In the Table 3.3 we introduce an overview of the literature. We can observe that anomaly detection is more popular for addressing one-class classification on text data than outlier deteciton. Another observation lies in the fact that outlier detection for text tends to use more older approach than for anomaly detection.

Regarding most of the methods of the literature, hierarchical property of topics and semantic is absent. Interestingly, the nature of text leads to observe complex relationships where data are often similar in structure but different at several level such as semantic. We do not observe contributions that focus on these properties of text and the hierarchical nature that topics can carry.

3.5 Evaluation of outlier detection approaches

Evaluation step in numerous task is important for highlighting strengths and weaknesses of a proposed approach. While the quantitative evaluation metric are the same for outlier detection in text than presented in Section 2.6, the difference lies in the data preparation.

3.5.1 Existing evaluations

In Section 3.3.4 we have introduce our taxonomy of outlier which can handle single and multiple outliers, and two different kind of outlier: point outlier and contextual outlier. As introduced in Section 3.1 there exist challenges when comparing different approaches of the literature. The principal reason lies in the experimental preparation of data which incurs contaminating a dataset X following a contamination rate (we note it ν). For recent works of the literature, the topic hierarchy presented in Equation 3.5 and Equation 3.6 is not handled. Most of reference works prepare the data in two different ways. It is either a cherry pick of several documents from few topics or a simple data contamination without holding any attention to which topic is contaminating inlier distribution.

For the former scenario, a special attention is given to choosing the right topics so that inliers greatly differ from outliers. Considering that inlier topics have documents with a different vocabulary from outlier topics, this scenario is often referring to point outliers (Equation 3.5). Although such a practice strongly incorporates and reinforces the bias problem, it still represent a practical scenario where inliers and outliers are highly different.

The second scenario only considers the Equation 3.4 in which $\mathcal{O}$ is the outlier subset of X considering an inlier topic ζ. While this scenario does not requires any selection step, each topic of the dataset can be used. This preparation integrates both

point outliers and contextual outliers. In such context, if the amount of documents in the training set is not high enough, results reproducibility can be hard to achieve. Comparing methods from literature can be challenging if no attention is given to the kind of contaminating outlier, the contamination rate and also the amount of data for training the approaches. While there exists several protocol for preparing data splits, we deplore the lack of variety in used corpus.

3.5.2 GenTO: Generation of Topic-level Outliers

Considering that documents may carry information about one or more topics, we assume that a hierarchy of topics from one corpus exists. This kind of hierarchy is needed in order to properly evaluate contextual outliers without dedicated dataset. In classification task, hierarchical approaches are not rare (Toutanova et al., 2001) and structure with two or three levels already exists. It is then possible to identify the outliers that may appear depending on the application. We present in this section how the datasets are chosen, and the data prepared. Thus, we introduce the characteristics that are researched for performing outlier detection with text data. In this section we present GenTO (Pantin et al., 2022), a method that prepares a dataset, in Section 3.5.2. GenTO is a generic approach that can be applied to any dataset with at least two different topics.

Mandatory characteristics of datasets

In Section 3.3.4, a taxonomy of textual outliers has been presented. Based on it, we target datasets of natural language processing that are compatible with the preparation of one or all kind of outliers. For the first type of outlier, the independent outlier, dataset with two classes are the minimal requirement. For text, independent outliers do not share any direct topic relationship with the inlier class. The researched characteristic is the difference of subject or information carried by the outlier.

The second type of outlier is the contextual outlier. Based on Section 3.3.4, we consider that topics (categories) are organized as a tree-like structure where they can be associated to at most one parent. We also add the constraint that a category is unique, and that it can not appear as child or parent twice. In this setup, documents can only appear at leaves level, and can not therefore be parent of a topic. Let H be a hierarchy, each elements $h \in H$ admit a set of children that leads to the next level of the hierarchy that is either a topic, a leaf or $\{\emptyset\}$.

For group outliers, semantic relationship can be difficult to work with. If text data can be easily contextual on a semantic point of view, group outliers are also contextual. It is important to separate independent group outliers from contextual group outliers. The former is rare with texts but can still occurs, while the latter is more likely dependent of an expert point of view. Indeed, if a group of data is considered as outlier against the inlier distribution, we can interpret such scenario as a new topic or a new generation of a data. Because this kind of outliers needs more

Algorithm 1 GenTO: Generation of Topic-level Outliers

Require: Inlier topic ζ, corpus X, split size l, contamination rate ν
Ensure: $0 < l \leq N$
  $c \leftarrow l\nu$
  $i \leftarrow 0$
  Initialize empty matrix Z
  $\mathcal{O} \leftarrow \{\mathrm{x}_j \times \mathrm{y}_j \in \mathrm{X} \times \mathrm{Y} | \forall j \in [0, N], \mathrm{y}_j \neq \zeta\}$ ▷ Outlier Matrix
  $\mathrm{X}_\zeta \leftarrow \{\mathrm{X} \backslash \mathcal{O}\}$ ▷ Inlier Matrix
  while $|\mathrm{Z}| < c$ **do**
    if Compare(Parent(y_i), Parent(ζ)) **then**
      Append($\mathrm{x}_i, \mathrm{y}_i$) to Z
    end if
    $i \leftarrow i + 1$
  end while
  Fill Z with X_ζ until $|\mathrm{Z}| = l$
  return Shuffle(Z)

attention, we propose to mainly focus on independent and contextual outliers for our experiments.

Also, we note that the preparation of the data can encounter different kinds of hardship regarding the proposed taxonomy. If we strictly focus on the label difference for preparing a dataset, it is interesting to note that the type of document may affect the value of such taxonomy. As an illustration, let say that our data are mails with one label that can be either *spam* or *ham*. In this setting, building independent outliers is possible but results of methods using LSA or OCSVM demonstrate good performances. Because the preparation of the data does not follows any semantic feature or label, results shows that the associated class is dependent of a topic. Thus, unknown characteristics of some datasets can be explored through the evaluation step.

GenTO

Based on Equation 3.5 and Equation 3.6 we propose the approach GenTO that generates outliers for text data. In the previous section we have defined two kinds of outliers: point and contextual. GenTO can be applied for each one of them with definition of a comparison function. Algorithm 1 describes GenTO and the comparison method Compare returns true if the outlier is either contextual or independent. The Compare function corresponds to Equation 3.5 for point outliers and Equation 3.6 for contextual outliers. There are three notables input parameter which are the inlier topic, the target split size and the contamination rate. GenTO allows to prepare numerous data split with different level of interpretation. Varying the split size can describe the how an approach succeed from a specific amount of available data. The same goes with the contamination which can demonstrate how an approach is robust with different level of contamination. These parameters are important for highlighting and fairly comparing approach of literature.

Dataset	Task	Documents	Vocabulary	Tokens (avg.)	Classes	Hierarchy
20 Newsgroups[1]	Classification	11000	24000	189	20	✓
DBpedia 14[2]	Classification	560000	152000	45	14	✓
Reuters-21578[3]	Classification	6500	9000	112	90	
Web of Science[4]	Classification	47000	41000	192	134	✓
Enron[5]	Spam Detection	33000	59000	238	2	
SMS Spam[6]	Spam Detection	5500	2600	15	2	
IMDB[7]	Sentiment Analysis	25000	35000	231	2	
SST2[8]	Sentiment Analysis	67000	12000	8	2	

TABLE 3.4: Presentation of datasets from the literature of outlier detection and inherent tasks. We describe these corpus with showing the document number that the train split contains. The size of the vocabulary and the average number of token in the documents (after stopwords filtering) are based on a naive preprocessing step with a BOW. The existence of a topic hierarchy in the original corpus labels is also specified.

3.6 Experimental study

In this section, conducted experiments on TOD with outliers generated with GenTO are presented. We describe dataset and how they are used, from preprocessing step to preparation. The evaluation metrics Area Under the Receiver Operating Characteristics curve (AUROC) and Area Under the Precision-Recall curve (AUPRC) are detailed in the second part. The complete baseline is introduced in the third point, in addition to their configuration. Finally, the results of our experiment is presented.

3.6.1 Data

Even though there are datasets dedicated to outlier detection, such as ODDS or UCI, they mainly provide multi-dimensional, time series and computer vision data. Applications like email spam detection and text classification have a rich set of available corpus. Recent works (Lai et al., 2020; Ruff, Zemlyanskiy, et al., 2019; Kannan et al., 2017; Mahapatra et al., 2012) use classification datasets such as Reuters-21578 and 20 Newsgroups with a dedicated preparation in order to benchmark their approaches.

Available datasets

In this chapter we have introduced one of the main problems of performing outlier detection with text, which is the lack of dedicated dataset. To address this problem, approaches in the literature perform special preprocessing on datasets from other tasks. The most targeted type of datasets are text classification datasets such as Reuters-21578 and 20 Newsgroups. However, although they can be prepared for performing and benchmarking outlier detection approaches, there still lack reference works focusing on the preparation procedure for text data. To be more precise, the literature of outlier detection with text is mainly concerned on introducing new approaches. Although this interest is important, the problem of evaluating the approaches judiciously is equally important.

For this reason, we have identified different kind of datasets among the tasks presented in Section 3.1.3. The Table 3.4 illustrates the most popular datasets that can be found in the literature related to outlier detection, spam detection, anomaly detection and novelty detection. These datasets are the principal materials that we plan to use among the conducted experiments on textual outlier detection. We plan to use datasets of text classification, spam detection and sentiment analysis. 20 Newsgroups and Reuters-21578 are the most popular datasets for outlier detection with text data. Because Reuters-21578 does not have any original hierarchy, we propose to use the topic hierarchy of Toutanova et al. (2001). In addition of those corpus, we propose to use DBpedia 14 and Web Of Science that both have a hierarchy of topic. While these last two corpus are not so popular in textual outlier detection, they are perfect for preparing outliers. Enron, IMDB and SST2 are other dataset that can be found in the literature but do not present any hierarchy of topic (original labels). We propose to use SMS Spam that is a popular corpus for spam detection. It has interesting features: spam data are often linked with a topic.

The selection of these corpus is motivated by the completeness of the evaluation of the approaches. What is missing in the literature is the usage of different kind of document with different sources. We can see in the Table 3.4 that the characteristics of the dataset differs from each other. It is also the case for the average length of documents (token number) and number of categories. Another important value is the size of the vocabulary that impacts the performances of some kind of approaches.

Preparation

We use the corpora presented in Table 3.4 and for each available category, we apply independent outlier and contextual outlier preparation with GenTO. To be fair with each method and dataset, we first set the preparation subset size to 350 and results are averaged through 10 runs. The value 350 is picked for corresponding to the overall available amount of document for each topic among the various corpora, and also for choosing a close value from the literature. The data are preprocessed with lowercase and stopwords removal. The train split of each corpus is used for training and the test split for evaluation. The TFIDF model is applied to the entire train set and only tokens that appear at least three times are kept in the vocabulary. At first, we set the contamination rate $\nu = 0.10$.

20 Newsgroups We separate subtopics between seven principal topics: computer, forsale, motors, politics, religion, science, sports. We do not count forsale topic for contextual outliers because it does not has any sub topics.

[1] http://qwone.com/~jason/20Newsgroups/
[2] X. Zhang et al. (2015)
[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[4] Kowsari et al. (2017)
[5] https://www.cs.cmu.edu/~enron/
[6] Almeida et al. (2011)
[7] Maas et al. (2011)
[8] Socher et al. (2013)

Reuters-21578 The corpus has documents associated with several topics. We remove all of these documents in order to keep those with single topic only. We reorganize topics in order to get a hierarchy, based on Toutanova et al. (2001) work. Thus, four parent topics are created: commodities, financial, metals and energy. We apply GenTO to the eight topics that have the higher number of train documents.

DBpedia 14 We create the topic hierarchy based on the provided ontology[9] and count six parent topics.

Web Of Science This corpus is often used in benchmark of hierarchical classification and provides three level of topic hierarchy. The third level topics are divided among the corresponding first level parents. Thus, seven parent topics are present and for child topics that are associated with more than one parent, we keep the largest child set and remove others.

Others For other corpora we prepare them with the independent GenTO and do not propose additional fine tuning.

3.6.2 Representation of text

For achieving a complete view of the literature, we introduce a comparative study on three representation of text: TFIDF, GloVe and Distill RoBERTA (Reimers and Gurevych, 2019; Y. Liu, Ott, et al., 2019). The three of them represent text using different approaches. Performing a comparative study around those representation models allow us to compare older model against recent works. Because the representation of text is one of the most important step, we make the hypothesis that recent works highly benefits from recent advances in this field.

3.6.3 Evaluation

Outlier detection is a task with highly imbalanced data and where inliers are predominant. As a consequence, average precision is often used to get a good idea of performance. The different representations are compared by means of the AUROC curve and the AUPRC. These classical metrics are derived from the confusion matrix and are both often used for the outlier detection task. The ROC curve displays True Positive Rate (TPR) on False Positive Rate (FPR) for many thresholds. Increasing or decreasing this threshold influences true positives with respect to false positives. It helps to choose the best threshold for the classifier.

The AUROC can be then considered as an accuracy metric. All experiments have been conducted on ten runs where Average Precision and AUROC are averaged. In this work we focus on outlier analysis for text data and it implies that, given a corpus, we have to detect documents that do not belong to a subset of topics. The evaluation

[9] mappings.dbpedia.org/server/ontology/classes/

step is then performed on prepared test split. Doing the evaluation step on test split is similar to test the robustness of a model to novelty.

While it can be deceiving, there exist a difference between an anomaly, a novelty and an outlier. An anomaly is an observation of interest, a novelty is a new kind of observation that requires models to be updated and an outlier is an instance that is frequently regarded as a data that should be removed (Ruff, Kauffmann, et al., 2021). Evaluating on test set highlights the chance of novelty to arise.

3.6.4 Baseline

Our data preparation is quite similar to Kannan et al. (2017), Lai et al. (2020), Ruff, Zemlyanskiy, et al. (2019), and Fouché et al. (2020) allowing us to perform our experiment on their implementation. We closely follow their advice and parameter recommendation in order to be as fair as possible. RSRAE (Lai et al., 2020) is setup with described parameters in their work, in addition with latent dimension set to 10, a learning rate of 0.00025 and 200 epochs. The architecture of our One-Class Autoencoder (OC-AE) is exactly similar with the one of Section 2.5.8. We set the same number of hidden layers for DeepSVDD, RSRAE and our OC-AE: $\mathcal{H} = [128, 64, 32, 32, 64, 128]$ which corresponds to the dimension for each hidden layers (first hidden layer has an input dimension of 128 and an output dimension of 64, etc ...).

For the implementation of LOF, Isolation Forest (IForest), OCSVM (Manevitz and Yousef, 2001), KNN (Ramaswamy et al., 2000), PCC (Shyu et al., 2003) and DeepSVDD (Ruff, Vandermeulen, et al., 2018) we use the PyOD (Zhao, Nasrullah, and Li, 2019) tool. The distance metric for LOF is cosine and the number of neighbors is set to 20, the same goes for KNN. We report better result with neighbors $\in [20, 30]$ but 20 seems to be one of the best stable value. Isolation Forest (IForest) (F. T. Liu et al., 2008) is a tree-based approach that scores outlierness of an observation against a corpus. We take the default setup proposed by the authors. Based on PCA, PCC is a robust principal component classifier that was originally evaluated against KDD'99 dataset. It computes a low dimensional hyperplane constructed by k eigenvectors and estimates outlier scores as the sum of the projected distance of an instance on all eigenvectors. We keep all components in our setup. We use the RBF kernel for OCSVM.

For comparison with works on low-rank approximation, we propose to use Latent Semantic Analysis (LSA) and Non-Negative Matrix Factorization (NMF). Experiments for both of them are conducted with Scikit-learn and the outlier score is processed on the transformed matrix with the $l2-$norm. For LSA, we chose the rank $r = 30$ and for NMF we set $r = 30$, these choices have been performed after several attempt with higher and lower ranks. The rank 30 is a middle ground value that perform well in most cases. Our implementation of NMF for outlier detection is similar to the one of TONMF, but instead of using the Block Descent Coordinate solver we use the Descent Coordinate one. This variant is more robust and has almost always

better results than TONMF. We set tolerance $= 1e-6$ and β-loss is computed with Frobenius norm. For both of them we process the l_2 norm over the low-rank matrix.

3.6.5 Results

Results are presented in the Table 3.5 presents the results of the baseline on all corpus with independent outliers and the Table 3.6 presents the results for contextual outliers.

Independent outliers

Starting with the results for independent outliers (3.5) and the TFIDF representation, we observe that IForest is the approach that under perform the most compared to the baseline. Globally, results are better on text classification corpus than others, particularly on sentiment analysis dataset (IMDB and SST2) which have the worst results. The hypothesis of Section 3.5.2 that supposes that binary classification corpus can be difficult to work with is true in this situation. Indeed, it appears that IMDB and SST2 are sentiment analysis corpora that do not rely to semantic relationship only. Often, negative statements and sarcasm are considered for performing such task. On the other side, the good performances observed on spam detection corpus strengthen the assumption that different applications of outlier detection share similar characteristics (see Section 3.1.3).

TFIDF Overall, we observe that half of the approaches have similar performances on independent outliers. If isolation forest records the worst results, k-nearest neighbours, latent semantic analysis and robust subspace recovery autoencoder are the best approaches. Right behind these last methods, we find local outlier factor, one-class support vector machine and one-class autoencoder that also achieve good results. Unfortunately, the AUPRC metric shows that they do not find true outlier as much as the best approaches. With the taxonomy presented in Section 3.4, distance-based methods and neural networks (reconstruction networks) seems to be the more stable approaches. The number of neighbours for KNN ($k = 20$) seems to be the correct setting for outlier detection with sparse matrix. Because we have a naive setup (we do not apply stemming or lemmatization for instance) for training our BOW model, the dimension of our data tends to be wide and the values of our features approximate 0. In such scenario, the distance metric is less exposed to bias and ambiguous tokens. We are aware of the existence of more popular methods for text representation (see Section 3.2) but bag-of-word allows us to be fair with old and recent approaches. For this reason, we did not compare to the results of Ruff, Zemlyanskiy, et al. (2019) and Manolache et al. (2021) because they are designed on top of a BERT-based language model.

GloVe The results on GloVe representation present notable differences on the area under the curve and on the average precision. Both of them are decreasing for all approaches and we can notice that KNN, OCSVM and PCC are performing the best with this representation. RSRAE is originally introduced with TFIDF representation

Independent										
TFIDF										
Model	**Metric**	**Newsgroups**	**Reuters**	**WOS**	**DBpedia 14**	**Enron**	**SMS Spam**	**IMDB**	**SST2**	**avg.**
LOF	auroc	0.844	0.771	0.849	0.888	0.671	0.753	0.542	0.545	0.732
	auprc	0.339	0.240	0.369	0.482	0.146	0.338	0.117	0.105	0.267
KNN	auroc	0.879	**0.950**	**0.970**	0.967	**0.776**	0.718	0.515	0.516	**0.786**
	auprc	0.364	**0.740**	**0.846**	0.791	0.326	0.335	0.111	0.093	**0.450**
OCSVM	auroc	0.761	0.877	0.932	0.919	0.731	0.644	0.538	0.558	0.745
	auprc	0.252	0.508	0.762	0.727	0.252	0.270	0.119	0.102	0.374
IForest	auroc	0.541	0.770	0.641	0.594	0.552	0.543	0.500	0.486	0.578
	auprc	0.136	0.295	0.194	0.180	0.114	0.191	0.100	0.113	0.165
PCC	auroc	0.649	0.863	0.868	0.734	0.609	0.720	0.543	0.519	0.688
	auprc	0.198	0.409	0.500	0.363	0.141	0.281	0.109	0.115	0.264
LSA_{l2}	auroc	**0.916**	0.841	0.951	0.931	0.741	0.764	**0.576**	**0.611**	**0.791**
	auprc	**0.484**	0.450	0.785	0.792	**0.333**	**0.411**	**0.128**	**0.126**	**0.438**
NMF_{l2}	auroc	0.717	0.936	0.790	0.500	0.502	0.712	0.503	0.557	0.652
	auprc	0.175	0.574	0.429	0.088	0.088	0.357	0.092	0.099	0.237
OC-AE	auroc	0.689	0.876	0.911	0.800	0.663	0.720	0.570	0.531	0.720
	auprc	0.241	0.436	0.630	0.492	0.169	0.281	0.122	0.121	0.311
DSVDD	auroc	0.615	0.730	0.635	0.654	0.590	0.603	0.514	0.531	0.609
	auprc	0.162	0.241	0.199	0.204	0.154	0.158	0.110	0.117	0.168
RSRAE	auroc	0.812	0.916	0.952	**0.974**	0.757	**0.767**	0.571	0.545	**0.787**
	auprc	0.283	0.539	0.768	**0.817**	0.262	0.345	0.122	0.099	0.404
GloVe										
LOF	auroc	0.666	0.783	0.875	0.852	**0.633**	0.568	0.527	0.515	0.677
	auprc	0.164	0.251	0.462	0.468	0.157	0.275	**0.128**	**0.130**	0.255
KNN	auroc	**0.703**	0.843	**0.910**	0.873	0.584	0.569	0.525	0.509	**0.690**
	auprc	**0.185**	**0.376**	**0.627**	**0.499**	0.142	0.277	0.127	0.125	**0.295**
OCSVM	auroc	0.684	0.837	0.883	**0.886**	0.575	0.568	0.527	**0.521**	0.685
	auprc	0.175	0.34	0.558	0.497	0.134	0.265	0.126	0.128	0.278
PCC	auroc	0.685	**0.844**	0.897	0.868	0.577	0.570	**0.528**	0.515	0.686
	auprc	0.176	0.357	0.586	0.475	0.134	0.268	**0.128**	0.126	0.281
SVD_{l2}	auroc	0.556	0.493	0.560	0.567	0.518	0.531	0.505	0.510	0.53
	auprc	0.149	0.124	0.188	0.219	**0.167**	0.145	0.120	0.123	0.154
NMF_{l2}	auroc	0.442	0.559	0.52	0.473	0.394	0.499	0.480	0.499	0.483
	auprc	0.114	0.190	0.172	0.159	0.099	0.123	0.113	0.116	0.136
OC-AE	auroc	0.556	0.737	0.655	0.599	0.536	0.523	0.503	0.501	0.576
	auprc	0.129	0.284	0.195	0.154	0.129	**0.302**	0.118	0.119	0.179
DSVDD	auroc	0.509	0.586	0.504	0.529	0.521	**0.576**	0.484	0.463	0.521
	auprc	0.123	0.158	0.141	0.153	0.128	0.155	0.110	0.11	0.135
RSRAE	auroc	0.623	0.735	0.776	0.774	0.582	0.558	0.513	0.513	0.634
	auprc	0.152	0.269	0.316	0.287	0.141	0.269	0.124	0.124	0.210
Distill RoBERTA										
LOF	auroc	0.880	0.768	0.938	0.984	0.730	0.569	0.540	0.562	0.746
	auprc	0.487	0.285	0.693	0.882	0.281	0.209	0.131	0.136	0.388
KNN	auroc	**0.955**	0.921	**0.982**	0.993	**0.747**	0.632	**0.544**	0.561	0.792
	auprc	**0.765**	0.652	**0.914**	0.951	**0.335**	0.384	**0.139**	0.141	**0.535**
OCSVM	auroc	0.948	0.917	0.981	0.993	0.723	0.693	0.539	0.575	0.796
	auprc	0.739	0.626	0.910	0.954	0.308	0.372	0.138	0.139	0.523
PCC	auroc	0.952	0.938	**0.982**	0.992	0.724	0.685	0.542	0.576	0.799
	auprc	0.742	0.681	0.908	0.946	0.317	0.383	0.139	**0.144**	0.533
SVD_{l2}	auroc	0.928	0.721	0.954	0.9	0.707	0.636	0.535	0.548	0.741
	auprc	0.632	0.317	0.789	0.690	0.245	0.368	0.126	0.137	0.413
NMF_{l2}	auroc	0.407	0.570	0.448	0.485	0.479	0.485	0.518	0.510	0.488
	auprc	0.099	0.168	0.129	0.132	0.123	0.116	0.126	0.118	0.127
OC-AE	auroc	0.697	0.732	0.856	0.837	0.592	0.514	0.517	0.499	0.656
	auprc	0.233	0.318	0.516	0.586	0.168	0.351	0.121	0.116	0.301
DSVDD	auroc	0.510	0.519	0.507	0.512	0.524	0.505	0.510	0.513	0.513
	auprc	0.168	0.137	0.179	0.168	0.140	0.144	0.118	0.123	0.147
RSRAE	auroc	**0.955**	**0.940**	**0.982**	**0.994**	0.731	**0.704**	0.540	**0.577**	**0.802**
	auprc	0.731	**0.690**	**0.914**	**0.956**	0.323	**0.388**	**0.139**	0.141	**0.535**

TABLE 3.5: Results of state of the art models for independent outliers with the contamination rate $\nu = 0.10$. Average precision (AUPRC) and Area under ROC (AUROC) are evaluation metric. For making the results easier to read, we provide a column that average the results of the corresponding rows. The experimental study is performed on three representation of text: TFIDF, GloVe and Distill RoBERTA.

Each result is performed on test split prepared through GenTO.

and perform well with very high-dimensional data. The GloVe space only presents 300 dimension. Neural-based approaches are poorly performing against distance-based approaches and density-based approaches. The most important observation we can do is that GloVe representation poorly perform on both text classification corpora and sentiment/spam corpora. We remove IForest from the benchmark for this representation and Distill RoBERTA because its results are too low.

Distill RoBERTA For the last representation of text, we can observe another change in metric values and also on successful approaches. RSRAE is the best performing approach on distill RoBERTA representation and fall behind KNN few times. We can observe that performance of the top approach from TFIDF and GloVe representation greatly benefits from Distill RoBERTA. The clear difference with TFIDF lies in the AUPRC which increases of almost 25%, meaning that RoBERTA give more robust detections to state of the art approaches. Surprisingly, for independent outliers, the text representation seems to not benefits that much from recent language models.

Contextual outliers

Table 3.6 displays similar trend among the ranking of the approaches. This time, KNN is the best method and is only lacking on one dataset: 20 Newsgroups. Overall, there is a noticeable drop in results with almost all approaches. Contextual outliers are more difficult to find than independent outliers. With KNN, the average AUPRC on text classification corpus and independent outliers is 0.685, against 0.434 for contextual outliers. Also, we find the same group of approaches for the ranking of the results but this time, both groups do not have near performances.

TFIDF This evaluation can also illustrate some characteristics of the data. For instance, articles from 20 Newsgroups are more difficult to handle than documents from Web of Science. While they have a similar text size (tokens number), the difference of results can be explained with the BOW model of Web of Science that is trained on more data than 20 Newsgroups. Also, the difference can come from the number of categories.

GloVe For GloVe representation we can observe a loss in the overall performances from the literature. Similarly to independent outliers, GloVe lack success against TFIDF and is also falling behind with top approaches. Approaches based on distance metrics are also preferred here with neural-based approaches falling behind.

Distill RoBERTA Similarly to independent outliers, the same increase of performance from AUPRC is observed. In addition, we can observe that distill RoBERTA is representation that greatly increase the success of all approaches. We can observe that this language model is particularly efficient for contextual outlier detection with textual documents. We can note that the robust PCC is the best approach for

Contextual										
TFIDF										
	Newsgroups		**Reuters**		**WOS**		**DBpedia 14**		**avg.**	
Model	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
LOF	0.186	0.707	0.118	0.570	0.239	0.758	0.204	0.700	0.186	0.683
KNN	0.192	0.725	**0.462**	0.748	**0.547**	**0.888**	**0.537**	**0.912**	**0.434**	**0.818**
OCSVM	0.153	0.644	0.420	**0.834**	0.514	0.861	0.490	0.861	0.394	0.800
IForest	0.110	0.527	0.202	0.613	0.146	0.580	0.137	0.567	0.148	0.571
PCC	0.135	0.587	0.233	0.656	0.268	0.734	0.231	0.656	0.216	0.658
LSA_{l2}	**0.253**	**0.782**	0.315	0.688	0.440	0.828	0.375	0.782	0.345	0.770
NMF_{l2}	0.127	0.638	0.400	0.788	0.332	0.759	0.075	0.500	0.233	0.671
OC-AE	0.146	0.607	0.245	0.669	0.312	0.764	0.290	0.706	0.248	0.686
DSVDD	0.118	0.545	0.163	0.623	0.151	0.580	0.141	0.591	0.143	0.584
RSRAE	0.158	0.664	0.400	0.784	0.434	0.840	0.462	0.854	0.363	0.785
GloVe										
	Newsgroups		**Reuters**		**WOS**		**DBpedia 14**		**avg.**	
Model	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
LOF	**0.137**	**0.576**	0.277	0.801	0.276	0.741	0.277	0.724	0.242	0.711
KNN	0.136	0.575	**0.431**	**0.866**	**0.306**	**0.758**	0.380	0.818	**0.313**	**0.754**
OCSVM	0.132	0.563	0.409	0.852	0.282	0.726	**0.383**	**0.829**	0.302	0.742
PCC	0.133	0.567	0.408	0.860	0.288	0.741	0.371	0.819	0.300	0.747
SVD_{l2}	0.127	0.519	0.122	0.411	0.146	0.518	0.180	0.525	0.144	0.493
NMF_{l2}	0.111	0.466	0.201	0.548	0.130	0.485	0.162	0.512	0.151	0.502
OC-AE	0.123	0.536	0.216	0.62	0.156	0.586	0.178	0.601	0.168	0.586
DSVDD	0.115	0.498	0.173	0.551	0.129	0.501	0.146	0.534	0.141	0.521
RSRAE	0.130	0.556	0.240	0.660	0.206	0.666	0.244	0.687	0.205	0.643
Distill RoBERTA										
	Newsgroups		**Reuters**		**WOS**		**DBpedia 14**		**avg.**	
Model	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
LOF	0.223	0.707	0.351	0.741	0.473	0.863	0.639	0.917	0.422	0.807
KNN	0.310	0.778	0.492	0.795	0.620	**0.900**	0.762	0.948	0.546	0.856
OCSVM	0.282	0.750	0.491	0.811	0.599	0.889	0.759	0.945	0.533	0.849
PCC	**0.314**	0.776	**0.518**	**0.828**	0.613	0.897	**0.771**	**0.954**	**0.554**	**0.864**
SVD_{l2}	0.250	0.722	0.305	0.686	0.433	0.826	0.411	0.763	0.350	0.749
NMF_{l2}	0.116	0.469	0.168	0.573	0.131	0.470	0.158	0.551	0.143	0.516
OC-AE	0.191	0.623	0.246	0.604	0.249	0.680	0.348	0.735	0.259	0.660
DSVDD	0.138	0.515	0.139	0.511	0.155	0.516	0.143	0.498	0.144	0.510
RSRAE	0.309	**0.779**	0.506	0.821	**0.621**	**0.900**	0.762	0.936	0.550	0.859

TABLE 3.6: Results of state of the art models for contextual outliers with contamination rate $\nu = 0.10$. Average precision (AUPRC) and Area under ROC (AUROC) are evaluation metric.

finding contextual outlier in text data. The dimensionality reduction performed by PCC is already handling outliers and thanks to distill RoBERTA it outperform other approaches.

Comparative discussion

We can observe that the representation of text is critical when designing an outlier detection approach for text data. Table 3.5 shows that for independent outliers, TFIDF perform well and succeed to get better results than GloVe. When comparing the results through AUROC and AUPRC, the main challenge that TFIDF is encountering is the out-of-vocabulary tokens. Such issue is highly mitigating when we use GloVe and Distill RoBERTA thanks to their contextual analysis of text. Also, Table 3.5 reveals that generic and reference approaches can be top contenders for independent outliers. In such cases, choosing TFIDF or Distill RoBERTA does not make a fundamental difference due to the nature of outliers. Outlying documents are not supposed to

Contextual								
	Distill RoBERTA				*GloVe*			
	Newsgroups		**Reuters**		**Newsgroups**		**Reuters**	
Model	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
KNN	0.310	**0.778**	0.492	0.795	**0.136**	0.575	**0.431**	0.866
OCSVM	0.282	0.750	0.491	0.811	0.132	0.563	0.409	0.852
PCC	**0.314**	0.776	**0.518**	**0.828**	0.133	0.567	0.408	0.860
RSRAE	0.158	0.664	0.400	0.784	0.130	0.556	0.240	0.660
CVDD	-	-	-	-	-	0,771	-	**0,969**
DATE	-	-	-	-	-	**0,832**	-	-

TABLE 3.7: Contextual contamination against Ruff, Zemlyanskiy, et al. (2019) and Manolache et al. (2021) with contamination rate $\nu = 0.10$. Average precision (AUPRC) and Area under ROC (AUROC) are evaluation metric.

share a similar vocabulary than inliers, leading to being more dependent from syntax differences.

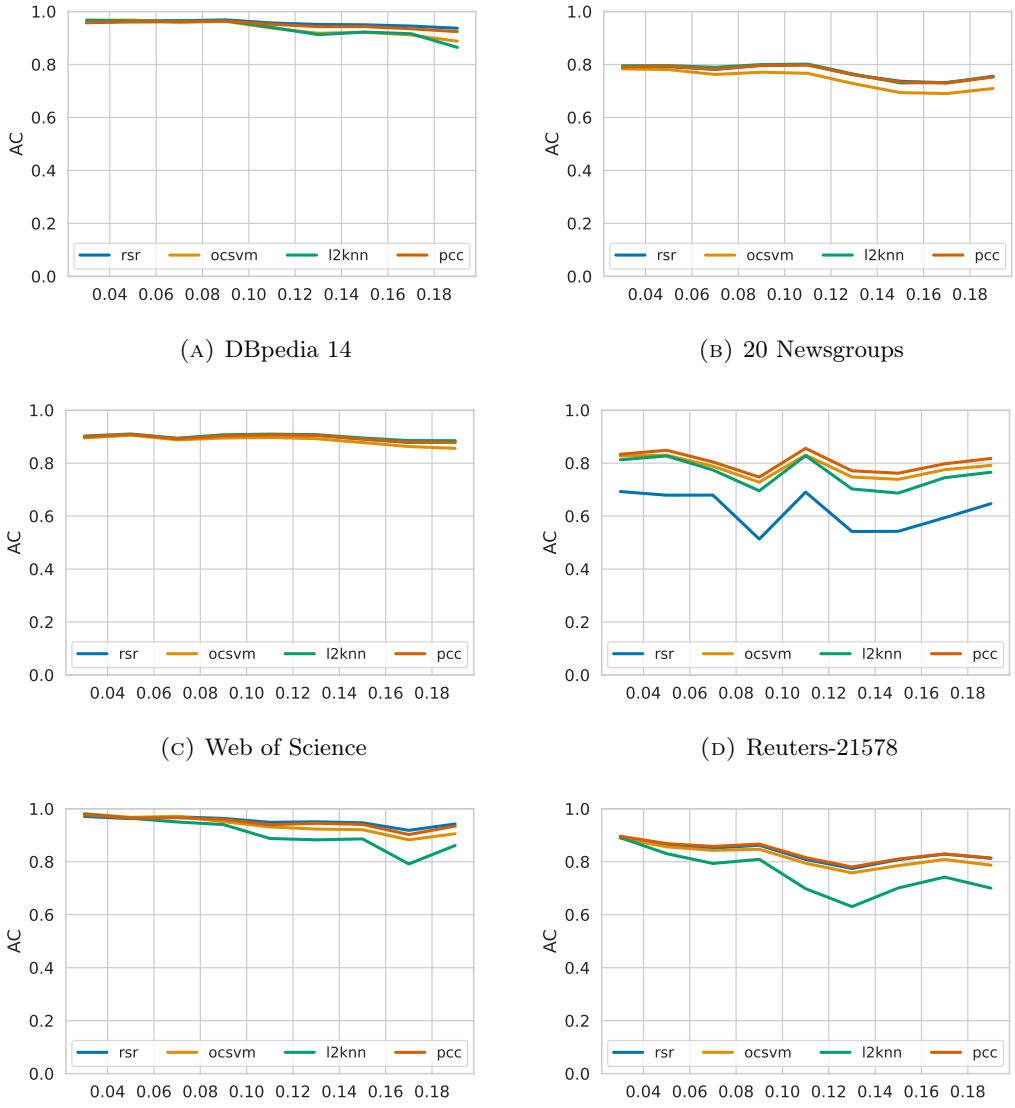
From the results we can observe that point outliers are easier to handle for most of the approaches than contextual ones. It is not surprising, and we can also add that 20 Newsgroups is the most difficult dataset. For neural networks, we can see that they are more robust against contamination rate. Kernel and distance approaches do not have great results while low rank approximation mechanisms find competitive results in most cases. We note that these latter are more robust to point and contextual outlier than other methods.

While the TFIDF and distill RoBERTA seems to be the best choices for performing outlier detection in text, we have to be alert about several parameters. One of them is the split size parameter from GenTO (Section 3.5.2) which can create a bottleneck on the inlier representation. While the number of inlier increase, it can be more difficult to detect outliers if the diversity of vocabulary increase. In this scenario, TFIDF is expected to fall behind against GloVe and distill RoBERTA.

Neural networks approaches can be penalized with a low amount of training data. The split size of 350 that we have set in GenTO can also be a problem for such approaches that benefits from large corpora. It can also be a challenge for dimensionality reduction approach which best perform with more reference samples.

On the other hand, we have not compared our results against recent text anomaly detection. The principal reason is the problem of reproducibility of their work. Thus, we propose to compare their recorded results against ours so that we can display their best results on corresponding corpora. Table 3.7 records the result that corresponding authors have shared. We can observe that they find success with GloVe representation and both outperform the literature. Considering RoBERTA representation, only DATE succeed to beat other approaches of the literature. We note that their protocol is similar to performing independent contamination and that the contamination is higher than $\nu = 0.1$ (Manolache et al. (2021) have recorded results on several contamination rates). We can observe that they do not perform AUPRC on their model, it is quite difficult to efficiently compare robustness of their approach against a low number of outliers. Considering that the experimental contamination of Ruff,

Zemlyanskiy, et al. (2019) is similar to independent contamination, approach of the literature a competitive against new and recent works (see Table 3.5). Also, it appears that results of both contributions are recording their evaluation on only two corpora, making further difficult compare their state of the art results. In our experimental results we can observe that four models have top performance on different corpora.



(A) DBpedia 14

(B) 20 Newsgroups

(C) Web of Science

(D) Reuters-21578

(E) DBpedia 14 (1000)

(F) Reuters-21578 (1000)

FIGURE 3.4: Analysis of ν for contextual contamination on the four best performing models from Section 3.6. The split size is set to 350 and the AUROC is (AC here) is displayed against ν. The text representation is Distill RoBERTA. We add two additional comparison on a split size of 1000 instead of 350.

3.7 Discussion

In the experimental study we have introduced the use of GenTO for performing a comprehensive study of literature's approaches. The results have been benchmarked against the area under the receiver operating characteristic curve and the average precision. Although pure performance record through evaluation metric can describe the success of an approach against another, we can add more explanation to the results thanks to varying GenTO parameters. Two main problems can be observed in the results displayed in Table 3.5 and Table 3.6: the training splits are set to 350, making greedy-data approaches fall behind, and contamination rate ν is set to 0.1 which is not ideal because real-world problems often occur an unknown amount of outliers. For this reason we propose to study both problem in Section 3.7.1 and Section 3.7.2. We also propose a short statistic study of the output of reference approaches in Section 3.7.3. Finally, we conclude with a discussion on the correlation between ν and the training sample size.

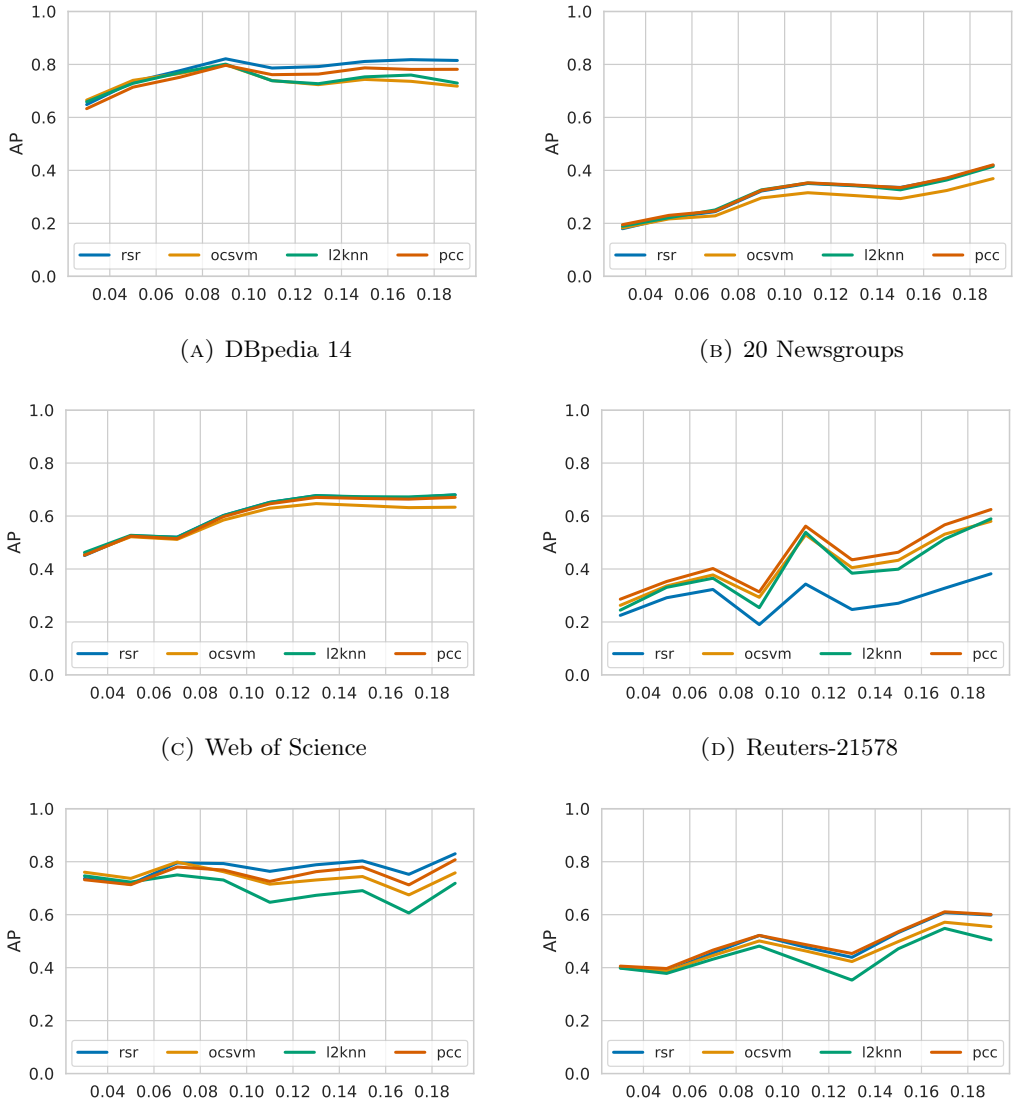
3.7.1 Influence of contamination rate ν

We introduce the study of the ν parameter in Figure 3.4 and Figure 3.5. Both average precision (AP in Figure 3.5) and area under the receiver operating characteristic curve (AC in Figure 3.4) are performed on a baseline of four approaches. Chosen approaches are RSRAE, OCSVM, KNN with l_2 norm and PCC. They are the top performing approaches presented in our experimental study (Section 3.6.5) and are categorized in different kind of approaches (see taxonomy in Section 3.4). The split size is set to 350 and Distill RoBERTA is used as text representation.

The AC (or AUROC) markedly represents how an approach succeeds to separate positive observations from negative observations (Section 2.6). With outlier detection, such metric tends to have a high value because outliers are very low amount (they can't be outliers if it was not the case). We can observe on all corpora that the AC is kind of high and slightly decreases when the ν contamination increases. As seen in Section 3.6.5 the PCC is best performing approach and is the most robust against contextual contamination. We can observe that RSRAE displays poor AC on Reuters-21578 corpus. Another observation is the decreasing results on Reuters-21578 corpus.

We propose to increase the split size to 1000, instead of 350, for DBpedia 14 (Figure 3.4e) and Reuters-21578 (Figure 3.4f). Both corpora are the only corpora that can perform contextual contamination with 1000 documents per inlier. In in this scenario, KNN performances are worst and RSRAE is the most competitive approach. As we hint such possibility in Section 3.6.5, neural networks often benefit from getting a large amount of instances. While doubling the number of training samples, the RSR autoencoder (Lai et al., 2020) outperform other approaches and succeeds with Reuters-21578 corpus.

The average precision is an important evaluation metric for imbalance data, it acts as a relative performance metric which depends of how an approach succeeds to



(A) DBpedia 14

(B) 20 Newsgroups

(C) Web of Science

(D) Reuters-21578

(E) DBpedia 14 (1000)

(F) Reuters-21578 (1000)

FIGURE 3.5: Analysis of ν for contextual contamination on the four best performing models from Section 3.6. The split size is set to 350 and the AUPRC is (AP here) is displayed against ν. The text representation is Distill RoBERTA. We add two additional comparison on a split size of 1000 instead of 350.

find outliers without falling against false positives. We can observe few differences with AC results: for DBpedia 14, RSRAE is outperforming other approaches and it is more evident to see that PCC is the more robust approach against contextual contamination. Another observations lies in RSRAE getting the best performance with an higher amount of training samples. RSRAE succeeds to correctly represent inliers with lower and higher amount of instances. From corpora point of view, 20 newsgroups is the hardest corpus to handle because its documents are long (see Table 3.4) and its topics are similar. The same observation can be proposed for Reuters-21578.

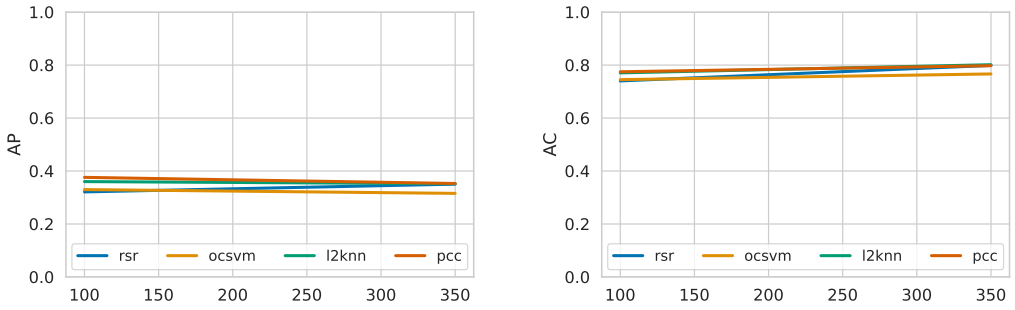
We can conclude that all approaches succeed to be stable against the contamination of a corpus. On the other hand, we observed that the amount of training sample is important and can impact the performance. Comparatively, PCC is the best approach for tackling real-world problems and seems to be robust against the size of a corpus.

3.7.2 Importance of inlier representation

Previously we have studied the influence of contextual contamination on reference approaches. One observation is that this parameter has a noticeable impact on the overall performances. While models tend to get better performance the more the contamination is, we have also observed significantly changes while increasing the amount of documents to train on (Figure 3.5e and Figure 3.5f). We propose to investigate furthermore the influence of the amount of documents in all training splits for contextual contamination of $\nu = 0.1$. The same protocol is applied, using GenTO, for four values of split size which are $\{100, 350, 1000, 5000\}$. All corpora do not have as much documents so the performances are processed when it is possible. Thus, all corpora are compatible with split size of 100 and 350, Reuters-21578 and DBpedia 14 are compatible with amount of 1000 documents and finally 5000 for DBpedia 14.

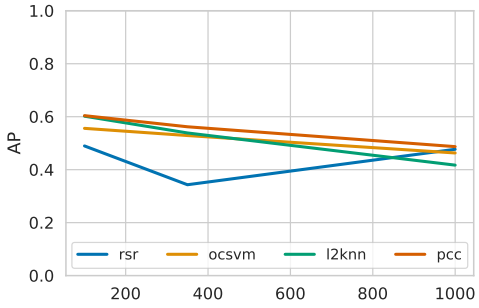
Table 3.6 records AC and AP for each models. The first observation is the almost no difference in performance for both 20 Newsgroups and Web of Science. For these corpora it seems that 100 documents or 350 is not a significant difference. Despite this we can note that RSR and OCSVM have a slight increase of their AP and AC for 20 Newsgroups which indicates that if there was more documents by inlier, results can be different. For performances recorded on Reuters-21578 and DBpedia 14 there are notable differences to note. According to the hypothesis stated in Section 3.6.5, increasing the amount of available data has a positive influence on neural-based approach RSRAE. On Reuters-21578 we also observe that RSRAE results are underperforming against other approaches until the amount of documents is increased to 1000. The observation is reversed when comparing KNN results, and confirms that distance-based approaches are great for a low amount of documents.

One conclusion is that the amount of documents for training an approach has a noticeable importance. For recent neural-based approaches we can observe that their experimental setup involves corpora with various sizes. We can also see a logical link between the motivation of using an optimization term similar to PCA, as seen in Equation 2.19, and the first RSR loss term $L_{RSR_1}(A) = \sum_{i=1}^{N} \|\mathrm{x_i} - A^\top A\mathrm{x_i}\|_2$ in Equation 2.37. We have observed that PCC is a robust approach against contextual contamination. On the other hand, contributions like Fouché et al. (2020) that uses distance or similarity metrics for optimizing early selection of inliers can relate to the success of KNN in our study.

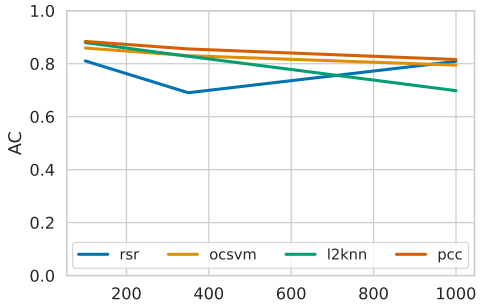


(A) 20 Newsgroups

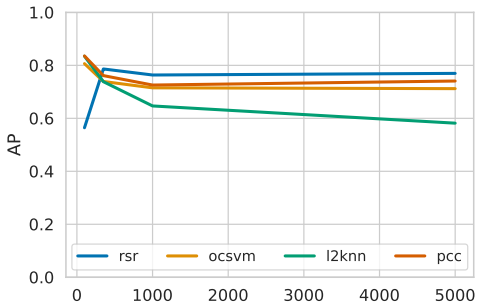
(B) 20 Newsgroups



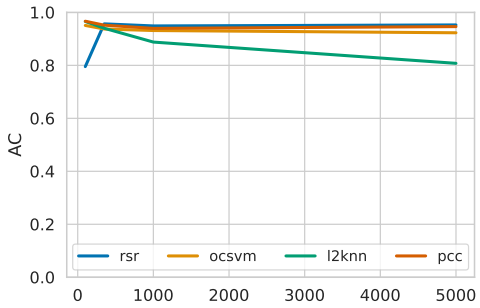
(C) Reuters-21578



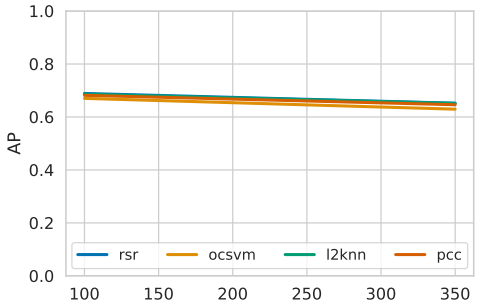
(D) Reuters-21578



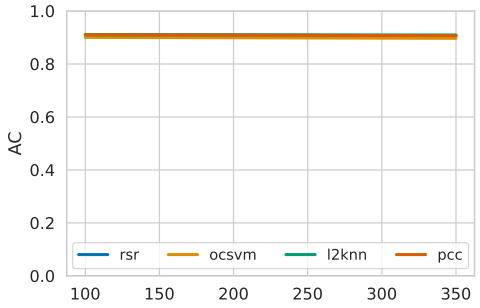
(E) DBpedia 14



(F) DBpedia 14

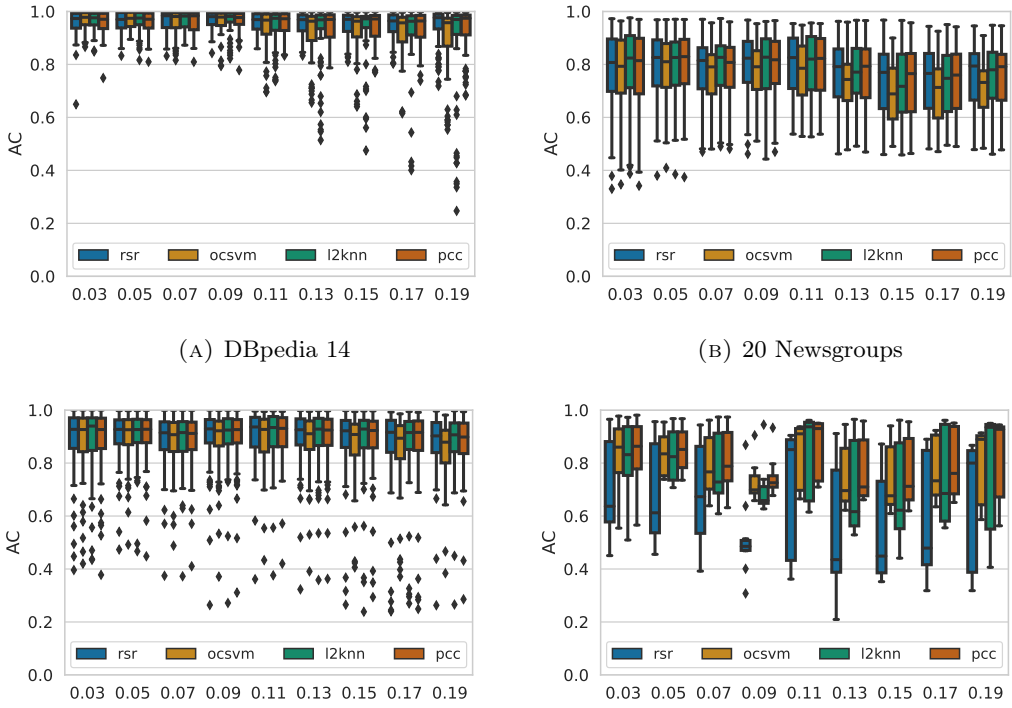


(G) Web of Science



(H) Web of Science

FIGURE 3.6: Analysis of split size $\{100, 350, 1000, 5000\}$ for contextual contamination on the four best performing models from Section 3.6. The ν contamination is set to 0.1 and the AUROC (AC here) is displayed against the split size. The text representation is Distill RoBERTA.



(A) DBpedia 14

(B) 20 Newsgroups

(C) Web of Science

(D) Reuters-21578

FIGURE 3.7: Presentation of AC performance of reference approaches through boxplots on different contextual contamination ν. The split size is 350.

3.7.3 On the bias problem

We present a statistical analysis of approaches results in Figure 3.7 through boxplots. A box plot, also known as a box-and-whisker plot, is a graphical representation that summarizes the distribution of a dataset. It provides a concise and informative way to visualize the central tendency, spread, and presence of outliers in the data. In the context of models, a box plot can be used to compare the performance of different models or visualize the distribution of evaluation metrics obtained from multiple runs of the same model. A typical box plot consists of a rectangular "box" that spans from Q1 to Q3, a horizontal line inside the box representing the median (Q2), and two "whiskers" extending from the box to the minimum and maximum data points within the 1.5*IQR (interquartile) range. Any data points beyond the whiskers are shown as individual points and are considered potential outliers.

By visualizing the distribution of evaluation metrics (e.g., accuracy, precision, recall, F1 score, etc . . .) using box plots, you can quickly compare the central tendency and variability of model performance across different models or different runs of the same model. Key insights that can be derived from box plots include:

- The position of the median within the box provides an estimate of the central tendency of the data.

- The length of the box (IQR) indicates the spread of the data. A wider box suggests higher variability, while a narrower box suggests lower variability.
- The presence of outliers beyond the whiskers can indicate extreme values or unusual cases in the data.

Box plots are particularly useful for comparing the performance of different models side by side, identifying models with consistent or varying performance, and gaining insights into the variability of model evaluation metrics. They provide a clear and compact summary of the distribution of data, making them a popular choice for data visualization and model evaluation.

Similarly to C. C. Aggarwal and Sathe (2015) and J. Chen, Sathe, et al. (2017) we present the AC results of RSRAE, KNN, OCSVM and PCC with boxplots in Figure 3.7. Our experimental study with GenTO comprises 10 runs on each compatible inlier candidates by corpus. It results in a large number of evaluation records, making the study through boxplots important for displaying the real consistency of a model against another one. First we can see that results on DBpedia and Web of Science display small boxes with increasing number of outlier as long as the contextual contamination increases. Overall, RSRAE is the most robust approach in these corpora, displaying less variance than others. For Reuters-21578, RSRAE is the worst model and often lies under an AC of 0.5 (random classifier). KNN is also an approach with high variance and is more unstable as the number of outliers in the corpus increases. Finally the results displayed on 20 Newsgroups are similar over each models.

What we can conclude of this analysis is that approaches are not performing similarly (as observed in Section 3.6). Also, as the contamination rate increases approaches encounter more hardship to get robust detection of outliers. Neural-based approach are have the more robust outlier scores over different contextual contamination, and distance-based approaches fall behind in this regard. Another observation is that all approaches display high variance against more difficult corpora. Such phenomena can be tackled through approaches like ensembles and is the subject of further analysis in the next Section 4.

3.8 Conclusion

In this section, we provided an overview of how to address the outlier detection task with text data. In Section 2, we presented the recent advancements in outlier detection. While this task has been explored across various types of data, we highlighted in Section 3 the lack of reference works detailing outlier detection with text data. Moreover, our analysis of the state-of-the-art literature revealed an ambiguous usage of the terms "anomaly" and "outlier," often interchangeably. By offering an inclusive overview of both outlier detection and anomaly detection, we were able to clarify their distinctions and similarities. Consequently, we established that outlier detection techniques are inherent in anomaly detection, novelty detection, out-of-distribution detection, and other related areas.

Within this context, we proposed an applied taxonomy that connects outlier detection in text data with other types of data. Notably, despite the preparation of corpora being similar among reference works, no dedicated contribution specifically addressing the issue of preparing an experimental setup for outlier detection with text data exists. To address this, we introduced a general algorithm called GenTO, which facilitates the generation of textual outliers based on two categories: point outliers (Equation 3.5) and contextual outliers (Equation 3.6). Our contribution addressed three key challenges in the literature: varying the contamination rate (ν) of a corpus, analyzing results with different sizes of the training set, and comparing the performance of approaches against both types of outliers in a fair manner. In our experimental study (Section 3.6), we benchmarked reference works from the literature using GenTO, enabling us to assess the approaches on eight corpora instead of the typical three in the literature. Additionally, we introduced a topic hierarchy on four corpora to facilitate to perform contextual outlier detection.

Since text representation is a critical aspect of working with text data, we compared selected representations from the literature with more recent ones. Our experimental study revealed that well-established and older approaches can benefit significantly from recent representations. Surprisingly, they also demonstrated competitiveness with recent works, indicating promising avenues for extensions in the context of text data.

In Section 3.7, we conducted an extensive analysis of the experimental results. For text data, approaches generally exhibited better performance and robustness against point outliers compared to contextual outliers. Consequently, contextual contamination (ν) and the number of training samples emerged as crucial characteristics for comparison with reference works. Furthermore, the lack of document samples proved to be problematic for some approaches, highlighting the significance of considering the contamination rate as a critical parameter for all contributions of outlier detection in text. Based on the experimental results, we introduced the problem of bias and variance in this context. Although this problem has been addressed with other types of data, it remains relatively unexplored for text data.

Based on this contribution, we can further investigate the use of *local* synthetic outliers instead of a *global* setting for outlier preparation. GenTO serves as a global setup for conducting experimental studies of outlier detection approaches. However, we aim to incorporate special cases and more qualitative outliers into the preparation setup. Including such outliers is akin to introducing local outliers for specific corpora, and it opens up avenues of exploration in domains such as explainable artificial intelligence (XAI) and counterfactual interpretability. Interestingly, this local approach has the potential to make classifiers more explainable and robust, specifically with recent advances in text representation.

Throughout our experimental study, we observed that several approaches encountered significant challenges under specific scenarios and setups. Handling multiple corpora with distinct characteristics for a model can be difficult, and hyperparameters play a crucial role in addressing this issue. While ensemble learning techniques

are commonly used to tackle this problem with other types of data, no contributions currently explore outlier ensembles with text data. In the next part, we address outlier ensembles with text data and introduce several key notions to perform qualitative analysis of one-class classifiers.

Chapter 4

Outlier Ensemble

In the previous chapters we have shown that there are several ways to perform outlier detection with text. In Chapter 2, the taxonomies of outliers, the origins as well as the different methods that allow to analyse these anomalies were introduced. It is in Chapter 3 that we propose a survey of the task for text. In the latter, we propose an application of the original taxonomy so that it can be applied to text. Although there are approaches in the literature that perform outlier detection in text, we observe very few works that deal with the use of ensemble approaches. Also, it is possible to see the work of Ruff, Zemlyanskiy, et al. (2019) which is interested in using another representation of the text than TFIDF. Despite ensemble approaches and representation of text can be combined for giving promising results, none work can be found for the textual outlier detection task.

The high-dimensional aspect of textual data implies to propose dedicated text representations. As introduced in Section 2.5.7, ensemble methods have the strength of limiting decision biases as well as offering robustness against more difficult datasets. Nevertheless, the advantage of working with text leads to question the usage of more text representation types. Indeed, one of the problems revealed in the results of Section 3.6.5 and the discussion of Section 3.7 is that the TFIDF representation has limits to handle all datasets. This is the case for SST2 and IMDB, both of which are datasets derived from sentiment analysis. However, Ruff, Zemlyanskiy, et al. (2019) manage to score very good results on anomaly detection with IMDB, using a one-class classification approach with a language model. Our results from the Chapter 3 and the previously cited have similar observations with 20 Newsgroups and Reuters, but their OCSVM is not used against IMDB. The addition of a more recent text representation model to our benchmark would thus increase performance on IMDB.

With the aggregation of several specialized representation of text, it is possible to get more hints on which characteristics of the text is important in the final decision of a classifier. For example, the problem of using a unique text vectorizer, such as word2vec, make it difficult to know if it is the topic or anything else that makes a text a more outlying point than others. Regarding this concern, the outlier detection task with textual data raises interesting problems for explainability. It can mitigate the explanation issues about a dataset or also about a classifier.

In this chapter we are addressing some of the issues let open in the previous chapter. Thus, we propose a different approach for representation of text, following

the statement that a textual data carries different kinds of information. Tasks such as data fusion, ensemble outlier or also multi-modal classification are important in this chapter. We present an example of fused representation with semantic and polarity features. Furthermore, we demonstrate, through experiments, that our representation can be applied to many types of texts (news articles, reviews, social media, etc.) while achieving state-of-the-art results. Also, we present how outlier analysis can positively contribute to explainability of models. The structure of the chapter is organised as follows. Section 4.1 presents data fusion and outlier ensemble, with additional information about text representation. Section 4.2 introduces REATO, our robust ensemble autoencoder that tackles outlier detection with text while mitigating bias-variance tradeoff. Section 4.3 presents PoLSA, our fused representation of text using early fusion with semantic and polarity features. Section 4.3.1 presents a comparison with existing approaches and highlights the benefits of our proposal. Section 4.4 presents an analysis of the predictions. Section 4.5 concludes the paper.

4.1 Outlier ensembles and fusion

In this section, approaches for outlier detection are presented, focusing on the case of textual data. A focus on approaches for outlier detection on texts is presented. This section introduces required knowledge for tackling outlier ensemble in our context and representation of text with multimodality.

4.1.1 Context

As introduced in Chapter 2, and following the Definition 2.3.9, *An outlier is an observation that is significantly different from the remaining data.* Often, outlier detection methods aim at defining the "normal" class in order to properly identify outliers. One-class classification (Manevitz and Yousef, 2001) methods are often used to do so. Outlier detection can be used to help models remove anomalous data during training and find decision boundaries. Often, OD methods are based on a hyper-parameter defined as the *contamination rate* (Ruff, Kauffmann, et al., 2021) that represents the ratio of outliers in the dataset. For real-world data, this ratio is unknown and can be difficult to value. Recent works introduce methods capable of performing outlier detection without such a hyper-parameter. The type of data is also an issue and high-dimensional data can be hard to work with (H. Liu et al., 2018).

For instance, outlier detection on texts still poses problems due to their specific characteristics: textual data are high-dimensional data with many complex underlying peculiarities (semantic, grammar, syntax, synonyms, ...) and they are sparse. Recent methods can partially solve these problems but specific approaches are still needed. One approach is to use dimensionality reduction techniques, as presented in Section 2.5.5, to keep meaningful features. We have seen that latent semantic analysis reduces the dimension of the term frequency matrix and highlights relationships between terms and topics.

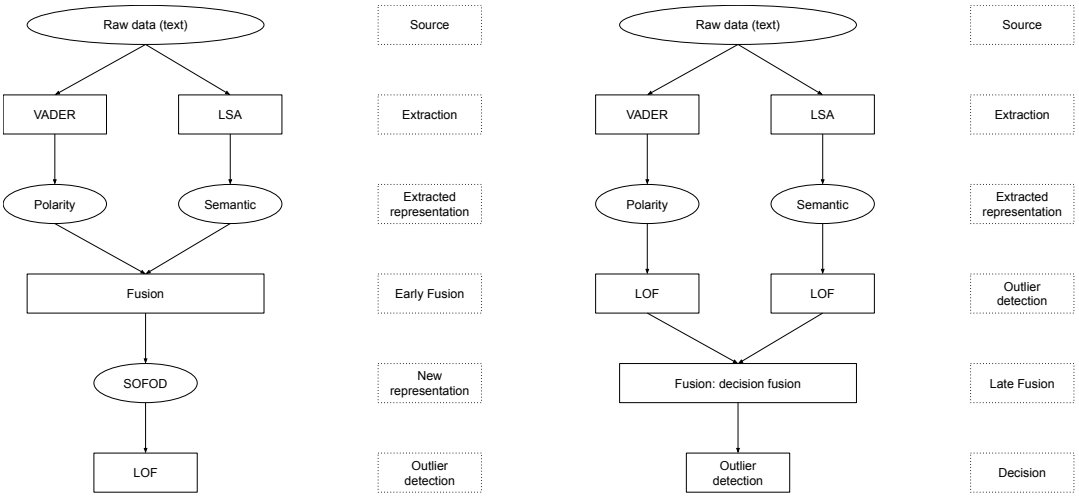


FIGURE 4.1: Examples of our case study with two representation of text: polarity for opinion mining and semantic for text classification. Left is early fusion and right the late fusion.

Another example of complex data are text with special features like emotions and sentiments are good candidates for representing specific features of a text (Medhat et al., 2014). Sentiment analysis is a field of study that focuses on people's feelings toward entities, peoples or topics (Medhat et al., 2014; B. Liu, 2012). Existing approaches identify polarity (*ie.* if a text/word is positive or negative) and subjectivity (Soleymani et al., 2017). These approaches consider features of the text useful for many classification problems and are often used on social media data (Yue et al., 2019; Vashishtha and Susan, 2019; Ruz et al., 2020). However, few approaches have been proposed for News (W. Zhang and Skiena, 2010) and much more less for outlier detection.

4.1.2 Data fusion and ensemble methods

Data fusion is a prevalent way to deal with imperfect raw data for capturing reliable, valuable and accurate information (Bleiholder and Naumann, 2009). It consists to integrate multiple data sources for mitigating issues of a unique distribution. There exists several surveys that highlight the principal characteristics of data fusion (Bleiholder and Naumann, 2009; J. Gao et al., 2020; Meng et al., 2020). Three categories of fusion are often addressed: early fusion (low), late fusion (high) and hybrid fusion (intermediate). They correspond to the stage at which the fusion takes place. For the early fusion, features are integrated immediately after their extraction. Intermediate fusion combines outputs from early fusion and individual predictors. Finally, late fusion performs integration after each of the predictors has made a decision. The Figure 4.1 displays the difference between early fusion and late fusion. For early fusion, it takes place after the text transformation, and for late fusion it takes place after each predictors.

Associated methods for the fusion process are disposed as follows:

1. Early fusion often consists to aggregate respective representations

2. Late fusion uses mechanisms such as averaging, voting schemes, weighting, learn model and many others

3. Intermediary fusion attempts to exploit the advantages of both in a dedicated framework

In such context, multi-modality of sources data can be naturally added and this is what J. Gao et al. (2020) are implying. Thus, multimodal data fusion aims to integrate the data of different distributions, sources, and types into a global space in which both intermodality and cross-modality can be represented in a uniform manner. Achieving early fusion of polarity representation (left diagram in Figure 4.1) with the semantic one is one kind of multimodal early fusion. With more sources and differences between the original data, the fused representation can have richer information. One drawback of early fusion is the risk to merge unwanted bias or ambiguous conflicts between sources. One example can be: the text "Yes .. very good" can have a sarcastic sense in twitter but not in blog (or vice-versa). Following this example, in absence of sarcasm modality in the final representation, the resulting representations may be contradictory.

With outlier detection, we often encounter outlier ensemble methods. This kind of methods is one of the most popular approach for outlier outlier analysis (C. C. Aggarwal and Sathe, 2015; C. C. Aggarwal, 2017a; Zimek, Campello, et al., 2014; Zhao, Nasrullah, Hryniewicki, et al., 2019; J. Chen, Sathe, et al., 2017). We have introduced it in the Section 2.5.7 and most of the time, a model solves its shortcomings by itself by adjusting its hyperparameters. Regarding the late fusion, we can observe some similarity, if not the same process, between these two types of approach. Thus, when we use our fused representation we are referring to the early fusion process and late fusion when presenting several predictors for on representation.

4.1.3 Outlier detection with ensemble approaches for text data

A short outline of Section 3 can be used for introducing challenges of outlier ensemble with text data. Recently an approach introduces a non-negative matrix factorisation method (Kannan et al., 2017) based on low rank approximation technique. Another successful technique consists in using additional contextual information (Mahapatra et al., 2012) with Latent Dirichlet Allocation (LDA). The Robust Subspace Recovery AutoEncoder (RSRAE) (Lai et al., 2020) takes advantage of autoencoders and is applied to textual and image data. One characteristic of this approach is that the knowledge of a contamination rate is not required, and the autoencoder assumes that the data is highly polluted. It succeeds in getting state of the art results for many outliers contamination thresholds on corpus. Several simple features from text can be efficiently used, such as author, genre, topic or emotional (D. Guthrie, L. Guthrie, et al., 2007).

Successful methods of outlier detection on texts encounter several issues such as interpretability (black box models), lack of diversity of document types (news, mails, sms) and sensitivity to very low contamination rate. Text representation is difficult

to interpret, whether one uses TF-IDF or language models such as BERT (Devlin et al., 2019). Neural network methods are not interpretable because of the combination of non-linear activation neurons on several hidden layers. TF-IDF is based on huge dictionary size (often several thousands) and this same characteristic makes it difficult to get high interpretation level.

Outlier detection methods for texts succeed to learn patterns in text and find anomalies but are difficult to generalise for all types. Dimensionality reduction can be applied on representation of text with low rank approximation techniques. An other type of textual features is the *polarity* of the opinion it contains. Usually, polarity is valued as positive, negative, or neutral and focuses on a different characteristic of text than topic. Often, it is studied in data mining on social media.

4.1.4 Latent Semantic Analysis

Singular value decomposition is performed on a bow (or tfidf) matrix and transforms it in a low rank approximated matrix of rank k. It is used to analyse the relationship between several documents with the hypothesis that words with similar meanings are found in similar texts. Regarding this last characteristic, LSA associates a single meaning to each term and has difficulty handling synonyms.

4.1.5 Sentiment Analysis

Several methods have emerged to perform sentiment analysis (Medhat et al., 2014). A first kind of approaches takes advantage of term frequency (TF-IDF) with algorithms like Naive Bayes, or Support Vector Machines (SVM) (Wawre and Deshmukh, 2016). A second type focuses on the definition of lexicons or lists of specific terms for a rule-based model (Hutto and Gilbert, 2014). Among the remaining approaches, neural networks have recently found great success (Ramadhani and Goo, 2017). Although most of these methods are used in the context of social media where documents are short, they can be applied to longer texts (Urologin, 2018). Despite the lack of approaches that use features from sentiment analysis for outlier detection on texts, some rare works (Savage et al., 2014) demonstrate that the use of sentiment can capture specific type of anomalies: abnormal polarised discourse. These anomalies are documents that are highly negative or positive as compared to normal documents.

4.2 Ensemble autoencoder approach for textual outliers

In this section, we consider the problem of applying ensemble methods for outlier detection on textual data. Outlier ensemble analysis decomposes the outlier detection error into two components which are the outlier score and the variance. The variance refers to the problem of an algorithm returning a divergent score when applied to different subsets of the baseline distribution. In most cases, the available distribution is generated from an unknown distribution of observations. Bias refers to the inconsistent range of scores of a model depending on the application. In the absence of a

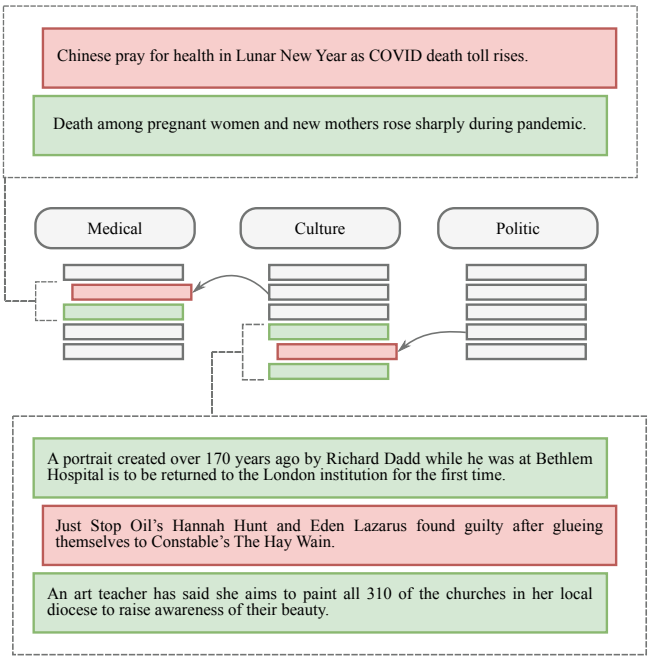


FIGURE 4.2: Presentation of the studied problem with three documents topics: medical, culture and politic. Under each topic we represent a textual document with colored rectangles. Gray and green are inliers and red ones are outliers. The detailed documents are the abstract of the news articles taken from sources like Reuters, New York times, BBC, ... The first scenario is the apparition of a culture-related document in a medical feed, and the second scenario is a political document in the culture feed.

label, the correct score cannot be returned and may also be misleading. The outlier set directly addresses these two components at different levels: data or model.

Outlier ensemble methods are among the most popular approaches to outlier analysis (Zimek, Campello, et al., 2014; C. C. Aggarwal and Sathe, 2015; C. C. Aggarwal, 2017a; J. Chen, Sathe, et al., 2017; Zhao, Nasrullah, Hryniewicki, et al., 2019). Most of the time, a model solves its own defects by adjusting its hyperparameters. As far as late fusion is concerned, we can observe some similarity, or even the same process, between these two types of approach. Thus, when we use our fused representation, we refer to the early fusion process, and to late fusion when we present several predictors for a representation.

4.2.1 Context

Performing outlier detection on textual data is less common than many other types of data (image, time series and medical) but it comes with several useful applications that helps discerning wrong web content, hateful message, spam or also errors in news feed. The difficulty to reproduce experimental protocols and results from the literature is one of the reason of the unpopularity of the task with text. Indeed, there is a great difference between tackling independent outliers and contextual outliers (Mahapatra et al., 2012; Fouché et al., 2020) using semantic in text. For the former, the classifier

needs to differentiate two kinds of documents that come from unrelated topics (sports and computer) but for the latter, one topic is contaminated with another "sibling" topic. The Figure 4.2 describes such scenario. Most of the recent works are contaminating corpora without addressing the problem of which kind of anomaly/outlier is added (Manevitz and Yousef, 2001; Kannan et al., 2017; Ruff, Zemlyanskiy, et al., 2019; Lai et al., 2020).

Recent advances in word embedding with language models like GloVe (Pennington et al., 2014), Fast-Text (Bojanowski et al., 2016), BERT (Devlin et al., 2019) or RoBERTa (Y. Liu, Ott, et al., 2019) have shown promising characteristics for outlier detection. Only few methods of the literature propose their usage (Ruff, Zemlyanskiy, et al., 2019; Manolache et al., 2021). Other methods like One-Class Support Vector Machine (OCSVM) (Schölkopf, Platt, et al., 2001) and Textual Outlier using Non-negative Matrix Factorization (TONMF) (Kannan et al., 2017) rely on tf-idf. On the other hand, recent methods are not using outlier ensemble methods (Zimek, Campello, et al., 2014; C. C. Aggarwal and Sathe, 2015; Zhao, Nasrullah, Hryniewicki, et al., 2019) for performing outlier detection with text data. Additionnaly, AutoEncoders (AE) have been used for anomaly/outlier detection with high-dimensional data (J. Chen, Sathe, et al., 2017; Kieu et al., 2019) and are also successful with other kind of data (An and S. Cho, 2015; C. Zhou and Paffenroth, 2017; Z. Chen et al., 2018; Lai et al., 2020), but the risk of using autoencoders with language models is the apparition of degenerate solution in the learning step. Robust properties are needed in such scenario.

We introduce a novel outlier ensemble method that performs outlier detection on text using word embedding and a Robust Subspace Recovery (RSR) (Rahmani and Atia, 2017; Lerman and Maunu, 2018) layer. The autoencoder use the RSR layer for mapping the normal distribution in a subspace where outliers are at the edge (Lai et al., 2020). Our method, called Robust subspace recovery Autoencoder ensemble for Text Outlier (REATO), build a RSRAE (Lai et al., 2020) ensemble whose are randomly connected. RSRAE are a kind of robust autoencoders which aim make the assumption that outliers are in low-dimensional subspaces. REATO can also be seen as an ensemble of several subspace that aims to find normal data with different *manifold*. In short, such learning method are making the hypothesis that the distribution is highly contaminated and the inliers (normal data) lie in a low-dimensional subspace. The performance of REATO are experimented against other state of the art methods on a total of eight corpora. We are proposing a definition of two different outliers that can be applied on available corpora and REATO outperforms the literature with more robust results.

Our autoencoder has the characteristic to perform local neighboring in in its manifold. As we have introduced an outlier taxonomy in Section 3.3.4 for textual data, this characteristic is particularly efficient for finding contextual outliers. Thus, experimental results are performed on contextual outliers.

4.2.2 REATO: Robust subspace recovery ensemble autoencoder for text outliers

This section presents our approach, REATO, and the description of its properties. While robust subspace recovery autoencoders have successfully tackle anomaly detection with text (see Section 3.6), they lack locality and geometry awareness for mitigating manifold collapse in transformer-based language models. For this reason we introduce Robust subspace recovery Ensemble Autoencoder for Text Outliers (REATO) which integrates locality in the latent representation through locally linear embedding technique.

The section is structured with a presentation of the randomly connected autoencoders, followed by a presentation of RSR loss. We then introduce the locally linear embedding loss term of REATO before presenting its ensemble method. Finally, we present the representation of text.

Randomly Connected One-Class Autoencoder

Instead of using fully connected autoencoders, we propose to use randomly connected autoencoders. In the case of RSRAE, it is a novel approach and allow us to build ensemble autoencoders with different base detectors.

Let X be a dataset of N instances such as $\mathrm{X} = \{\mathrm{x}_1, ..., \mathrm{x}_\mathrm{N}\}$. Each instance has D dimension which correspond to its attributes: $\mathrm{x_i} = \{x_1, ..., x_D\}$. An Autoencoder (Section 2.5.8) is a neural networks in which the encoder $\mathcal{E}$ maps an instance $\mathrm{x_i}$ in a latent representation noted $\mathrm{z}_i = \mathcal{E}(\mathrm{x}_i) \in \mathbb{R}^e$ of dimension e. The RSR layer is a linear transformation $\mathbf{A} \in \mathbb{R}^{d \times e}$ that reduces the dimension to d. We note $\hat{\mathrm{z}}_\mathrm{i}$ the representation of $\mathrm{z_i}$ through the RSR layer, such as $\hat{\mathrm{z}}_\mathrm{i} = \mathbf{A}\mathrm{z_i} \in \mathbb{R}^d$ The decoder $\mathcal{D}$ maps $\hat{\mathrm{z}}_\mathrm{i}$ to $\hat{\mathrm{x}}_\mathrm{i}$ in the original space D. The matrix $\mathbf{A}$ and the parameters of $\mathcal{E}$ and $\mathcal{D}$ are obtained with the minimization of a loss function.

Similarly to J. Chen, Sathe, et al. (2017) we introduce autoencoders with random connection such as we increase the variance of our model. In the autoencoders ensemble each autoencoder has a random probability of having several of its connections to be cut. Thus, we setup the probability disconnection with a random rate between $[0.2, 0.5]$.

Robust Subspace Recovery Layer

The RSR autoencoder follows the reconstruction problem detailed in Section 2.5.8 which aim at generalize the original data in a lower representation and learn to reconstruct it through an optimization problem. We detail the original RSRAE reconsutruction loss (Lai et al., 2020) presented in Section 2.5.8. The loss function minimizes the sum of the autoencoder loss function noted L_{AE} with the RSR loss function noted L_{RSR}.

$$L^p_{AE}(\mathcal{E}, \mathbf{A}, \mathcal{D}) = \sum_{i=1}^{N} ||\mathrm{x_i} - \hat{\mathrm{x}}_\mathrm{i}||_2^p \tag{4.1}$$

which is the $l_{2,p} - norm$ based loss function for $p > 0$.

For performing the subspace recovery, we denote two terms that have different roles in the minimization process. The first term enforces the RSR layer to be robust (PCA estimation) and the second enforces the projection to be orthogonal:

$$L^q_{RSR}(\mathbf{A}) = \lambda_1 \sum_{i=1}^{N} ||\mathrm{z_i} - \mathbf{A}^{\mathrm{T}}\hat{\mathrm{z}}_\mathrm{i}||_2^q + \lambda_2 \sum_{i=1}^{N} ||\mathbf{A}\mathbf{A}^\top - \mathbf{I}_d||_f^q \tag{4.2}$$

with $\mathbf{A}^\top$ the transpose of $\mathbf{A}$, $\mathbf{I}_d$ the $d \times d$ matrix and $||\cdot||_f$ the Frobenius norm. λ_1 and λ_2 are hyperparameters and $q = 1$ is corresponding to the optimal $l_{p,q}$ norm (Maunu et al., 2019). If we simplify Equation 4.2 we have:

$$L_{RSRAE}(\mathcal{E}, \mathbf{A}, \mathcal{D}) = \lambda_1 L^1_{AE}(\mathcal{E}, \mathbf{A}, \mathcal{D}) + \lambda_2 L^1_{RSR}(\mathbf{A}) \tag{4.3}$$

Locally linear embedding term

Locally Linear Embedding (LLE) (Roweis and Saul, 2000; J. Chen and Y. Liu, 2011) is a popular nonlinear dimensionality reduction technique that aims to preserve the local geometry of the data in a lower-dimensional subspace. It is based on the assumption that data points in a local neighborhood can be linearly represented by their neighboring data points. The LLE term in the loss function encourages the autoencoder to learn representations that preserve the relationships between data points in their local neighborhoods. By doing so, it helps to project the Euclidean distance with its neighbors in the learned subspace. Based on Equation 4.2, the reconstruction loss function of RSRAE enforces robustness with L^1_{AE} and the orthogonality with L^1_{RSR}. Because the learned representation of the encoder is compressed in a e dimension space, the locality of the subspace is not handled.

For tackling this problem, we propose to introduce a third term to L_{RSRAE} based on locally linear embedding. Given a set of data points $\{\mathrm{x_i}\}_{i=1}^N$ in the input space, the goal of LLE is to find a lower-dimensional representation $\{\mathrm{z_i}\}_{i=1}^N$ in the output space (the subspace learned by the autoencoder) such that the local relationships between data points are preserved. We note:

$$L_{LLE}(\mathbf{A}) = \sum_{i=1}^{N} \left\| \mathrm{x_i} - \sum_{j \in \mathcal{N}_i} w_{ij} \mathrm{x_j} \right\|_2^2 \tag{4.4}$$

where $\mathcal{N}_i$ represents the set of indices of the k-nearest neighbors of $\mathrm{x_i}$ (excluding $\mathrm{x_i}$ itself) and w_{ij} are the weights assigned to the neighboring data point $\mathrm{x_j}$ in the linear reconstruction of $\mathrm{x_i}$. The weights w_{ij} can be computed using the least squares method to minimize the reconstruction error: $\min_{\mathbf{w_i}} \left\| \mathrm{x_i} - \sum_{j \in \mathcal{N}_i} w_{ij} \mathrm{x_j} \right\|_2^2$ subject to the constraint $\sum_{j \in \mathcal{N}_i} w_{ij} = 1$.

The LLE term encourages the autoencoder to find a representation for each data point as a linear combination of its k-nearest neighbors in the input space. By minimizing the LLE term in the loss function, the autoencoder learns to preserve the local

linear relationships, which ultimately helps to project the Euclidean distance with its neighbors in the learned subspace. Our loss function is comuted as follows:

$$L_{REATO}(\mathcal{E}, \mathbf{A}, \mathcal{D}) = L_{RSRAE}(\mathcal{E}, \mathbf{A}, \mathcal{D}) + \lambda_3 L_{LLE}(\mathbf{A}) \\ := L_{RSRAE}(\mathcal{E}, \mathbf{A}, \mathcal{D}) + \lambda_3 \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} w_{ij} \|\mathbf{A}\mathrm{x_i} - \mathbf{A}\mathrm{x_j}\|_2^2 \tag{4.5}$$

In Equation 4.5, the new term $L_{LLE}(\mathbf{A})$ represents the locally linear embedding term, which measures the preservation of local neighborhoods. The weight w_j assigned to the neighbor $\mathrm{x_{ij}}$ in the local linear reconstruction of $\mathrm{x_i}$ are determined based on the distance between data points and their neighbors. The inclusion of the LLE term in the loss function encourages the autoencoder to preserve the local geometric structure of the data in the learned subspace. The parameter λ_3 controls the influence of the LLE term on the overall loss. Because it controls the influence of locality of the manifold the term is preferred to be low for avoiding collapsing results.

Ensemble Learning

The main idea behind ensemble methods is that a combination of several models, also called *base detectors*, and their outputs is more robust than usage of a single model. Such robustness can be observed against the bias-variance tradeoff and also for tackling the issue of overfitting. Although the possibility to combine multiple base detectors is intuitive, the design of such approaches needs special attention regarding normalization of outputs. In REATO, we use the RSR reconstruction error of each autoencoders and then we normalise each base detector scores through the standard deviation of one unit. We then take the median value for each observation.

Text Representation

In our REATO approach, we use RoBERTA (Y. Liu, Ott, et al., 2019) for text representation instead of GloVe, FastText or TFIDF. Ruff, Zemlyanskiy, et al. (2019) and Manolache et al. (2021) have recorded their results on these language model, in addition of BERT, but with meticulous observation of the results of Section 3.6 RoBERTA is a top performing representation. The REATO model is not based on the self-attention mechanism, such as for Ruff, Zemlyanskiy, et al. (2019) and Manolache et al. (2021), and we propose to use the implementation of Reimers and Gurevych (2019).

4.2.3 Experiments

Setup

We reproduce the exact same experimental setup as in Section 3.6. We use GenTO (see Section 3.5.2) for preparing contextual contamination on each candidate inliers possible with $\nu = 0.1$ and a split size of 350. All results are performed on AUROC and

Contextual										
Distill RoBERTA										
	Newsgroups		**Reuters**		**WOS**		**DBpedia 14**		**avg.**	
Model	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
KNN	0.310	0.778	0.492	0.795	0.620	0.900	0.762	0.948	0.546	0.856
OCSVM	0.282	0.750	0.491	0.811	0.599	0.889	0.759	0.945	0.533	0.849
PCC	0.314	0.776	0.518	0.828	0.613	0.897	0.771	**0.954**	0.554	0.864
OC-AE	0.191	0.623	0.246	0.604	0.249	0.680	0.348	0.735	0.259	0.660
RSRAE	0.309	0.779	0.506	0.821	0.621	0.900	0.762	0.936	0.550	0.859
RCEAE	0.194	0.623	0.278	0.615	0.448	0.810	0.368	0.747	0.322	0.698
REATO	**0.362**	**0.793**	**0.538**	**0.880**	**0.687**	**0.921**	**0.840**	0.951	**0.606**	**0.886**

TABLE 4.1: Results of state of the art models for contextual outliers with contamination rate $\nu = 0.10$. Average precision (AUPRC) and Area under ROC (AUROC) are evaluation metric.

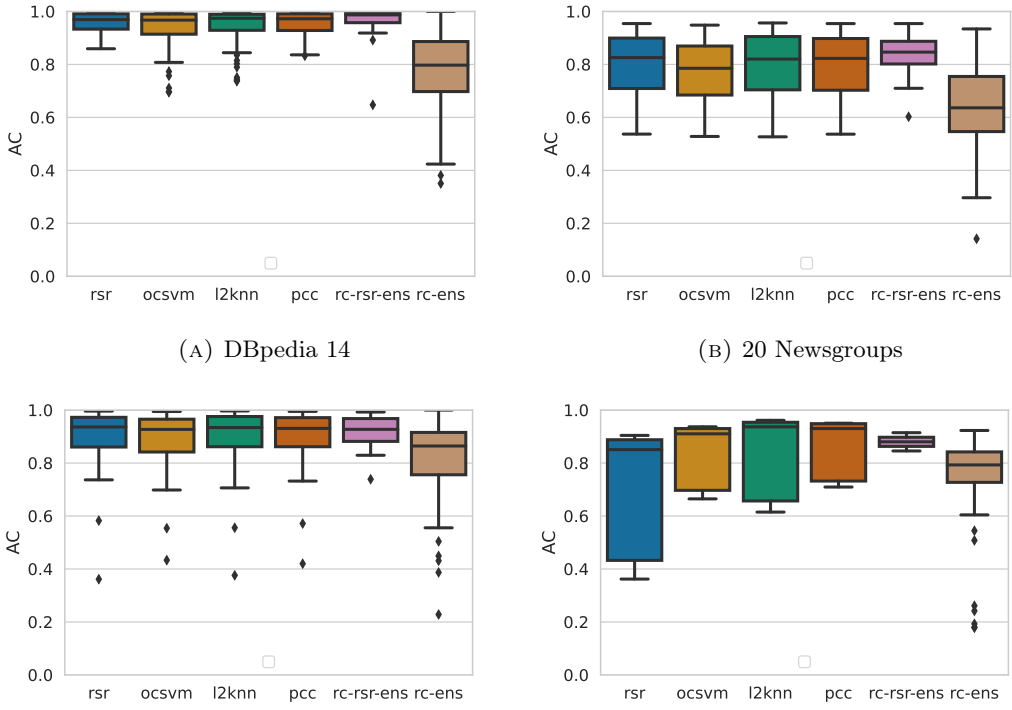
AUPRC reference works from the previous Section. We integrate results of one-class autoencoder and we also benchmark results on a randomly connected autoencoder ensemble (RCEAE) (J. Chen, Sathe, et al., 2017). The architecture is similar to J. Chen, Sathe, et al. (2017) and the autoencoders are following the settings of Section 3.6. The same goes for our approach REATO that follows the setup of Lai et al. (2020). We also keep the number of runs for each corpus to 10.

For REATO and RCEAE we setup similarly than with the autoencoder and we setup the number of base predictors to 25. Additionally, we also set hyperparameters $\lambda_1 = 0.1$, $\lambda_2 = 0.1$ and $\lambda_3 = 0.05$. For avoiding manifold collapse problem and degenerates solutions, we advise that $\lambda_3 < \lambda_1$. On the other hand, we set the epoch number to 30 and random connection probability between $[0.2, 0.5]$.

Results

Table 4.1 displays the experimental results conducted with our approach REATO. We observe that our approach is outperforming others model with AUROC metric and AUPRC metric. We can see that usage of REATO allow to mitigate unstable decision of the original RSRAE. We can also see significant difference of performance with Web of Science corpus and Reuters-21578. PCC is the only approach that succeeds to beat our approach against AUROC metric of DBpedia 14. Additionally, we can observe that the original one-class autoencoder highly benefit from randomly connection and ensemble technique, as it close the gap with other models.

While our performances are competitive, the principal purpose of tackling outlier detection with ensemble methods is to mitigate the bias-variance tradeoff. We propose to compare the model results with boxplots, similarly to the previous chapter. The main objective of our contribution is to robust outlier scores for contextual outliers with text. The Figure 4.3 and the Figure 4.4 displays an outperforming results from our approach (rc-rsr-ens). We can see that the variance of our model is noticeable as the box variance are always smaller than its competitors. Also, the min and max possible scores are close from the median scores, concluding to see that our approach is more efficient, more robust and can handle well language model like RoBERTA.



(A) DBpedia 14

(B) 20 Newsgroups

(C) Web of Science

(D) Reuters-21578

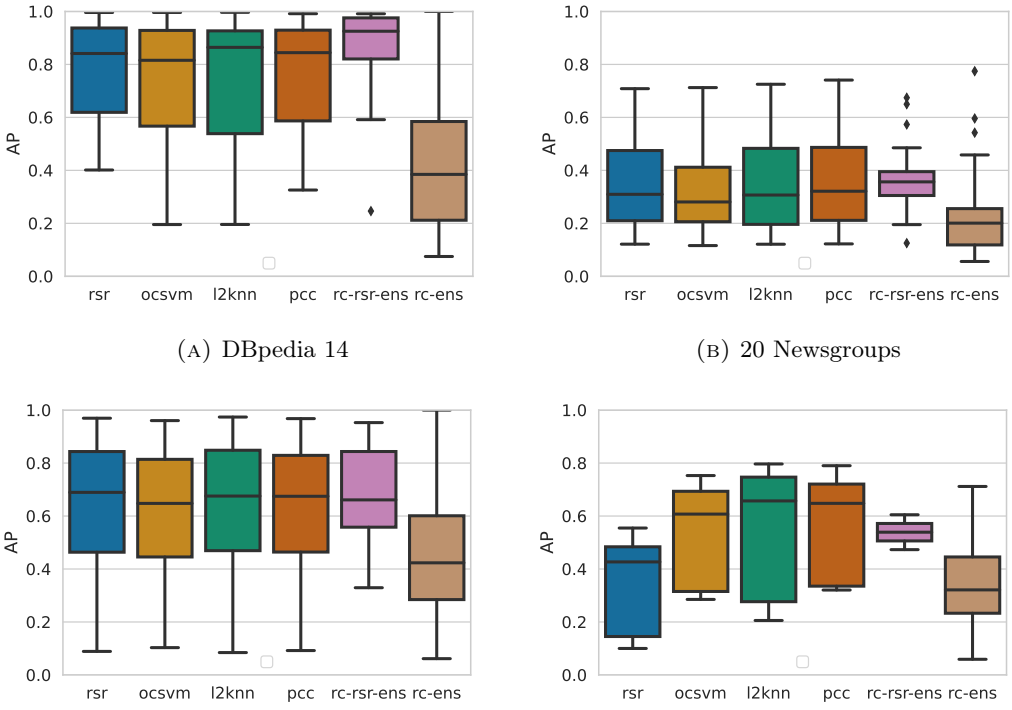
FIGURE 4.3: Results of our experimental study with $\nu = 0.1$, split size of 350 and number of base detector of 25. The performance metric is AUROC (AC) and the text representation is RoBERTA.

4.2.4 Discussion

In this section we have introduced REATO, an ensemble approach with RSR a autoencoders, otpimzed through LLE for tackling contextual outlier in text. Further work are planned to be conduct regarding the sensitivity of λ_3 hyperparameter as well as other hyperparameters. Another perspective is to study the integration of attention head for mitigating the black box problem of our model. It is common recently to display text with their corresponding temperature, thanks to recent language model based on transformers. The representation of text is a key concept that we want to investigate in the near future.

4.3 Adding polarity features for outlier detection

In this section, we propose an experimental approach to enrich texts for outlier detection in texts. This approach is based on a dedicated representation of text for LOF and density-based methods. TF-IDF is not efficient to address all characteristics of textual data such as semantic, synonyms, syntax and many others. In our approach, we perform a low-rank approximation using Singular Value Decomposition (SVD) on TF-IDF matrix. Each dimension t of the matrix is an explainable topic with a combination of terms. A document is then transformed into a low k-dimensional vector



(A) DBpedia 14

(B) 20 Newsgroups

(C) Web of Science

(D) Reuters-21578

FIGURE 4.4: Boxplots of results of our experimental study with $\nu = 0.1$, split size of 350 and number of base detector of 25. The performance metric is AUPRC (AP) and the text representation is RoBERTA.

of size k. Different ranks have been studied for LSA. The optimal dimension number is found empirically through experiments presented in Section 4.3.1.

It is clear that words carry other information than semantic or frequency. For instance, polarity is such kind of information and we propose to use it in addition to LSA. As presented in Section 4.1.5, VADER (Hutto and Gilbert, 2014) is a simple and efficient approach that performs sentiment analysis without supervision. It is an approach based on lexicon and rule-based sentiment analysis that is specifically designed to detect sentiments expressed in social media. This approach outputs four attributes: *negative*, *neutral*, *positive*, and *compound*. We build a sentiment vector with the *negative* and *positive* attributes. This sentiment vector is appended to the low k-dimensional vector to form a representation vector of the text.

Let C be a TF-IDF matrix $m \times n$, representing m documents with n features (i.e. terms), and d_i a document of C where $i \in [1, m]$. In the new feature space, d_i becomes:

$$d'_{i,k} = (t_1, t_2, ..., t_k, s_1, s_2) \tag{4.6}$$

$(t_1, t_2, ..., t_k)$ is the corresponding LSA transformation of C and (s_1, s_2) the sentiment vector of d_i. The dimension of the new feature space is thus $k' = k + 2$.

This representation of text is performed to help LOF to detect outliers. The main idea of LOF is to process local density based on nearest neighbors with a distance

metric. In our approach, we use the Manhattan distance (C. C. Aggarwal, Hinneburg, et al., 2001) because we have found it performs the best against other distance metric in our empirical test. Distance metrics benefit from low dimensional spaces and expressive attributes. Polarity can handle abnormal documents that are highly positive or negative. Depending of context, special words can be used and bring with them weak signals. With the addition of sentiment attributes, other outliers than usual ones can thus be detected.

4.3.1 Experiments

This section presents results of the conducted experiments and studies each approach in respect to three parameters: contamination rate, rank and outlier detection algorithms. First, the experimental setup is introduced, then we present our experiments on early fusion with news articles (20 Newsgroups and Reuters-21578). We display the conducted experiments on early fusion with IMDB and finally, we introduce the results on late fusion (outlier ensemble) with our new representation. All experiments have been conducted on an Intel Core i7-4770 processor with 16 GB of RAM.

Experimental Setup

Data Experiments have been conducted on three popular datasets: the 20 Newsgroups dataset, the Reuters-21578 dataset and the IMDB Movie Reviews dataset. 20 Newsgroups and Reuters-21578 are state-of-the-art reference datasets for outlier detection on text. They are originally designed for textual classification tasks on News articles. Also, news articles tend to be neutral. The IMDB Movie Reviews dataset is used to experiment our proposed approach on subjective and emotional texts. In this work, we introduce PoLSA, a novel text representation that performs early fusion of polarity features with semantic features. Our contribution aims to propose a dedicated representation for tackling polarity-based outliers and semantic outliers at the same time.

The experimental protocol is the one proposed in (Lai et al., 2020) for 20 Newsgroups and Reuters-21578. For all datasets we keep the original train/test split and for each experiment, inliers are documents from fixed class and outlier are sampled from other classes. We contaminate inliers with outlier with different *contamination rates* (see Section 3.6). For instance, a class with 300 documents leads to a subset containing 270 inliers and 30 outliers when prepared with a contamination rate of 0.1. The positive class represents the outliers. The experiments have been performed with the following contamination rates: 0.01, 0.05, 0.1 and 0.15 to cover different conditions of difficulty to detect outliers.

Representation of texts Four representations are compared: TF-IDF, LSA-k (with different ranks k), sentiment and our PoLSA. In a preprocessing step, we lowercase raw text and filter stopwords before removing punctuation and special characters. For the 20 Newsgroups, we keep the body message and filter empty documents. We

apply LSA on the TF-IDF representation with $k \in \{30, 50, 100, 200, 300, 500\}$ and show the best results. For representation with sentiments, VADER is applied on raw documents using NLTK (Bird et al., 2009). Finally, we concatenate LSA-k with sentiment features to build our proposed representation of texts.

Outlier detection PoLSA is evaluated with different popular outlier detection approaches that have been proved to be efficient in the literature and often found in state of the art baselines: Local Outlier Factor (LOF) (Breunig et al., 2000), One-Class SVM (OCSVM) (Schölkopf, Platt, et al., 2001) and Isolation Forest (IF) (F. T. Liu et al., 2008). Even if these methods are not recent, they have shown good results on high-dimensional data.

LOF highly benefits from dimensionality reduction but isolation forest is also a candidate to improvements. Unlike LOF and isolation forest, OCSVM records great results on outlier detection for text. LOF is an interesting approach to study with its density characteristic. Compared to LOF, Isolation Forest predictions are hard to explain and interpretability of this model can be hard to do. Black box methods such as OCSVM are the hardest methods to interpret. Our proposed approach aims to help density-based approach like LOF to get comparable results while getting better interpretable properties. To obtain benchmarks on those methods, we adapt our code with PyOD (Zhao, Nasrullah, and Li, 2019) package for getting outlier scores.

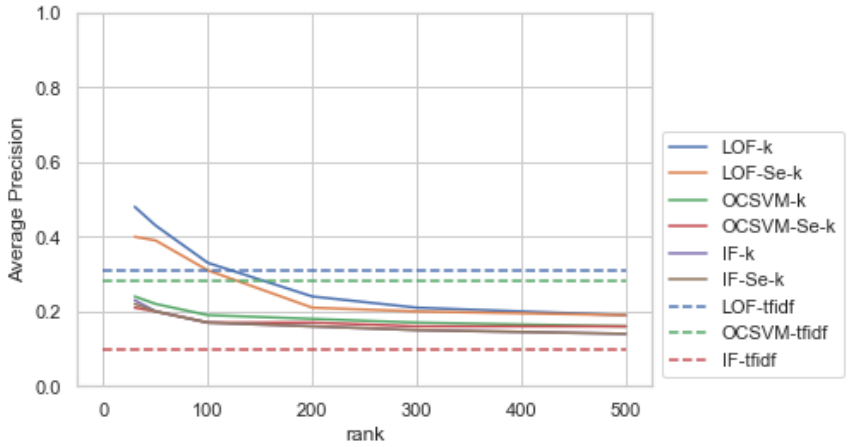
Evaluation Metrics Outlier detection is a task with high imbalanced data where inliers (true negatives) are predominant. As a consequence, average precision is often used to get a good idea of performance. The different representations are compared by means of the Area Under the Receiver Operating Characteristics curve (AUROC) and the Average Precision. These classical metrics are derived from the confusion matrix and are both often used for the outlier detection task. The ROC curve displays True Positive Rate (TPR) on False Positive Rate (FPR) for many thresholds. Increasing or decreasing this threshold influences true positives with respect to false positives. It helps to choose the best threshold for the classifier.

The AUROC can be then considered as an accuracy metric. For both metrics, the corresponding implementation from Scikit-Learn (Pedregosa et al., 2011) is applied on each conducted experiment. All experiments have been conducted on five runs where Average Precision and AUROC are averaged for all contamination rates and all datasets.

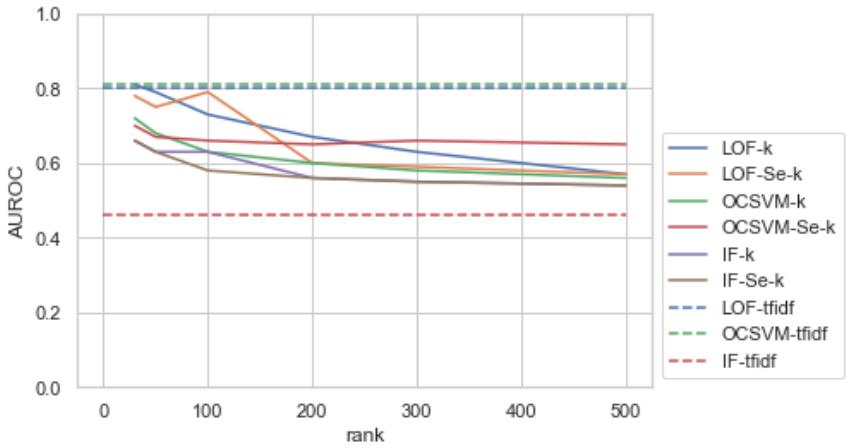
4.3.2 Early fusion: News corpora

Setup

We prepare 20 Newsgroups and Reuters-21578 similarly to (Kannan et al., 2017) and (Lai et al., 2020). The difference is that we keep the original size of each class, as opposed to the Section 3.6 where we have used a size of 350. We use GenTO (Section 3.5.2) for preparing all of the split with the independent contamination. For



(A) 20 Newsgroups AUPRC



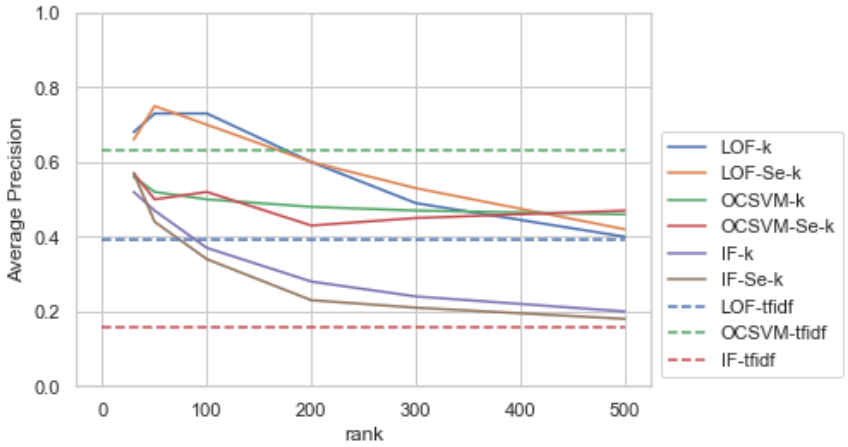
(B) 20 Newsgroups AUROC

FIGURE 4.5: Average precision (AUPRC) and AUROC for different ranks for 20 Newsgroups with *contamination* = 0.10. k is the corresponding rank used for dimensionality reduction with LSA and *Se* means that the sentiment vector is used.

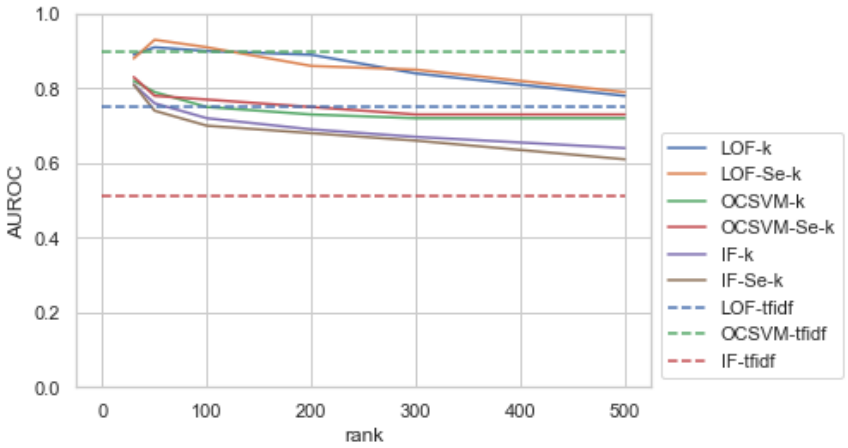
Reuters-21578, we remove all documents that are originally associated with more than one class. We keep the largest five classes *acq*, *crude*, *earn*, *money-fx* and *trade* as candidates to be inliers. For TF-IDF representation, we setup LOF with *Cosine Distance* and the number of nearest neighbors is experimentally set to 20. On other representations we use the *Manhattan Distance*. For OCSVM we choose the *Radial Basis Function* (RBF) with $\gamma = \frac{1}{n\_features}$. The number of isolation trees in Isolation Forest is 100. Further studies will be focused on the influence of these hyper-parameters for the results.

Results

We first present the results of each method on Figure 4.5 and Figure 4.6. These figures show AP and AUROC with different ranks. Reducing the dimensionality of TF-IDF with LSA greatly help LOF and IF to get better performances. The results



(A) Reuters-21578 AUPRC



(B) Reuters-21578 AUROC

FIGURE 4.6: Average precision (AUPRC) and AUROC for different ranks for Reuters-21578 with *contamination* = 0.10. k is the corresponding rank used for dimensionality reduction with LSA and *Se* means that the sentiment vector is used.

indicate that LOF clearly outperforms other methods in most scenarios with rank 30 and 50. For Reuters-21578, LOF with our representation and $k = 50$ outperforms other approaches and has similar results as LSA. When $k > 100$, OCSVM with TF-IDF outperforms all approaches. For 20 Newsgroups, LOF with our representation and $k <= 100$ gets similar performance as with LSA on AP. For $k = 30$, AUROC of LOF with our representation get similar results than OCSVM and LOF with TF-IDF. Table 4.2 shows AP and AUROC for several contamination rates. The results for LOF with LSA for $k \in \{30, 50\}$ (LOF-30 and LOF-50) and results of IF and OCSVM with $k = 30$ (OCSVM-30 and IForest-30). We then show results of LOF, OCSVM and IF with sentiment vector (LOF-Se, OCSVM-Se and IForest-Se) that is built with both *negative* and *positive* attributes.

Finally we present results on proposed approach with sentiment representation and LSA. At rank 30 and 50, LOF with LSA outperforms other baselines and is the

Models	Reuters-21578								20 Newsgroups							
	Average Precision				*AUROC*				*Average Precision*				*AUROC*			
	0.01	*0.05*	*0.10*	*0.15*	*0.01*	*0.05*	*0.10*	*0.15*	*0.01*	*0.05*	*0.10*	*0.15*	*0.01*	*0.05*	*0.10*	*0.15*
tfidf																
LOF	0.07	0.37	0.39	0.42	0.46	0.76	0.75	0.70	0.11	0.22	0.31	0.36	**0.85**	**0.82**	0.80	0.78
OCSVM	**0.62**	0.66	0.63	0.66	**0.95**	0.92	0.89	0.86	0.07	0.17	0.28	0.36	0.82	0.80	0.80	0.79
IF	0.03	0.09	0.16	0.20	0.44	0.50	0.51	0.49	0.02	0.06	0.10	0.15	0.44	0.47	0.46	0.47
lsa																
LOF-30	0.57	0.70	0.68	0.66	0.93	0.93	0.89	0.88	**0.21**	**0.37**	**0.48**	**0.54**	0.79	0.81	**0.81**	**0.80**
LOF-50	0.59	0.70	0.73	0.72	**0.95**	**0.94**	0.91	0.89	**0.21**	0.35	0.43	0.50	0.80	0.80	0.77	0.76
OCSVM-30	0.49	0.56	0.60	0.62	0.84	0.86	0.82	0.82	0.08	0.15	0.24	0.31	0.75	0.72	0.72	0.70
IF-30	0.50	0.51	0.59	0.57	0.83	0.80	0.81	0.78	0.09	0.14	0.23	0.29	0.69	0.67	0.66	0.64
sentiment																
LOF-Se	0.06	0.13	0.15	0.20	0.50	0.58	0.55	0.52	0.01	0.06	0.11	0.16	0.53	0.53	0.51	0.52
OCSVM-Se	0.04	0.10	0.16	0.23	0.61	0.59	0.59	0.60	0.03	0.12	0.14	0.19	0.52	0.53	0.52	0.52
IForest-Se	0.03	0.12	0.17	0.21	0.55	0.61	0.60	0.59	0.02	0.08	0.13	0.18	0.55	0.57	0.55	0.55
PoLSA																
LOF-Se-30	0.54	0.67	0.66	0.67	0.92	0.92	0.88	0.86	0.18	0.31	0.40	0.47	0.77	0.78	0.78	0.76
LOF-Se-50	0.57	**0.71**	**0.75**	**0.73**	0.92	0.92	**0.93**	**0.90**	**0.21**	0.30	0.39	0.45	0.76	0.76	0.75	0.75
OCSVM-Se-30	0.44	0.55	0.57	0.57	0.86	0.86	0.83	0.79	0.06	0.13	0.21	0.27	0.69	0.70	0.70	0.68
IForest-Se-30	0.35	0.48	0.57	0.59	0.80	0.83	0.81	0.78	0.06	0.15	0.22	0.29	0.68	0.67	0.66	0.65

TABLE 4.2: Average precision and AUROC for Reuters-21578 and 20 Newsgroups with different representations of text and for several contamination rates ($contamination \in \{0.01, 0.05, 0.10, 0.15\}$).

most robust approach against contamination rate. Our approach with LOF is robust to low contamination rate and succeeds to challenge other methods in many cases. We observe that the sentiment representation gets poor results for all cases but results on Reuters-21578 tends to imply that the corpus admits more polarized discourse than 20 Newsgroups.

In Section 3.7 we have supposed that better text representations than TFIDF may considerably increase results. We observe that overall approaches benefit from the LSA topics and even isolation forest that was the most criticized approach is now getting good results. We note that in the aforementioned experiments, we had set the split size to 350, leading to all dataset being evaluated similarly. We observe that increasing the size can get completely different results, in particular the AUPRC which is relative to the distribution size and contamination. With our fused representation, LOF is now outperforming the best approaches on Reuters-21578 recorded in the previous chapter. LSA also increases results on 20 Newsgroups (without sentiment representation) and outperforms the previous methods also.

Discussion

The results show that LSA is a good technique to help density-based approaches to detect outliers on text. Performances show that LSA is better than TF-IDF for outlier detection and it also gets better robustness against low contamination rates. One of the problem is related to terms that are out of vocabulary on test split. It particularly affects the results of OCSVM and LOF on 20 Newsgroups leading them to get lower results than on Reuters-21578. We have proposed to use VADER for our approach but the method is not only dedicated to news articles and can be exchanged with a specific one. However, we observe that our representation detects outliers with sensitive topics that are not recognised by other representations. We also observe that polarity has a positive impact on LOF. While keeping topic level detection, our

Models	IMDB Movie Reviews			
	AUPRC		***AUROC***	
	0.05	*0.10*	*0.05*	*0.10*
tfidf				
LOF	0.08	0.11	0.56	0.53
OCSVM	0.06	0.10	0.49	0.50
IF	0.06	0.10	0.51	0.52
lsa				
LOF-50	0.09	0.13	0.64	0.58
LOF-100	0.10	0.13	0.64	0.57
OCSVM-50	0.06	0.10	0.56	0.52
OCSVM-100	0.06	0.09	0.57	0.51
IF-50	0.06	0.10	0.56	0.51
IF-100	0.06	0.10	0.55	0.49
sentiment				
VADER	0.14	0.18	0.73	**0.67**
LOF-Se	0.08	0.14	0.55	0.53
OCSVM-Se	**0.21**	0.23	0.66	0.61
IF-Se	0.14	0.19	**0.76**	**0.67**
PoLSA				
LOF-Se-50	0.13	0.17	0.59	0.62
LOF-Se-100	0.11	**0.24**	0.58	**0.67**
OCSVM-Se-50	0.20	**0.24**	0.66	0.62
OCSVM-Se-100	0.18	0.23	0.64	0.63
IF-Se-50	0.07	0.13	0.59	0.60
IF-Se-100	0.07	0.14	0.59	0.60

TABLE 4.3: Results on the IMDB Movie Reviews dataset with different representations of text and for different contamination rates (0.05 and 0.10).

representation succeed to detect additional outliers than other representations. We also note that another gain related to dimensionality reduction is that training and test time take significantly less time than with TF-IDF. The table 4.4 shows the execution times that were recorded for all methods and displays a comparison for each data set and representation. We observe that our representation significantly reduces the execution time compared to TF-IDF while obtaining similar records with LSA. Our representation benefits from dimensionality reduction and shows better run times for all cases. We note that VADER inference makes our representation relatively slower compared to LSA alone.

4.3.3 Early fusion: Movie reviews dataset

Setup

The IMDB Movie Reviews is a dataset with 50000 documents that is commonly used for sentiment analysis task. Each document is labelled as *negative* or *positive*. For both classes, we randomly sample between 2000 and 3000 documents and outliers

	Run Time (Seconds)		
	TF-IDF	**LSA-30**	**LSA-Se-30**
20 Newsgroups			
LOF	165.44	30.21	37.18
OCSVM	514.16	30.03	36.57
IF	74.25	31.46	39.31
Reuters-21578			
LOF	174.31	25.02	27.04
OCSVM	847.58	24.59	26.52
IF	60.26	25.01	27.17
IMDB Movie Reviews			
LOF	216.08	14.03	15.30
OCSVM	627.57	14.05	15.26
IF	77.31	14.11	15.31

TABLE 4.4: Comparison of execution times with TF-IDF, LSA-30 and LSA-Se-30 for all datasets (time includes the training time of the corresponding representation).

are documents from the other class. The setting of baselines is the same as for 20 Newsgroups and Reuters-21578. We add VADER results based on the output of the approach.

Results

Table 4.3 presents the results for the IMDB Movie Reviews dataset. For contamination rate 0.1 (a medium rate chosen for this first study), LOF with our proposed representation outperforms other methods on average precision and AUROC. OCSVM with sentiment features with our representation performs the best on low contamination rate. We observe that our approach has similar results as dedicated methods for sentiment analysis. While LSA and TF-IDF perform poorly on this dataset, our approach improves LOF and OCSVM results.

Once again, if we compare the results with those of the Section 3.6, we observe a great improvement. Indeed, the comparison of the results on IMDB between the Table 3.5 and the Table 4.2 displays an AUPRC doubled and an AUROC greatly higher. With the results on the News dataset, we can see that early fusion is working as intended and does not penalized results of one or another modality. We observe a clear drawback on the use of isolation forest of our representation instead of the VADER one.

Discussion

The results on the IMDB Movie Reviews show that our approach performs well on documents with polarity. We clearly observe that documents of this dataset can not be fully discriminated using topics but our approach succeed to get better performance with contamination rate 0.1. Our observation is that LOF use the semantic representation (LSA) to place documents in its space and sentiment vector to discriminate

Models	Independent		Contextual		Collective	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
LSA 50 components						
LOF	0.315	0.832	0.180	0.729	0.239	0.754
Autoencoder						
Average	0.464	0.892	0.362	0.832	0.383	0.898
Maximisation	0.488	0.839	0.342	0.820	0.358	0.885
AOM	0.448	0.890	0.351	0.828	0.363	0.891
MOA	0.457	0.889	0.360	0.829	0.382	0.897
LSA						
Average	**0.517**	**0.959**	**0.427**	**0.883**	**0.494**	0.889
Maximisation	0.475	0.953	0.420	0.872	0.487	0.884
AOM	0.477	0.950	0.426	0.876	0.487	0.884
MOA	0.497	0.957	0.426	0.882	0.496	0.891
PCA						
Average	0.452	0.935	0.390	0.870	0.397	**0.912**
Maximisation	0.430	0.922	0.382	0.864	0.364	0.895
AOM	0.432	0.922	0.388	0.866	0.380	0.898
MOA	0.446	0.931	0.392	0.867	0.387	0.898

TABLE 4.5: Results of late fusion methods on three dimensionality reduction techniques on Reuters-21578 dataset. Average precision (AUPRC) and Area under ROC (AUROC) are evaluation metric.

them. Table 4.4 shows that our representation is faster than TF-IDF but slower than LSA alone. In future work, emotional features such as *fear*, *joy* and *anger* will be studied deeper to highlight their influence.

4.3.4 Late fusion: News corpus

Setup

We prepare Reuters-21578 with GenTO (Section 3.5.2) for independent, contextual and collective outliers (see Section 3.3.4). The collective preparation dataset is done with half the contamination from one independent outlier class and half from another contextual outlier class. Thus, collective outliers are created with one independent cluster and one contextual cluster. We also note that the contamination rate $\nu = 0.10$. For the experiment, we use our fused representation with a LSA of $k = 50$ components. Also, for each model of the baseline we train the same model ten times with different values for hyperparameters. We average the results on five runs (five different prepared splits).

Our baseline is formed with an one-class autoencoder, a latent semantic analysis and a PCC. For LSA and PCC we train models with different values of $k \in \{10, 15, 20, 25, 30, 35, 40, 45, 50, 55\}$. Regarding the autoencoder, we setup five kinds of architecture with different hidden layers settings and for each one of them we set the dropout rate to 0.2 and 0.4. After that, we aggregate the output of all predictors with the introduced methods of Section 2.5.7.

Results

The results displayed in Table 4.5 record great performance for all the baseline. If we compare those results against the previous one, we can see that ensemble methods are more efficient and more stable. Indeed, we do not observe a considerable drop of scores from AUPRC and AUROC, even on contextual and collective outliers. Collective outliers are also harder to work with than independent, but still easier than contextual. The *average* process is outperforming the others but its scores still are near. These results demonstrate that the late fusion can also perform better results with the same models.

Discussion

The addition of early fusion and late fusion are considerably improving the results for outlier detection on text data. With picking the right representation and the right models, we have seen that the outcomes are more stable, robust and efficient. The results of this section also demonstrate that outlier detection is the problem of knowing the distribution. Indeed, all this setting with early and late fusion is possible thanks to the supervised evaluation. In the case with no label in hand, the situation is different harder to optimize. One solution may be to extract the most significant features (statistically for instance) and estimate few results on which a human can be confident to understand. Another way may be to use the characteristic of the corpus and its document for estimating an expectation of what is structurally "normal".

4.3.5 Conclusion

In this section we have conducted an experimental study of early fusion and late fusion techniques, adapted to outlier detection with text data. We have introduced PoLSA, a text representation that integrates a multimodal vector with polarity features and latent features. Through such aggregation, PoLSA is an efficient and fast representation of text that can tackle different kind of corpora without decreasing result from original representation. If we compare results from Table 3.5 with Table 4.3, we observe a significant performance gap between RoBERTA representation and PoLSA_{50} for IMDB corpus. The benchmark is performed on traditional approaches like local outlier factor and one-class support vector machine, promising future works consists to use recent methods and REATO.

Late fusion (or outlier ensemble) have been explored in Section 4.2 with the introduction of REATO. In this section we explore furthermore traditional technique for performing late fusion in our context. We applied those techniques on three dimensionality reduction approaches and we observe in Table 4.5 that outlier ensemble technique are promising for performing outlier detection in text data. They find success with independent, contextual and collective outliers.

While this section introduce a wide number of viable scenarios for performing outlier ensemble and fusion at several level, we can conclude that they are promising methods for future research. The very recent advances in language model can be a

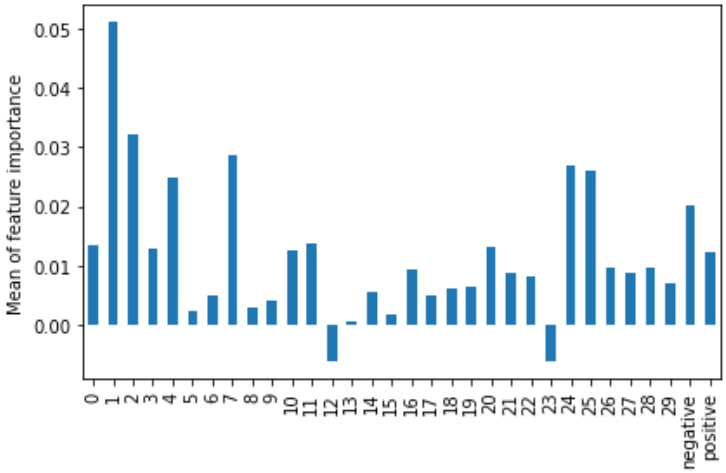


FIGURE 4.7: Feature importances using permutation with LOF and $k = 30$. Evaluation metric is average precision and the class is *baseball*.

beneficial addition to our study and we aim to investigate reduction techniques in this context. State-of-the-art works Ruff, Zemlyanskiy, et al. (2019) and Manolache et al. (2021) have used attention head based techniques for performing anomaly detection in text, such approach can be applied in this context.

4.4 Interpretability

Methods applied on textual data are often hard to interpret due to high dimensionality and hidden semantic. TF-IDF partially tackle semantic issue but is a representation of text that is explainable. Compared to recent works on language model (Devlin et al., 2019), dimensionality reduction based on a term matrix are more interpretable. Interestingly, BERT-based language models can be analyzed and studied through their attention head that indicate important learned features. This characteristic has allowed numerous reference works to succeeds in explaining several behaviors from such kind of representation (Clark, Khandelwal, et al., 2019; Vig and Belinkov, 2019; Jain and Wallace, 2019). Our proposition to use Latent Semantic Analysis for outlier detection also aims to study results.

LSA maps terms and documents under topics that can be retrieved with documents based on terms. While performing SVD, LSA associate patterns between terms that are unique. This property reduces noise and estimates textual information, assuming that a term has nearly one meaning. When density-based methods predict with LSA, we can retrieve terms that are associated to each topics. Table 4.4 illustrates how topics can be retrieved for Reuters-21578 with our approach. Similarly to research papers that aim to explain high-dimensional models Ribeiro et al. (2016), Kim et al. (2016), and Lundberg and S.-I. Lee (2017), we discovered that LSA succeeds to find few topics that are positives or negatives.

Outlier detection in textual data can be difficult to define. In this work we have shown that outlier data can be formalised depending of the application (sentiment

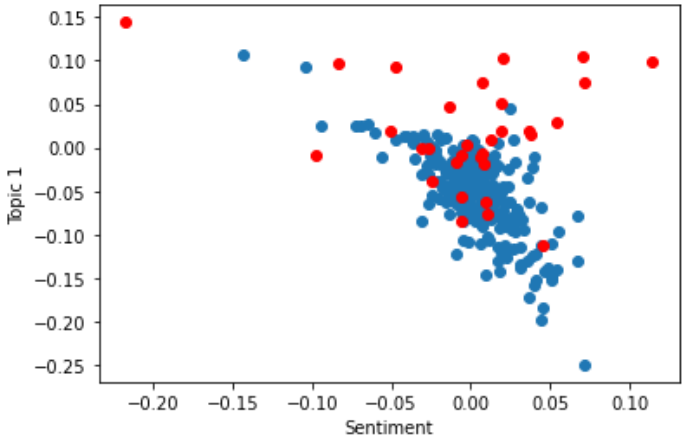


FIGURE 4.8: Representation of documents in 2-D dimensions with a projection of sentiment on Topic 1 vector. Blue points are inliers and red points are ground truth outliers. The contamination rate is set to 0.10.

analysis and text classification). For News corpus we observe that semantic level features are the most efficient direction to find outliers. On the other hand, finding outliers in reviews or social media data needs a different approach. Features from opinion mining such as polarity are great to handle characteristics of this type of documents. In this section we explore documents representation with the help of feature permutation importance.

Model explanation is a blossoming field and there are several approaches that have emerged (Ribeiro et al., 2016; Lundberg and S.-I. Lee, 2017). In addition to the emergence of outlier detection and anomaly detection studies, we can find overview such as Yepmo et al. (2022). In such work, the problem of explaining an anomaly is addressed in the context of eXplainable Artificial Intelligence (XAI). Numerous works have been conducted in this context (Dang et al., 2014; Tang et al., 2013; N. Liu et al., 2017; Macha and Akoglu, 2018; Gupta et al., 2019; Siddiqui et al., 2019; H. Xu et al., 2021; Smits et al., 2022), and outlier analysis can successfully help to explain model decision.

Recently, fairness and bias in outlier detection have also been subject of concerns (Davidson and S. S. Ravi, 2020; Shekhar et al., 2021). In machine learning, an algorithm trains a model that produces predictions and, in that regard, interpretability can be global or local. Depending on the type of interpretability, different techniques can be applied to explain a prediction for an instance or a group of instances.

One of the interesting properties of LOF is that its prediction can be studied with its nearest neighbors. While the prediction of LOF can be explained with low-dimensional data, it is impossible with high-dimensional data such as texts with terms matrices. LSA allows us to perform low-rank approximation and then reduce dimension of text. Each dimension is a topic, represented by a group of terms, which facilitates the interpretation of the text. With LSA, one or several topics can be associated to a document.

Terms	Topic 1	Topic 2	Topic 3
term1	bank	increase	commission
term2	florida	barrel	minister
term3	operations	intermediate	reduce
term4	company	price	electronic
term5	billions	markets	equipment
Polarity	neutral	neutral	negative

TABLE 4.6: Top 5 terms for three topics of Reuters-21578 with our representation ($k = 30$).

In our work, results on 20 Newsgroups demonstrate that our representation is less successful on this dataset than on Reuters-21578. While AUROC has similar performance with our representation and LSA for $k = 30$, it gets lower results on AP. We propose to study our representation with feature selection. Figure 4.7 shows feature importance for class *baseball* of 20 Newsgroups, performed with permutation features. Topic 1 is the most important feature for this class. We observe in Figure 4.7 that sentiment vector positively contributes. We illustrate the impact of the polarity vector by combining negative and positive in one feature where a negative score means negative and a positive score means positive.

Figure 4.8 displays documents of *baseball* class from 20 Newsgroups with selected features. We observe that half of the outliers are correctly isolated from inliers. Evaluation of AP for this class with our representation is originally 0.49. With the selected features, AP increases to 0.52. Based on LSA, we can retrieve terms that compose topics and define which words contribute in the outliers detection. We can observe that few inliers are isolated according to their polarity. In a different context, where an outlier is defined as "neutral news", these isolated documents would likely be outliers. Exploration of interpretability on local instance will be conducted in further works as well as rule-based approaches.

4.5 Conclusion

In this chapter we have explored outlier ensemble methods and applied it to text. We introduced the different challenges and motivations that led us to conduct this work. Based on this context, we introduced a novel ensemble outlier approach for text data: REATO. The experimental study reveals that our approach REATO outperform state-of-the-art methods and succeeds to get a robust outlier score with contextual contamination. Results also demonstrate that REATO is less sensitive to bias and variance.

We have introduced PoLSA, a new representation of text built from LSA and sentiment attributes. This approach succeeds to detect outliers on neutral and emotional documents. Our study of low rank approximation with LSA confirms that density-based approaches can get state of the art results. In addition to reducing the dimensionality of a TF-IDF matrix, it associates documents with a topic. Through

PoLSA, our representation can facilitate the interpretability of predictions compared to a high-dimensional representation such as TF-IDF.

There are several directions for future work on dedicated features like emotion tagging and syntax analysis. Moreover, a deep study on how to estimate contamination rate will be conducted. When contamination is set as hyper-parameter, methods are forced to label inliers into outliers in order to fulfil the ratio. Similarly, LOF results can be improved with study of hyper-parameter such as nearest neighbours.

Chapter 5

Improved Abstractive Summarization Through Outlier Analysis

In the preceding chapters, we delved into outlier analysis and explored its significance and the various approaches when dealing with text data. We addressed the challenges of detecting outliers in textual content and proposed novel methods for identifying both point and contextual outliers effectively. Additionally, we examined the potential of outlier ensembles and the interpretability of outlier detection models.

Based on Chapter 4, we now shift our focus to an intriguing application of outlier analysis in the context of abstractive summarization. Abstractive summarization involves generating concise and coherent summaries that capture the essential information from a given piece of text. This approach has shown remarkable promise in distilling large volumes of information into succinct summaries, but it is not immune to challenges, particularly in handling complex or diverse texts.

In this chapter, we investigate how outlier analysis can significantly improve the performance and robustness of abstractive summarization models. By leveraging the outcomes of Chapter 3 and Chapter 4, where we developed advanced outlier detection techniques tailored for text data, we explore novel works to enhance the summarization process. Our primary focus is on the utilization of outlier detection to bolster the quality, interpretability, and reliability of abstractive summaries. Furthermore, we also investigate the problems that can be associated to the evaluation process.

At first, we present the task of abstractive summarization with neural networks in Section 5.1 and general knowledge associated with. We then introduce the different challenges to tackle for performing unsupervised text summarization in Section 5.2. By harnessing neural networks and unsupervised learning, we lay the groundwork for our improved abstractive summarization model. Next, we present an in-depth study of evaluation methods of summarization task, comparing existing methods and assessing their performance under various scenarios. We present the different problems when performing automatic evaluation with abstractive summarization. An overview and concluding guideline are given in this context.

Finally, a significant part of this chapter revolves around robust abstractive summarization, where we investigate how outlier analysis can enhance the summarization

Text	Summary (abstractive)	Summary (extractive)
the srilankan government on wednesday announced the closure of government schools with immediate effect as a military campaign against tamil separatists escalated in the north of the country.	srilanka closes schools as war escalates	srilankan government announced closure of schools as a military campaign

TABLE 5.1: Illustration of the abstractive summarization task (Q. Zhou, N. Yang, Wei, and M. Zhou, 2017) as well as by extractive. The input text is found in the DUC 2004 dataset.

process in the face of complex, noisy, or out-of-distribution texts. We explore in Section 5.4 how outlier detection mechanisms can contribute to producing more reliable summaries, even in challenging and ambiguous contexts.

In conclusion, we summarize the contributions of this chapter and discuss the implications of our findings. We highlight the potential of outlier analysis with abstractive summarization and lay the foundation for future research in the field. Furthermore, we outline possible directions for extending this work and applying outlier analysis to other natural language processing tasks. By seamlessly connecting previous contributions with the domain of abstractive summarization, this chapter presents a novel perspective on improving the summarization process and offers valuable insights into the potential synergies between outlier detection and neural network-based text summarization.

5.1 Abstractive summarization with neural networks

The purpose of the summarization task is to generate a compressed version of a text based on the information of the original document. State of the art methods are similar to the one of machine translation task. Despite strong differences in the final result, some techniques that have been proposed in the field of machine translation can be used in automatic text summarization. Among the approaches that best perform we find artificial neural networks. Despite their success in this task, they are dependent on the quality and quantity of the data. We present in this section the principal architecture of neural networks, the attention mechanism and techniques that make models of the literature more robust.

5.1.1 A general pipeline

Abstractive summarization is different from extractive summarization, but both aim to generate a textual summary from an original document. The extraction method cut the most important parts of the text to assemble a summary. The abstraction method aims at interpreting the important information contained in the text in order to spell it differently (eventually). Table 5.1 gives an illustrative example of both of them.

The task is separated into three main steps. The first step represent the input data into a space so that an artificial neural network can ingest it. We use a word representation learning method called *word embedding* (Mikolov et al., 2013). Very often, this technique allows to represent each word of a dictionary by a vector of real numbers.

In the next step, the *sequence-to-sequence* (Sutskever, Vinyals, et al., 2014) architecture takes place. This approach moreover uses the architecture of *encoder-decoder* (K. Cho et al., 2014). This one consists in taking as input a sequence (of words) x, which is then encoded in a sequence vector z. The *decoder* takes the sequence vector z as input and produces a sequence y as output. This *decoder* is usually auto-regressive, which means that its outputs are fed back as inputs to the *decoder*. If the encoder takes the entire word sequence as input, the decoder can generate a word sequence in one go. Nevertheless if the encoder takes a single word as input, the decoder can also generate a single word.

The third and final step is performed by a heuristic search algorithm such as *beam search*. For choosing the best summary, each part of the proposed sequence is processed and different words are then submitted. *Beam search* is used to select the best proposal by considering the grammatical structure as well as several other criteria.

The use of a reference abstract poses a problem in supervised learning since it may be penalized by the absence of a sufficiently rich sample of abstracts. Unsupervised learning is implemented when there is no reference for each data. The latter therefore avoids the problem mentioned above.

5.1.2 Sequence-to-Sequence

The *sequence-to-sequence* (*seq2seq*) architecture consists in generating sequences with inputs that are also sequences. With the summarization task, sequences of words (*tokens*) are used. This task can first be handled by *encoders-decoders*, as proposed by (Sutskever, Vinyals, et al., 2014) in the machine translation task. This proposal uses several layers of Recurrent Neural Networks (RNN) and more precisely Long Short-Term Memory (LSTM) for the encoder/decoder. Nallapati et al. (2016) have proposed to define an encoder with *Gated Recurrent Units*, or GRUs, sharing their hidden state with the decoder. The disadvantage of such a model is that its vocabulary is limited to the words it has learned.

5.1.3 Large vocabulary trick

The *Large Vocabulary Trick* (LVT) was proposed by (Jean et al., 2015) for giving to a trained *seq2seq* model a vocabulary that has not yet been encountered in the learning step (Nallapati et al., 2016).

5.1.4 Attention mechanism

(Bahdanau et al., 2014) have proposed an *attention* mechanism for optimizing word generation from the source hidden state, along with the soft-max layer of the vocabulary. It is characterized by an additional vector that perform a weighted average of the hidden states of the *encoder*. Thus, the resulting vector becomes the hidden state of the *decoder*. We can then see the *attention* as a weight distribution. This technique influcences the model to learn to focus on specific parts of the input sequence vector when decoding, instead of relying solely on the hidden vector of the decoder. At each decoding step, a new attention vector, also called a context vector, is computed. It is with these two techniques that (Nallapati et al., 2016) propose an architecture that addresses a wide variety of topics and tackle the problem of unknown tokens.

5.1.5 The redundancy issue

A known problem of this architecture is repetition: for the generation of long summaries, the model tends to repeat itself and formulates the same information several times. (Vinyals et al., 2015) propose an architecture called *pointer network* which aims at establishing token correspondences between input and output. A *pointer network* aims to "point" to certain elements of the input from a probability rather than by weighting.

Another cause of this defect is that the *seq2seq* architecture does not have information about all the word positions in a document. This results from its inability to perceive the relative position of a word in relation to the global state of the positions. More exactly there is no mechanism allowing the model to take into account what has been previously generated when a token of the vocabulary is chosen (Tu et al., 2016a; Mi et al., 2016).

5.1.6 Vocabulary extension

(See et al., 2017) propose to use the global covering mechanism introduced by (Tu et al., 2016b) for minimizing the repetition problem. In order to prevent the attention vector (Nallapati et al., 2016) from strongly influencing the choice of one information against another, an extension is proposed. The extended vocabulary is built from the union of the LVT and the entirety of the words appearing in the source corpus. The decision method consists in comparing the original probability distribution with the extended vocabulary. A global coverage vector is then maintained throughout the attention technique. Such vector is performed with the sum of the non-normalized distributions of the attention vector on each previous stage of the decoder. This extension penalizes the repeated use of a token at a given location.

5.1.7 Hybrid Machine Learning

Despite the use of the global vocabulary coverage, approaches still tend to repeat the use of the same *token* of the input. If attention guarantees the use of different parts of

the encoded input sequence, an optimization on the attention decoder is required. The motivation is that the decoder can always generate redundant sentences constructed from its own hidden states. This phenomenon especially occurs when the generated sequence is long. (Paulus et al., 2017) propose to use reinforcement learning. They incorporate a discrete metric (see ROUGE in the section 5.3.1) for the selection of the right part of the input document. This technique allows the attention system to penalize the repeated use of a text area from the input.

The approach of Paulus et al. (2017) is not the unique one to proposes to use reinforcement learning (Pasunuru and Bansal, 2018; Jiang and Bansal, 2018; Q. Zhou, N. Yang, Wei, Huang, et al., 2018). Recent approaches use optimizations of the attention mechanism of See et al. (2017). This is the case of Gehrmann et al. (2018) who propose to apply a technique that add a step at the time of the attention estimation. For this, they use content targeting that determines which sentences in the source corpus should be part of the summary. This targeting is used as a "bottom-up" attention step for constraining the model according to the previously selected sentences.

5.1.8 Transformers

Vaswani et al. (2017) introduce the *transformers*, a type of architecture of *encoder-decoder*. The great success of these models encourages us, for the sake of completeness, to mention them, but these models need a lot of computational resources. The authors have developed this architecture in order to make full use of the attention mechanism. To do so, each position within the processed sequence is encoded, which allows to know all the states of each position in the text during the learning phase. The main characteristic of this proposal is that it allows the processing of these positions in parallel, thus accelerating the training stage.

In the case of the use of recurrent neural networks, the steps are sequential, while for the Transformers it is enough to have a single layer. The approach called *BERT* which uses *transformers* with great efficiency since they present competitive results in a wide variety of tasks in automatic natural language processing is proposed by Devlin et al. (2019). The overall gains in accuracy are remarkable and the transformers demonstrate a strong ability to extract critical features from text.

We note, however, that the computational power required to train the models built from this architecture is substantial. In addition to requiring a lot of computational resources, these approaches need to be trained over more iterations than approaches with LSTM, for example.

5.2 Unsupervised text summarization

This section presents unsupervised approaches, which have the advantage of not depending on the availability of a desired summary, which relies on a different data representation (*auto-encoders*). The first part focuses on the method used for the representation while the second part focuses on unsupervised approaches to text summarization.

5.2.1 Autoencoders

To perform the text summarization task, it is necessary to compress the text. Several approaches have been proposed, we detail the *auto-encoder* architecture which is often used. This one was initially introduced by (Rumelhart et al., 1985) then applied in many tasks including machine translation (Lample et al., 2017). This architecture is defined by the use of an *encoder-decoder* to which we add a reconstruction cost function. From the encoded we try to reconstruct (re-generate) the initial input.

In the case of text summarization, (Miao and Blunsom, 2016) propose to adapt the auto-encoder architecture in the context of sentence compression. They also use the *pointer network* architecture in the reconstructor (*decoder*).

(Fevry and Phang, 2018) propose an approach consisting in inserting noise in a data. The noise is represented by a subset of words extracted from another data. The goal of their contribution thus lies in the ability of their model to choose the right features of the input and reconstruct a sentence from them.

5.2.2 Sequence-to-Sequence with use of autoencoders

The work of Baziotis et al. (2019) shows the effectiveness of an auto-encoder based architecture in the context of sentence abstraction compression. Baziotis et al. (2019) introduce an *seq2seq* architecture called "SEQ^3". It consists in using two encoders/decoders, one in charge of compressing and the other one of reconstructing. The first one, called "Compressor", is in charge of producing a summary from the input text. The second, called "Reconstructor", tries to reproduce the input from the summary. They use the attention-based encoders and decoders proposed by Bahdanau et al. (2014). In an attempt to make the output summaries of the model as abstract as possible, they employ the technique *out-of-vocabulary*, inspired by Fevry and Phang (2018). This technique consists in using an external distribution to handle words that did not appear during the learning phase.

Despite competitive results, their model suffers from a major flaw. Like Nallapati et al. (2016)'s approach, SEQ^3 tends to copy the first tokens of the text to be summarized. According to Baziotis et al. (2019), this problem is due to the auto-regressive nature of the reconstructor where each word is conditioned on its predecessor, involving cascading errors. This problem would cause the compressor to choose the first words of the input text. Their approach also encounters difficulties in taking into account word positions. This feature is common to recurrent neural networks that depend on their hidden state.

West et al. (2019) propose an approach using an unsupervised extractive model to propose an abstraction-based summarization model. They use the *information bottleneck* technique defined by Tishby et al. (2001). This method is used in information theory to find the best compromise between precision and compression when summarizing a random variable X, for example. The iterative proposal of West et al. (2019), with the *information bottleneck*, searches for subsequences progressively shorter than

the proposed summary. Using only a pre-trained language model, the model succeeds in efficiently performing the sentence summarization task by extraction. They finally propose to have the outputs of the extractive model learned by an abstraction approach with a language model using *transformers*.

With emergence of BERT-based approaches, unsupervised abstractive summarization is still actively researched. Several approaches have been developed, we can note: sentence rewriting (Z. Zhang et al., 2023), AMR graph (Dohare et al., 2018) and contrastive learning (Zhuang et al., 2022).

The sentence rewriting approach is the simplest to implement, but it can be difficult to ensure that the rewritten sentences are both accurate and informative. The AMR graph approach is the most complex, but it can produce the most accurate and informative summaries. The contrastive learning approach is relatively new, but it has shown promising results. Shortly If the application requires summaries that are accurate and informative, then the AMR graph approach may be a good choice. If the application requires summaries that are generated quickly, then the sentence rewriting approach may be a good choice.

5.3 Evaluation

This section presents existing metrics for the crucial phase of evaluating and comparing existing abstractive summarization systems. In addition to these metrics, we also propose to evaluate some criteria such as the abstraction rate. These new metrics are necessary since the evaluation methods used by the state of the art are not sufficient.

CNN/DailyMail (Nallapati et al. (2016)), Gigaword (Rush et al. (2015)) and XSum (Narayan et al. (2018)) are the datasets mainly used for learning text summarization models. These datasets propose for a text, one or more summaries that have been written by humans. The appendix provides a more detailed presentation of these datasets.

Several methods have been proposed to evaluate the performance of summarization systems. The simplest and quickest approach to implement is to compare the *grams* of the candidate summary with the reference summary. A gram is an element of a sub-sequence called a gram (of size n) constructed from a sequence of data. The gram approach encounters several problems that we will raise in the rest of this section. To improve the evaluation, the authors of the state of the art approaches define additional manual approaches based for example on the criteria of language fluency as well as information coverage.

In this section we introduce our notations: for a text t, we note $r(t)$ the summary of t provided by the method we are evaluating and $r_*(t)$ the reference summary of t. In some cases, it is possible that a text has several reference summaries. We then note $R_*(t)$ the set of these reference summaries associated with the text t.

This section presents the automatic evaluation methods of the text summaries, then in a second time the manual evaluation methods. An assessment is proposed in the last part.

	With reference	With semantic	Information	Fluency	Copy rate	Question/Answer	Recall	Precision	Trust
BLEU	✓		✓	✓	✓		✓	✓	
ROUGE	✓		✓	✓	✓		✓	✓	
Pyramid	✓		✓	✓		✓			✓
METEOR	✓		✓	✓	✓		✓	✓	
pBE	✓	✓	✓			✓			✓
CompWE		✓	✓	✓		✓			✓

TABLE 5.2: Summary table of the characteristics of the automatic metrics seen in the section 5.3.1. The first two columns "With reference" and "With semantics" indicate respectively if the metric requires a human reference and if it requires the use of a Word Embedding model.

5.3.1 Auto text summary evaluation metrics

This part aims at detailing the automatic evaluation approaches, which can be observed in a synthetic way in the table 5.2. First, we propose to study the evaluation methods by gram and then some other methods.

BLEU

BLEU is an evaluation metric proposed by Papineni et al. (2002) in the context of machine translation. This method proposes to compare a candidate with one or several references by using their grams. In their proposal, Papineni et al. (2002) compute the F-measure of evaluations starting from n-grams, for $n = 1$ (unigrams) to $n = 4$. The F-measure is obtained thanks to the *accuracy* and the *recall*. In our context, the *accuracy* is the number of n-grams correctly found on the total number of n-grams proposed. The *recall* is the number of n-relevant grams found out of the total number of n-relevant grams.

We denote ng an evaluated gram and the function $Count(t, ng)$ the total number of occurrences of ng in the text t. The Equation 5.1 formalizes the gram count function ($clip$) used by BLEU.

$$Count_{clip}(r(t), ng) = min(Count(r(t), ng), Count(r_*(t), ng)) \tag{5.1}$$

This function then defines the BLEU-N metric on which the BLEU method is based:

	Candidate :	the the the the.	
	Text	$Count$	$Count_{clip}$
Reference 1	The cat is on the mat.	4	2
Reference 2	There is a cat on the mat.	4	1

TABLE 5.3: Unigram count example between a candidate and two references with BLUE. The number of match is shown as well as the number of match with the use of clip. It is possible to see that this example remains simple and naive.

$$\text{BLEU-N}(t,n) = \frac{\sum_{r_* \in R_*(t)} \sum_{ng \in r_*(t)} Count_{clip}(t,ng)}{\sum_{r_* \in R_*(t)} \sum_{ng \in r_*(t)} Count(t,ng)} \tag{5.2}$$

In the table 5.3 we can see that calculating the precision between the candidate and reference 1 gives 2/6 and not 4/6. Calculating the recall in this way allows to penalize candidates longer than their reference(s). However, when there are several references for a single candidate, it is not necessary to penalize the candidate each time. The authors define a method to answer this problem which they call *sentence brevity penalty*. To compute it, we define c the size of the candidate and g the size of the reference text. The *brevity penalty* (also called BP) is obtained as follows:

$$BP(c,g) = \begin{cases} 1, \text{if } c > g \\ e^{(1-r/c)}, \text{else } c \leq g \end{cases} \tag{5.3}$$

The final evaluation of BLEU therefore uses the accuracy normally computed with a single reference as well as a weighting with the BP. We note $N = 4$ and $w_n = 1/N$, with N the largest n-gram we compute and w_n the weighting of the BLEU result for an n-gram. We note $|t|$ the size of t. BLEU in the context of data that have multiple references per text is then obtained with the following equation:

$$\text{BLEU}(t) = BP(|t|,g) \cdot \exp\left(\sum_{n=1}^{N} w_n \text{logBLEU-N}(t,ng)\right) \tag{5.4}$$

Thus the BLEU metric evaluates a candidate against one or more references according to the number, choice and order of words. To conclude, the initial intention of this metric is, according to its authors, to quickly evaluate the output of a system against several references according to its syntactic and semantic composition. Nevertheless, BLEU has several weaknesses. BLEU does not attempt to perform a word-to-word correspondence between the candidate and its reference (see section 5.3.1). We can then say that the information match between a candidate and a reference is partially satisfied.

ROUGE

The ROUGE metric, or Recall-Oriented Understudy for Gisting Evaluation (Lin (2004)), is the most widely used evaluation method for evaluating textual summaries, either by extraction or abstraction. ROUGE-N is an extension of the BLEU metric and uses grams for its evaluation. It requires one or more reference summaries for a given text in order to evaluate a model. The equation 5.2 that is used by BLEU is also used by ROUGE. In the case of ROUGE, the accuracy calculation is differentiated for each gram, unlike BLEU.

The state-of-the-art proposals use ROUGE-1 (unigram), ROUGE-2 (bi-gram) and ROUGE-L (Lin (2004)). ROUGE-L represents the evaluation of the longest common subsequence, called *longest common subsequence* or LCS, between the candidate abstract $r(t)$ and a reference abstract $r_*(t)$. In ROUGE-L, m is the size of the candidate summary and n is the size of the reference summary. We note β the coefficient to weight the F-measure.

$$R_{LCS}(t) = \frac{LCS(r(t), r_*(t))}{m} \tag{5.5}$$

$$P_{LCS}(t) = \frac{LCS(r(t), r_*(t))}{n} \tag{5.6}$$

$$\text{ROUGE-L}(t) = \frac{(1+\beta^2) R_{LCS}(t) P_{LCS}(t)}{R_{LCS}(t) + \beta^2 P_{LCS}(t)} \tag{5.7}$$

The equation 5.7 represents the calculation of the F-measure of ROUGE-L. ROUGE-L is used to detect whether the model (or system) is learning the reference sequences correctly. It can detect several things such as recopy rate or fluency. Indeed, reference summaries are written by humans in a correctly constructed and fluent language. However, the coverage of these criteria is still very limited by the reference abstract. Evaluating an abstract summary with only ROUGE and its variants is therefore difficult since there are so many possibilities. If the system tends to be very abstract, the comparison with a single reference loses relevance. Nevertheless, ROUGE remains an efficient and quick metric to set up in order to obtain a performance indicator on a system.

Depending on the quality and the bias of the references used, ROUGE-L allows to determine if the words used by the system are comparable to those used by a human. It thus allows to judge if a machine manages to capture the same information as the human who created the reference summary. Nevertheless, this approach does not succeed in exhaustively evaluating textual summaries.

Another possible criticism of ROUGE is the lack of an exact match between the words of the candidate and the reference. In the case of a match of a candidate's gram to the reference, its position in the sequence is not taken into account. Matching a gram to the exact position of the reference is equivalent to matching it to a distant position (beginning of text vs. end of text for example). This feature becomes a problem when the desired summary is large and contains the same gram several times, for example.

First step of METEOR alignment

Candidate: The chair carver loves his job.
Reference: The chair carver loves his chair.
Matching unigram of chair:
$[(\text{chair}_{candidat\_pos\_4}, \text{chair}_{reference\_pos\_4}),$
$(\text{chair}_{candidat\_pos\_4}, \text{chair}_{reference\_pos\_7})]$

FIGURE 5.1: Application of the first step of METEOR alignment on a candidate/simple sentence. Only the case of the "chair" unigram is illustrated. The complete step consists in carrying out this treatment on all the unigrams of the candidate.

METEOR

METEOR is a method proposed by Banerjee and Lavie (2005) for evaluating machine translation systems. Like BLEU, it uses the grams that compose a text and is an extension of NIST (Doddington (2002)). It also seeks to solve problems of exact matching between a candidate gram and a reference gram that BLEU and ROUGE fail to perform.

To perform the evaluation, METEOR computes a score based on the explicit gram-to-gram correspondence between the candidate and the reference. To do this an alignment is created between them, defined as *mapping* of a candidate unigram with a reference unigram. Following this step, each unigram of each text is linked with 0 or 1 other unigram. This alignment phase consists of two steps. The first one consists in listing for a unigram all the possible corresponding unigrams. Figure 5.1 shows an example of this first step.

The second step is to identify the largest subsequence (largest n of grams) of these matches. It is then necessary to differentiate the crossings within the unigrams in such a way that at the end only a single candidate unigram is associated with a single reference unigram. Let $pos(t_x)$ be the position of the unigram t_x in the candidate, while $pos(r_y)$ is the position of the unigram t_y in the reference. Unigram crossing (UC), the method proposed by the authors, is as follows:

$$UC(t_i, t_k, r_j, r_l) = \begin{cases} true, \text{ if } (pos(t_i) - pos(t_k)) * (pos(r_j) - pos(r_l)) < 0 \\ false \text{ otherwise} \end{cases} \tag{5.8}$$

A crossover is said to exist when the result of the equation 5.8 is true. The equation 5.8 allows to know if two unigrams are in a crossing case or not. For each unigram alignment we proceed to the crossing test. In the case of multiple crossing, we choose the correspondence with the least crossing. With the example that is given in figure 5.1, the equation 5.8 shows that there is no crossing for the unigram "chair" since we obtain 0. If the exact match (same position between candidate and reference) is used, the resulting alignment at the end of the second stage can be seen in figure 5.2.

Final alignment
$[(\text{chair}_{candidat\_pos\_4}, \text{chair}_{reference\_pos\_4}), (\text{the}_{candidat\_pos\_1}, \text{the}_{reference\_pos\_1}), (\text{of}_{candidat\_pos\_3}, \text{of}_{reference\_pos\_3}), (\text{love}_{candidat\_pos\_5}, \text{love}_{reference\_pos\_5})]$

FIGURE 5.2: Final alignment found by METEOR at the end of the two steps on the chosen example in the Figure 5.1.

Instead of using the exact match method, which consists in choosing the match where the position of the candidate and reference unigrams is closest, the authors state the possibility of using weights.

When the final alignment is determined, the calculation of the F-measure is performed in the same manner as in the equation 5.2. Precision (P) is valued based on the candidate unigrams that have a match, divided by the total number of candidate unigrams. Recall (R) is valued based on candidate unigrams with a match, divided by the total number of unigrams in the reference. Finally, the F-measure with $\beta = 3$ is:

$$Fmean = \frac{10 \cdot P \cdot R}{R + 9 \cdot P} \tag{5.9}$$

The current METEOR F-measure (equation 5.9) does not take into account the largest matches every time. A penalty is used to deal with this feature. It is computed by searching for the longest matching subsequences. The function $CN(r(t), r_*(t))$ counts the *chunks number* of the largest common sequences between the proposed summary and the reference summary for a given text t. In the example provided in Figure 5.1, two sequences are found according to the positions in Figure 5.2: "the" and "of chair loves". MCU is the *max count unigram*, or the number of common unigrams between the proposed summary and the reference summary. The penalty is valued as:

$$Penalty(r(t), r_*(t)) = 0.5 \cdot \left(\frac{CN(r(t), r_*(t))}{MCU} \right)^3 \tag{5.10}$$

When calculating the penalty, if no bigram or longer match is found, the number of matching sequences is equal to the total number of unigrams that were matched. Finally, the METEOR score is defined as:

$$\text{METEOR} = Fmean \cdot (1 - Penalty) \tag{5.11}$$

If no bigram, or larger n-gram, is found, the penalty has the effect of reducing the METEOR F-measure in the equation 5.11 by 50%. When there are several references, the best score is kept for each proposed candidate. The global calculation of METEOR is performed in the same way as BLEU. METEOR allows to evaluate the fluency of a system in a better way than ROUGE or BLEU thanks to features like the use of positioning.

Summarization Content Units

A1 In 1998 two Libyans indicted in 1991 for the Lockerbie bombing were still in Libya.
B1 Two Libyans were indicted in 1991 for blowing up a Pan Am jumbo jet over Lockerbie, Scotland in 1988.
C1 Two Libyans, accused by the United States and Britain of bombing a New York bound Pan Am jet over Lockerbie, Scotland in 1988, killing 270 people, for 10 years were harbored by Libya who claimed the suspectscould not get a fair trail in America or Britain.
D2 Two Libyan suspects were indicted in 1991.

FIGURE 5.3: SCUs detection example (underlined text) with using the Pyramid method. Each sentence is assigned a letter that indicates from which reference it comes and a number that indicates the position of the sentence in the reference summary. These references are extracted from the DUC dataset.

METEOR thus proposes a more complex and efficient evaluation for a candidate/reference pair. Indeed, the notion of correspondence between the unigrams of a candidate and its reference allows the evaluation to favor systems that choose judiciously the positioning of words. It also proposes to take into account a characteristic not taken into account in the latter two, namely the correspondence of n-grams between a candidate and a reference. However, the metric does not propose a new treatment for the case of text with multiple references.

In conclusion, this proposal tackles very precisely the shortcomings found in BLEU (and ROUGE) and allows to deepen the evaluation of a system on other features. In the case of text summarization, this metric has been used alongside ROUGE-N many times, for example recently by See et al. (2017) or Guo et al. (2018).

Pyramid

Despite the importance of automatically evaluating a system's proposals, it is interesting to better understand how annotators go about creating summaries. This is what Nenkova and Passonneau (2004) proposes to do by giving a method called Pyramid. It is important to note that Pyramid is applied on datasets that have several annotations, preferably for a text. The intention is to extract from several annotations of a text, the most important parts that a system should choose. They define the Summarization Content Units (SCUs) allowing their method to automatically define what information should be contained in a text summary. An SCU is detected when a part of a reference is common with the other references. Figure 5.4 illustrates the SCUs found using the example in Figure 5.3.

When the UCSs have been found, it is necessary to give them a weight. The weight chosen by the authors is the number of references in which a UCS appears (one of the reference abstracts) for the same text. This weighting method allows us

to prioritize the UCSs and thus to define an *information score*. Using their method, the authors claim that it is possible to find a little more than 40 SCUs in abstracts of 100 *tokens*. This is justified by the fact that SCUs are found through the use of "factoid" detection (Van Halteren and Teufel (2003)).

Following these steps, a hierarchy is built with the highest weight detected at its top (in the Figure 5.4 the top is at level 4). This hierarchy is similar to a pyramid, giving its name to this method. Finally, the method outputs a score defined as a ratio between the sum of the weights of the SCUs and the sum of the weights of the SCUs closest to the top. The total of n tiers of the pyramid (T_i) is given with T_n its vertex and T_1 the first tier. The weight of a tier T_i is i and $|T_i|$ is the number of SCUs at this level. The score S_{max} of a summary with n SCUs is:

$$S_{max} = \sum_{i=j+1}^{n} i \times |T_i| + j \times (n - \sum_{i=j+1}^{n} |T_i|)$$
$$\text{with } j = \max_i(\sum_{t=i}^{n} |T_t| \geq n) \tag{5.12}$$

The score given by Pyramid is the result of a distribution of SCUs with a pre-determined weighting. This method allows to find an automatic consensus between several annotators (see section 5.3.2). It also provides an efficient way to prioritize information between several annotations.

Recent metrics

The evaluation metrics studied so far are from the field of machine translation and automatic text summarization. If we are interested in metrics for text summarization, many authors are satisfied with using ROUGE. METEOR is often used as a second metric to compare the fluency and word choice of a system. These two metrics are successors of BLEU and are mainly based on the unigrams of a corpus.

In recent proposals, it is possible to observe works such as ShafieiBavani et al. (2018b) and Honda et al. (2018) where automatic metrics that do not necessarily use human references are defined. More recently, Kryścinski et al. (2019) have proposed an approach which verifies consistency of a learned model. Their approach is based on a model that learn at the same time as the original approach and asserts the factual consistency of the summaries. SummEval Fabbri et al. (2021) is a comprehensive study of recent state-of-the-art approaches focusing on the re-evaluation of every models with a single experimental setup. Their contribution allow to update evaluation of all methods with mentioned metrics. Finally, X. Chen et al. (2022) have proposed an evaluation solution to two problems: summarization model fails to understand or capture the gist of the input text and the model over-relies on the language model to generate fluent but inadequate word.

In the case of ShafieiBavani et al. (2018b) the evaluation in absence of reference is based on the expression of five criteria:

Extracting SCUs

SCU1 (w=4): two Libyans were officially accused of the Lockerbie bombing
A1 [two Libyans] [indicted]
B1 [Two Libyans were indicted]
C1 [Two Libyans,] [accused]
D2 [Two Libyan suspects were indicted]

SCU2 (w=3): the indictment of the two Lockerbie suspects was in 1991
A1 [in 1991]
B1 [in 1991]
D2 [in 1991.]

FIGURE 5.4: Example of extraction of two SCUs on the example that is presented figure 5.3. The first UCS has a weight of 4 since it appears in 4 references. The second UCS has a weight of 3 as it appears in 3 references.

1. the semantic similarity of the words of the candidate summary with the original text;
2. the thematic relevance;
3. the relevance of the content of the summary by performing a query (Question Answering task) from the embedding of each word of the input text;
4. the coherence;
5. the capacity to interpret (presence of new words).

By means of these criteria, they claim that the proposed evaluation method allows to do without references. They describe and define these criteria and then combine them using a Support Vector Regression (SVR) model to learn a linear function that uses the same parameters as V. N. Vapnik (1999) for the purpose of combining the presented *features*. There are several problems with this approach such as the lack of consistency that a criterion has in giving a robust estimate and the dependence on the Glove *word embedding* model used to determine criteria 1 and 2. One feature that needs to be noted is that this metric does not assess fluency at all.

In a completely different way, Honda et al. (2018) propose a metric that does not take into account gram frequency and instead looks at semantic overlap using a *word embedding*. They modify and simplify the equation 5.2 by replacing the gram count with a binary function. This function uses the semantic similarity of the *word embedding* to determine the correspondence of a candidate unigram with a reference unigram. To perform this check, they define a set of basic features that represent the set of reference unigrams. This approach focuses only on the semantic feature for the purpose of evaluation. However its problem lies in the definition of the basic elements of reference and the choice of the *word embedding* model. If words are outside

the reference vocabulary, the metric cannot successfully perform the evaluation. The accuracy of this metric also depends on the model of *word embedding*.

Discussion

The problem of automatic evaluation of textual summaries is still being questioned and redefined. There are recent methods that allow to evaluate textual summaries, especially by extending ROUGE with graph theory as shown by ShafieiBavani et al. (2018a). Among all the evaluation metrics in the field of automatic text summarization, it is notable that some divergence exists within the community as to which approach to employ. The reason why ROUGE metric is commonly use lies in the definition of the task itself: there are many possible forms of writing an abstract and by using the basic elements of the language (Hovy et al., 2006) and specifically the grams, ROUGE provides a strong comparative indicator as to the quality of a system's summary output. In addition to giving an overall evaluation of systems, ROUGE allows the treatment of language fluency and information criterion through the different variants it offers. Directly extended from BLEU, it shares the same weaknesses and must eventually be complemented with other measures.

In the case of machine translation, proposals to overcome the weaknesses of BLEU have emerged. METEOR is a metric that addresses the problem of per-gram evaluation and proposes to reinforce the robustness of recall by defining a method of matching unigrams of a candidate summary to a reference summary (figure 5.2). METEOR acts directly on the information and consistency evaluation criterion. Nevertheless, the approach has a critical property which is the inference on the candidate/reference unigram pairs during several crossings. If the inference policy used is "exact", the match may be wrong in rare cases. However, METEOR remains a robust metric complementary to ROUGE. Another important feature of METEOR is its property to match grams efficiently if the candidate and reference summaries are long. It owes this feature to its alignment system (equation 5.8) and the penalty it applies on the length of matches (equation 5.10). This approach also allows us to accurately determine the copy rate that a system has.

Although the Pyramid metric is very successful for comparing human references and candidate abstracts, it can only be used when there is a representative number of references for a single text. Indeed, it is possible to use it to evaluate a candidate abstract only from several reference abstracts. The more references there are, the better the approach. However, the existing datasets for the automatic text summarization task rarely have more than three references for a text: in the two of the main datasets for training systems, namely Gigaword (Rush et al., 2015) and CNN/DailyMail (Nallapati et al., 2016)), there is only one existing reference for a text.

This metric can be used for manual evaluation of text summaries. It allows to reach a consensus between several validators. Both dependent on the *word embedding* approaches it uses and independent of language fluency, the approach proposed by ShafieiBavani et al. (2018b) allows to evaluate diversity, information and coherence of summaries.

The criterion of reliability represents the coherence between the summary produced and the input text. The evaluation of this criterion depends on the ability of the evaluation method to provide an evaluation both from the relative context (candidate/reference) and the absolute context (corpus globality).

In conclusion, it is possible to notice in the table 5.2 that the metrics performing their evaluation only from the grams encounter difficulties in front of the reliability criterion. The possible alternatives encounter other types of difficulties, such as the necessary number of references or the use of a *word embedding*. One of the solutions retained by a part of the community to evaluate textual summaries more efficiently is to use in parallel to automatic metrics, such as ROUGE and METEOR, and a manual evaluation performed by humans (section 5.3.2).

5.3.2 Manual evaluation protocol for text summarization

This section discusses the manual evaluation methods that can be observed in various state-of-the-art proposals. A manual evaluation is performed by several humans who follow a specific protocol. These methods are often done on an as-needed basis and are difficult to compare with each other. As presented hereafter, research in this area is still open today and many proposals continue to emerge.

We have chosen three proposals for automatic systems that define their own manual evaluation methods. The first section describes the method applied by Cheng and Lapata (2016) which focuses on information and fluency criteria. A second section presents the method proposed by Cao et al. (2018) which focuses on the information and vagueness criteria. Finally, the third section focuses on the method of Q. Zhou, N. Yang, Wei, Huang, et al. (2018) which adds the redundancy criterion to the first two.

Reference evaluation

This section focuses on the manual evaluation method by Cheng and Lapata (2016). It is based on the diversity of views (sufficient number of humans) and the ranking of the summaries. They require the participation of several people on twenty abstracts randomly selected from the UCR 2002 dataset (test). The texts from the dataset and the summaries made by several automatic systems (with their own) are provided at the same time. The summaries are ranked from best to worst according to two criteria: information and language fluency. The final score is an average of the rankings of each participant.

The information criterion is defined by the appearance of important terms from the original corpus in the summary. These terms can be proper nouns as well as adjectives. Fluency of language depends on the structure of the summary and more precisely on whether there are no spelling mistakes or inconsistent expressions. The authors ensure that the participants in the evaluation are not biased by using Amazon Mechanical Turk.

A total of five rankings per evaluated text are received at the end. The relevance of this evaluation has been demonstrated by its use in several publications such as Narayan et al. (2018) and Y. Liu and Lapata (2019).

Addition of information and faithfulness in evaluation

This section describes the method for manually evaluating summaries of Cao et al. (2018). In this paper the authors seek to demonstrate that summaries in their model are less likely to be ambiguous or to use bad semantics. As a reminder, one of the problems with ROUGE is that the evaluation is done at the n-gram level of the candidate and reference summaries. It regularly happens that the word order of a sentence has an impact on its semantics. In such cases, the ROUGE score does not penalize the evaluation of candidate summaries.

The proposal of this method is to define an evaluation based on the faithful interpretation of the input corpus. The manual evaluation consists in randomly drawing 100 texts from the dataset (test). A comparison of the results between their system and those of the state of the art is performed for the same texts. The participants are then asked to rank the summaries according to three classes: conform, false and uncertain. The final score is obtained by calculating, for each model, the percentage of coverage of the three classes.

The choice to evaluate by class discrimination is due to the desire to directly demonstrate whether the system is wrong or not. Often, this problem is negligible when the corpora of a dataset have references with several sentences. However, in the case where we have small summaries in output, the system can have a good score with ROUGE while being wrong.

Extended evaluation with redundancy criteria

We study the Q. Zhou, N. Yang, Wei, Huang, et al. (2018)'s approach in this section. It is possible to observe some similarities with Cheng and Lapata (2016) in the way of manually evaluating a system. Nevertheless, the authors decide to simplify the method by reducing the number of steps.

Three volunteers are chosen for the purpose of evaluating the results of the system as well as the comparison system. Fifty texts in the UCR 2002 (test) dataset. For each of these texts, summaries of the systems to be compared are also provided. Participants rank the summaries from best to worst according to three criteria: information, redundancy and overall quality.

This manual evaluation approach has several questionable points. It is true that it is difficult to perform an unbiased evaluation based on two systems without considering the reference summary. If we follow this approach precisely, we cannot compare the results of one model without comparing ourselves to the reference model chosen by the authors. The impact we observe as a result and that this evaluation is not intended to be repeated at a later date. This observation is common in manual evaluations in the field.

	Information	Fluency	Clarity	Redundancy	Inaccuracy
Cheng and Lapata (2016)	✓	✓			
Cao et al. (2018)	✓				✓
Q. Zhou, N. Yang, Wei, Huang, et al. (2018)	✓	✓		✓	✓
Hardy et al. (2019)	✓	✓	✓	✓	✓

TABLE 5.4: Summary table of manual evaluation and the criteria they address. In the case of several proposals, some criteria do not have the same label but define the same thing. Fluency can be described as "Well built" or inaccuracy can be referring to model issues.

The method offers the expression of interesting criteria to judge textual summaries. Indeed, repetition is often a problem pointed out by the community (Hardy et al. (2019)). The redundancy criterion is relevant because of its use in several publications as well as the existence of systems such as the one proposed by Ren et al. (2016). In this proposal the use of redundancy allows the system to choose the best possible proposal.

Discussion

We have seen that manual evaluation faces several problems and can be outlined with Table 5.4. An observation of this table leads to point out two criterion, clarity and redundancy, which are poorly represented in the literature. The first problem to tackle is the definition of the manual evaluation. Indeed, the great variety of proposals demonstrates an uncertainty as to the choice of the right methodology to apply. A second problem concerns the choice of criteria and their definition. In the case of Cao et al. (2018), the definition of classes is relevant but remains applicable mainly in their context.

What we retain from these manual evaluation methods is their "tailor-made" character. This character results in a better understanding of the strengths and weaknesses of a specific system. However, the wide variety of methods and criteria makes it impossible to compare proposals unless one wants to confront only a specific system. As we have seen, the use of redundancy is an optimization criterion often used in the context of abstract text summarization. Thus, while there are definite advantages in the ability to evaluate a system on a small subset of data, the methodology is cumbersome for evaluators to follow.

5.3.3 Overview

The ROUGE metric is still the most widely used method to evaluate text summaries provided by an automatic system. Recent contributions are accompanied by manual evaluation or several other automatic metrics. We use ROUGE to be able to position our approaches with the state of the art. It is possible to find in the table 5.5 the metrics complementary to ROUGE that we use. These metrics aim to cover the criteria raised in this section. They also allow to describe the characteristics of the approaches we propose.

Criteria	Description
Abstraction rate	Comparison of similar words between the input and the generated summary.
Compression rate	Reduction ratio of size between input and output.
CompWE	Previously mentioned metric (Section 5.3.1) that evaluates fluency.
Processing time	Indicates the performance of the calculation time during training.

TABLE 5.5: Table of the chosen metrics and their description.

5.3.4 Conclusion

We note that there are similarities between the proposals in the machine translation domain and the text summarization domain. The techniques that are advanced in one can positively impact the other. However, a number of techniques specific to the field of text summarization are missing. In the context of unsupervised learning approaches, it is possible to observe promising specific techniques (Baziotis et al. (2019), Tishby et al. (2001)).

It is also possible to note a lack of correlation between the abstraction of an abstract and the technique put forward by an approach. In our context, it is important to know the degree of abstraction of an approach. Indeed, as raised in section 5.3.1, the evaluation of the abstraction of an approach strongly depends on the dataset. We would like to be able to demonstrate that it is possible to use ROUGE while evaluating the abstraction. Copy rate and compression, as well as semantic similarity, are promising tools to perform this evaluation.

5.4 Robust abstractive summarization

In the previous section we have questioned the evaluation process of abstractive summarization. This section tackles some of the problems of unsupervised abstractive summarization. Section 4.4 of Chapter 4 has presented how the raw space $\mathcal{X}$ can be difficult to handle. For this reason, we have proposed different solutions and several ways of handling non-robust models or rather polluted data. Thus, we propose to explore two options that a generative model can benefit from: outlier removal in pre-processing step and adding robust techniques to the model (if possible). The purpose of this section is to present two case studies that demonstrate the use of our work in tasks other than outlier detection.

5.4.1 Outlier removal for robust learning

The first case study is the outlier filtering process. In this part we aim to improve the results of abstractive summarization by working on the original raw space $\mathcal{X}$.

Problem and motivation

Abstractive summarization is a task that is recently popular thanks to advances in neural networks. While the supervised approaches have state of the art results, the

Model	Rouge-1	Rouge-2	Rouge-L
supervised			
Rush et al. (2015)	29.55	11.32	26.42
distillBART-Gigaword	**35.73**	**16.29**	**32.07**
unsupervised			
Y. Wang and H.-Y. Lee (2018)	21.26	5.60	18.89
Zhuang et al. (2022)	28.10	**11.63**	24.14
Baziotis et al. (2019)	25.39	8.21	22.68
Baziotis et al. (2019) with 5%	24.11	9.34	25.18
Baziotis et al. (2019) with 10%	**28.47**	11.02	**27.59**

TABLE 5.6: Results of Seq^3 after application of outlier removal of the 5% and 10% most outlying documents. The corpus of training and evaluation is Gigaword. For comparison, we give a short baseline with supervised approach.

literature focusing on unsupervised abstractive summarization has not as much popularity. Yet, we can observe several works that aim to close the gap between supervision and non supervision. However, for every kind of approaches we observe datasets that are heavy. While they allow to perform a neural network training , these datasets are often gathered with automatized techniques and methodologies. One problem is that it is nearly impossible to manually check each document of a multi million document corpus (unless one is patient enough). We propose to use outlier analysis on the raw dataset and remove a part of the outlying points.

Experimental setup

For our experimental setting, we chose Seq^3 (Baziotis et al., 2019) that is an unsupervised approach based on multiple seq2seq layers. Two seq2seq exactly are working as a compressor and a decompressor, with a language model prior that handle the bottleneck. We use the dataset Gigaword that is a well-known corpus for performing abstractive summarization. It has an original train split of 3 803 957 documents and has short text compared to CNN Dailymail. For the baseline we have chosen to fine tune a distillBART model on 2 epochs. This approach has been originally trained on CNN Dailymail and Xsum. We also present SCR approach from Zhuang et al. (2022) and we record results they have introduced in their contribution. All of the results have been averaged on ten runs for mitigating some biases.

Finally, we build our outlier analysis model with an ensemble of KNN and LSA. For KNN, we take twenty models with $k \in [20, 120]$ in which we start from 20 and end up to 120 after stepping of 5 each time. All of them are using euclidean distance and the tfidf representation of text is chosen. We proceed similarly with LSA and its ranks. Based on results of Section 4.3.1, we select the *average* late fusion technique. Also, we propose to compare results on outlier removal based on contamination rate $\nu = 0.05$ and $\nu = 0.10$.

Results and discussion

The results of the Table 5.6 display clear benefits to remove outlying points with outlier detection. Based on these results, we can assume several things about Gigaword and Seq^3. First, Gigaword has a large number of documents that may cover a high number of topics and can also integrate contradictory inside the corpus. Some of the document can be similar but does not carry the same information, or without supervision it can be complicated to make learn such rules. In this approach, the model relies on the language model and its capability to make distinction.

With the analysis of scores of the outlier detection model, we can see that there exist a real multilevel representation of texts. We identify three levels, with most of the point being near from each other (it is still reassuring), and two others with one among them that have a consequent gap of value. A last level has a significant gap that make its associated observation outliers, but instead low outliers. It can be difficult to handle those last outliers because some of them may be outliers and the rest outlier, or vice versa.

5.4.2 Robust Subspace Recovery AutoEncoder for unsupervised summarization

We have previously observed a gain in performance from removing outlying points of the raw space. In this section we focus on applying robust technique for improving unsupervised abstractive summarization.

Problem and motivation

One drawback of outlier removal is the suppression of data that can actually help the model to best perform. In some way, it is a solution that is prompt to increase the performances but can also hide the real issue. The problem of Seq^3 is that it does not succeed to handle certain kind of documents. There exist several method of helping a machine learning model to be more robust against peculiar observations. Adversarial learning is one of them numerous approaches in the literature (Y. Liu, Li, et al., 2019) or also using ensemble learning (J. Chen, Sathe, et al., 2017). Previously in Section 5.4.1, we have witnessed that robust subspace recovery layers was a great addition for autoencoders. Because Seq^3 heavily relies on the language model prior and the hidden states, in addition of the original reconstruction loss we propose to also use our REATO loss function from Section 4.2 described in Equation 4.5.

Experimental setup

The experimental setting is the same as for outlier removal, but instead we do not train any outlier detection model (obviously). For the implementation, we keep the original code of Seq^3 with adding the reconstruction term and also the RSR layers on the bottleneck of each seq2seq. The experiment is processed once again on the Gigaword dataset and is evaluated with the Rouge metrics.

Model	Rouge-1	Rouge-2	Rouge-L
supervised			
Rush et al. (2015)	29.55	11.32	26.42
distillBART-Gigaword	**35.73**	**16.29**	**32.07**
unsupervised			
Y. Wang and H.-Y. Lee (2018)	21.26	5.60	18.89
Zhuang et al. (2022)	28.10	**11.63**	24.14
Baziotis et al. (2019)	25.39	8.21	22.68
Baziotis et al. (2019) with 10%	28.47	11.02	27.59
RSR-Seq^3	**29.73**	11.16	**28.97**

TABLE 5.7: Results of Seq^3 after application of outlier removal of the 5% and 10% most outlying documents. The corpus of training and evaluation is Gigaword. For comparison, we give a short baseline with supervised approach.

Results and discussion

Once again we observe an improvement of the results with this case study. While this last setup seems to be performing better than outlier removal, the results are still near from each other. The observation that can naturally occurs in this scenario is if the results are similar between the suppression of some data and the learning process on how to handle them, Gigaword possesses outlying point that are confusing models. Results of our approach RSR-Seq^3 are presentend in Table 5.7. Similarly to results of Section 5.4.1, ROUGE-2 results are outperformed by SCR (Zhuang et al., 2022) but still presents benefits from our robust autoencoders. We can also observe that RSR-Seq^3 outperform Rush et al. (2015) which is a supervised approach.

While this section introduces an application of our thesis contributions, we want to explore a better robust representation based on attention head of recent transformers-based language models. As we have successfully demonstrated with a short experimentation that abstractive summarization can benefit from robust representation, we want to explore in future works how it can be evaluated with work from Section 5.3.1 and Section 5.3.2. This evaluation can precisely suggest what is improved with our approach (fluency, ...).

5.5 Conclusion

In this chapter we have introduced an experimental application of work detailed in this thesis over abstractive summarization task. We have first presented the purpose and challenges of abstractive summarization and taken the opportunity to elaborate a case study in our context. While abstractive summarization can be tackled with either supervised or unsupervised approaches, we focus to unsupervised abstractive summarization. Precisely, this setup is challenging due to a complex structure (Section 5.1.1). In this context we performed an overview of the recent advances.

If difficulties can be encountered in creating an abstractive summarization approach, the natural characteristics of the task, which can be difficult to formalise

properly, make the evaluation step crucial. We introduce an overview of the evaluation step in Section 5.3. Both automatic evaluation and manual evaluation are compared and we present how they can be complementary with each other. In this context, we introduce our evaluation, which is an ensemble of automatic and manual criteria for evaluating abstractive summarization approaches. This evaluation can perform quantitative evaluation through traditional ROUGE and qualitative evaluation with abstraction rate, compression rate, CompWE (Section 5.3.1) and processing time. It helps to evaluate the main characteristics of abstractive summarisation and distinguish it from extractive summarisation.

Finally, we introduce two case studies based on our research contributions with outlier removal in training corpus and outlier awareness while training. In Section 5.4.1 we have introduce a preprocessing approach that tackles outlying observation before the training step. This approach and the conducted experiments demonstrate that Gigaword has several level of normal documents. Such observation can be useful for designing future approaches. Regarding this case study, future perspectives are a proposition of an unsupervised independent outlier removal and unsupervised contextual outlier removal.

In the second case study, we introduce RSR-SEQ^3 which is a robust and unsupervised approach performing abstractive summarization. In this contribution, we change the original autoencoders from Baziotis et al. (2019) with our REATO autoencoder. Results of this experiments are promising and our approach outperforms state-of-the-art approaches of unsupervised abstractive summarization. Our approach also outperforms older supervised methods from the literature and is surprisingly efficient with all corpora. Future perspectives opened with this case study lie in the presentation of the results under our introduced evaluation. Another perspective lies in an interesting approach consisting to explain furthermore the model with the integration of special outliers (independent or contextual).

With the results we have presented in Section 5.4, our textual outlier analysis can presented as a stepping work for improving performance of natural language processing tasks. Perspectives on abstractive summarization are numerous, but we can highlight several limitations. The performed evaluation in Section 5.4 is clearly simple and does not benefit from the Section 5.3 conclusions. Indeed, for getting a better understanding of the shortcomings of one model, all the listed criteria of Table 5.5 may be required. Another future work can be addressed, and it is the lack of comparison of usage of outlier ensemble instead of the RSR layer. On the other hand, XAI challenges can be directly involved in this context, and the interesting benefit of abstractive summarisation is the ability to see the impact of multiple changes through adversarial or counterfactual observations.

Chapter 6

Conclusion and perspectives

6.1 Outline of the contributions

The principal topic of this thesis work is the analysis of outliers in textual data. Since there are several approaches to deal with text, we have specified in our work that we are interested in the semantic level of a text. This interest also follows current problems encountered by a number of data mining approaches. These problems include the question of confronting a machine learning model with a desired or undesired perturbation. This perturbation can take many forms: noise in a corpus such as empty text or text filled with a single character, text representing information far from the subject of study (wrong category of news papers) or different data sources. In a second step, it is also a question of allowing a model to correctly process data that are more or less distant from what it has learned well.

6.1.1 An overview of outlier analysis

The Chapter 1 and the Chapter 2 introduce the basic notions to deal with outlier analysis in a general framework. The problem inherent in the outlier detection task is its freshness and the growing number of surveys and reviews in the field. In order to better understand the difference between the subtasks and the outlier formalism, we have reviewed a large number from state-of-the-art works. Our contribution lies in the consolidation of the differences between an outlier and an anomaly. We have conducted a comprehensive effort to formalize and define what an outlier can be depending of the application or research domain.

We have proposed in the Chapter 2 a summary as well as an overview of the outlier detection task. Various connections between applications are not completely detailed in the literature. Our overview bridges the gap by connecting similar tasks like outlier detection, anomaly detection, fake news detection, spam detection, and many others. We have stated their characteristics and how they can be related with each other. Thus, we introduced the different notions and definitions of the literature before proposing several taxonomies. Unlike reference surveys, we have proposed a detailed comparison of state-of-the-art outlier and anomaly taxonomies. This study allowed us to extract a general taxonomy that can be applied to numerous kind of data, and particularly text.

Finally, we introduced a common formalism for studying unsupervised outlier detection approaches. We have tackled most popular and recent methods from the literature and categorized them in a dedicated taxonomy. The purpose of presenting a large number of approaches is motivated by the lack of a sufficient amount of methods for text data. Our contribution lies in the adaptability of this overview of outlier detection that can be applied to any other kind of data than text. We have presented a generic ground for working with popular and recent methods of the task, and can be easily extended for similar applications.

6.1.2 Outlier detection in text

Recently, a prolific and blossoming literature can be observed among anomaly detection and outlier detection tasks. Unfortunately, we do not observe the same phenomena for outlier detection in text data. Although there exists several recent works that are actively interested in performing outlier detection with text, there is no existent survey dedicated to such kind of data. In absence of such contribution, we have noted several confusions among reference approaches of the literature.

While we have presented common knowledge for tackling text with unsupervised machine learning, we have addressed outlier detection with text data through our introduced generic ground. In our first contribution, we introduced a proper definition of a textual outlier as well as the different levels that it can occurs. Furthermore, we define syntax and semantic outliers which can be connected to different application and tasks. These definitions allowed our work to be connected with other tasks like fake news detection, email spam detection, sentiment analysis, ... With such connections, we have presented a survey of state-of-the-art approaches that have been proposed for outlier detection and compatible other applications.

One problem of performing outlier detection in text, is that references works often lack a proper analysis of what is detected and stops at the surface while not taking into account the specifies of text. This a recurring problem, and contributed to mitigate it with the introduction of a dedicated textual outlier taxonomy. Our taxonomy properly define what an independent outlier and a contextual outlier are. This contribution is critical and results to get another glimpse of the detection problem. For assessing this, we have introduced GenTO[1], a generic algorithm which preprocess corpora with either independent or contextual contamination.

Based on GenTO, we have proposed a comprehensive experimental study of state-of-the-art approaches for both independent and contextual outliers. This study revealed that independent outliers are more difficult to tackle than contextual outliers. Most of reference works are mostly contaminating their corpora with independent contamination, leading in an evident bias against works that mix both outliers randomly. Thus, we have proposed a comprehensive study of how to detect outliers in text and the associated problematic. GenTO when associated with the conducted experimental study also reveals that traditional methods can outperform very recent works. With the same experimental protocol, non-text dedicated approaches can outperform

[1]This work was published at the EGC Textmine 22 workshop (Pantin et al., 2022).

dedicated ones. It contributes to question the recorded results of the literature. Another contribution of GenTO lies in its generic setting which allows to apply it to any corpus.

6.1.3 Outlier ensemble in text

As outlier ensembles are very rarely applied for texts, we have also proposed a chapter dedicated to them. The motivation for their use lies in the will to perform bias reduction to different parameterization choices. We have made a comparison between these methods and the data fusion task, which shares some characteristics. In this context we introduced REATO, a robust subspace recovery ensemble autoencoder approach for text data. Unlike recent reference approaches, REATO is independent of the text representation model and tackles the problem of locality in its latent representation. While there are not any existing work performing outlier ensemble with text, our approach successfully outperform state-of-the-art methods. Our method also displays an incredible score robustness for contextual outliers, as opposed to other approaches.

Ensemble methods can be performed at different levels and we have introduced another approach which relies to represent text with polarity features. Our introduced representation, PoLSA, maintains a richer and multimodal representation of the text and increases performance of reference approaches with different kind of corpora (sentiment analysis and news papers). PoLSA is completed with an experimental study which consists to a progressive addition of early fusion and late fusion (outlier ensemble). If the text has several level of information that he can carry, the ability to integrate different specialized features (opinion, semantic, rules, text statistics, emotions, ...) is an important factor for motivating the choice of early fusion.

The introduction of PoLSA has also introduced possible extensions of our work with XAI domain. With a dimension reduction technique and polarity features, our contribution tackles the problem of explaining decisions of the trained model. PoLSA can be used for easing such explanation, as demonstrated in Section 4.4. The conducted research in Chapter 4 has open the perspective of performing outlier ensemble with text while tackling XAI connections.

6.1.4 Case studies

In the Chapter 4 we focus on the issues of interpretability and explicativity in our context. The given use case is to extract significant attributes, i.e. features that positively influence the decisions of the model. It is also a question of taking into account the added representations through data fusion. In our use case, we demonstrated that it is possible to extract features that are more important than others as well as the interpretation of the polarity of the text. In doing so, we were able to initiate a step of understanding the predictive model while extracting a simpler representation. We have thus introduced a way of explaining the model in terms of semantic and opinion attributes.

The Chapter 5 introduces the challenges of text summarization by abstraction as well as the recurrent problems. This context allowed us to illustrate two cases where outlier analysis becomes a tool for improving an approach. Thus, we introduce a comparison of the results of abstractive summarisation models with denoised data (outlier removal). We also present a comparison of results using a robust version of the chosen model. In both cases, we have shown that outlier analysis is very successful. Our approach, robust subspace recovery sequence-to-sequence-to-sequence (RSR-Seq^3), appears to outperform state-of-the-art approaches of unsupervised abstractive summarization. This work open promising perspective for numerous other applications with text data.

6.2 Significance and limitations

Our research on outlier detection in text data holds significant importance as it addresses the unique challenges posed by the contextual nature of textual information. By developing specialized methods that consider the intricate relationships between words and phrases, we aim to enhance the accuracy and robustness of outlier detection approaches in this domain. The expected impact of our research is twofold: in academia and industry.

In academia, our work enriches the body of knowledge on outlier analysis, providing valuable insights into handling contextual information in outlier detection tasks. The proposed methodologies and experimental findings serve as a reference for researchers and practitioners interested in exploring outlier detection in text data.

In the industry, the applications of our research are wide-ranging. The developed techniques can be applied in domains such as cybersecurity, financial fraud detection, healthcare analytics, and social media monitoring. Detecting unusual patterns and outliers in textual data helps businesses and organizations to identify hidden anomalies, gain actionable insights, and improve decision-making processes.

Our research offers novel insights into the field of outlier detection in text data by delving into the intricacies of contextual analysis. By addressing the challenges of identifying independent outliers and contextual outliers, we pave the way for future research to explore more sophisticated techniques and refine existing approaches. The proposed taxonomy and evaluation framework provides a structured methodology to compare different outlier detection methods effectively. It also open doors to the development of more specialized algorithms for specific text domains, such as legal documents, scientific literature, and social media content. Moreover, our work on outlier ensemble methods and interpretability contributes to the growing interest in transparent and explainable artificial intelligence systems. This spark further investigations into incorporating human domain knowledge into the outlier detection process and leveraging ensemble techniques for improved performance.

While our research presents contributions to the field of outlier detection in text data, it is essential to acknowledge its limitations. One such limitation lies in the

reliance on labeled data for evaluating some of the proposed algorithms. Future research can focus on exploring semi-supervised or unsupervised approaches to mitigate the labeling burden and extend the applicability of outlier detection to scenarios with limited labeled data. Moreover, as the landscape of text data evolves, challenges related to noisy, unstructured, and multilingual text may emerge. Exploring techniques that can handle these complexities will be critical to advancing the state-of-the-art in outlier detection for diverse textual information.

Our work opens up promising avenues for future research in fine-tuning the hyperparameters of ensemble methods and exploring novel techniques for combining multiple outlier detection models effectively. Additionally, investigating the interpretability of ensemble outcomes will contribute to building trust in the decision-making process based on outlier detection results.

6.3 Future works

6.3.1 Towards a unified textual outlier detection

In the various experiments, we have focused on the semantic aspect of the text. What about the syntax of a text, and how would this change the methodology applied in our work? Firstly, outlier detection at the lexical level already has a form in the Fake news detection and Plagiarism tasks. However, these tasks often make a combined use of lexical and/or semantic text representation. A future work is therefore to provide a study and comparison of lexical outlier detection methods. The main motivation is to be able to add to our current work, the possibility of formalising outlier detection on the largest number of situations. Thus, similarly than for plagiarism detection, syntax analysis can highly benefit detection of contextual outliers. Such remark is motivated with the assessment that, for instance, a technical article is not written similarly than a sport article (structure, vocabulary, ...).

In the same way that the task of sentiment analysis uses both morphological rules on the text and semantic attributes, we could observe that it was possible to do the same for outlier detection in the Chapter 4. It represents an interesting perspective for tackling special documents.

In our overview of outlier analysis and outlier analysis with text, we have categorized the usage of graph as an application or a data problem. Promising work Akoglu et al. (2015), Deng and Hooi (2021), and Ma et al. (2021) have presented usage of graph neural network approaches or graph-based structure for tackling outlier detection as a blossoming challenge.

Our approach GenTO is supervised approach that preprocess corpora and perform independent or contextual contamination. One perspective consists to investigate unsupervised contamination through a chosen set of methods. The idea of such perspective is to allow the possibility to perform contextual contamination to any corpus, not only for those with topic hierarchy. A large number of corpora can be candidate to such perspective, and can tackle the problem of performing contextual contamination with any language corpus.

6.3.2 Toward robust machine learning

There are three areas in which our work can intervene. There is the solution of cleaning and pre-processing the data sets before starting the learning stage. It is also possible to give the possibility to a model to represent with more robustness the data it ingests. We were able to demonstrate the possibility of these last two points in Chapter 4 and Chapter 5. In the former, the possible bias brought by the data as well as the parameters of the different approaches is dealt with through the use of method ensembles and data fusion. Indeed, these two methods have allowed not only to increase the global performance of the models, but also the stability in front of different pollution scenarios. Concerning the second chapter mentioned, we used on a natural language processing task the replacement of a part of the model sensitive to the input data. We have shown that by replacing the normal autoencoders by our REATO autoencoders, the performances of ROUGE-1 and ROUGE-2 have been greatly improved.

The third axis focuses on the definition and interpretation of outliers or anomalies once the results have been obtained, or before learning. The Chapter 2 and the Chapter 4 deal with this subject in depth. What can be said from this work is that the appearance of outliers is often uncontrolled and that knowing how to find them is as important as knowing why they appeared.

As a result of all this work, the opening up of different subjects is notable. The first perspectives envisaged concern an unsupervised evaluation method. In the literature, one can find works such as Marques et al. (2020) and Campos et al. (2016). However, in the context of the text, it is possible to add singular characteristics to this type of data. The creation of a metric taking into account size, vocabulary richness or the definition of a model dealing with non-vocabulary words seems to be a promising approach. Such a metric could benefit the task of building new datasets, as well as defining unsupervised learning policies.

A second perspective lies in the use of outliers to generate adversarial observations and thus make a model more robust. Identifying independent and contextual outliers have pave avenues for performing a generation of adversarial instances. On the other hand, outlier analysis can also be positive for counterfactual explanation. Furthermore, work conducted in the Chapter 5 indicates that text generation can be influenced. Such influence can be estimated with different metric, and promising work can involve a novel taxonomy definition regarding the kind of performed robustness.

Finally, one perspective can be addressed with connections of our work with different kind of data like images. RSRAE (Lai et al., 2020) is an approach that performs on image and text with great success. Based on this observation, our work can be declined for other kind of data and image are absolutely a promising direction. Transfer learning can be an interesting domain for tackling robust learning with several kind of applications. There exist application like image summarization that can benefits from our work.

Bibliography

Abraham, Bovas and George E. P. Box (1979). "Bayesian analysis of some outlier problems in time series". en. In: *Biometrika* 66.2, pp. 229–236 (cit. on pp. 11, 52).

Aggarwal, Charu C. (2017a). *Outlier analysis*. eng. Second edition. Cham: Springer (cit. on pp. 11, 12, 16, 17, 21, 22, 25, 26, 28, 33, 34, 38, 41, 47, 48, 52, 72, 100, 102).

— (2017b). "Outlier Detection in Categorical, Text, and Mixed Attribute Data". In: *Outlier Analysis*. Cham: Springer International Publishing, pp. 249–272 (cit. on pp. 55, 68, 74).

Aggarwal, Charu C., Alexander Hinneburg, and Daniel A. Keim (2001). "On the Surprising Behavior of Distance Metrics in High Dimensional Space". In: *Database Theory — ICDT 2001*. Ed. by Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Jan Van den Bussche, and Victor Vianu. Vol. 1973. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 420–434 (cit. on p. 110).

Aggarwal, Charu C. and Saket Sathe (Sept. 29, 2015). "Theoretical Foundations and Algorithms for Outlier Ensembles". In: *ACM SIGKDD Explorations Newsletter* 17.1, pp. 24–47 (cit. on pp. 20, 40, 73, 93, 100, 102, 103).

Aggarwal, Charu C. and Philip S. Yu (2001). "Outlier detection for high dimensional data". In: *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*. SIGMOD/PODS01: ACM SIGMOD International Conference on Management of Data. ACM, pp. 37–46 (cit. on p. 11).

Aggarwal, Vaibhav, Vaibhav Gupta, Prayag Singh, Kiran Sharma, and Neetu Sharma (2019). "Detection of Spatial Outlier by Using Improved Z-Score Test". In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). Tirunelveli, India: IEEE, pp. 788–790 (cit. on p. 27).

Ait-Saada, Mira and Mohamed Nadif (2023). "Unsupervised Anomaly Detection in Multi-Topic Short-Text Corpora". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1384–1395 (cit. on p. 70).

Akoglu, Leman, Hanghang Tong, and Danai Koutra (2015). "Graph based anomaly detection and description: a survey". In: *Data mining and knowledge discovery* 29, pp. 626–688 (cit. on p. 151).

Allan, Edward G., Michael R. Horvath, Christopher V. Kopek, et al. (2008). "Anomaly Detection Using Nonnegative Matrix Factorization". In: *Survey of Text Mining II*. Ed. by Michael W. Berry and Malu Castellanos. London: Springer London, pp. 203–217 (cit. on pp. 52, 68, 69, 74).

Almardeny, Yahya, Noureddine Boujnah, and Frances Cleary (2020). "A Novel Outlier Detection Method for Multivariate Data". In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1 (cit. on p. 35).

Almeida, Tiago A, José Maria G Hidalgo, and Akebo Yamakami (2011). "Contributions to the study of SMS spam filtering: new collection and results". In: *Proceedings of the 11th ACM symposium on Document engineering*, pp. 259–262 (cit. on p. 79).

An, Jinwon and Sungzoon Cho (2015). "Variational autoencoder based anomaly detection using reconstruction probability". In: *Special lecture on IE* 2.1, pp. 1–18 (cit. on p. 103).

Bahdanau, Dzmitry, Kyunghyun Cho, and Y. Bengio (Sept. 2014). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *ArXiv* 1409 (cit. on pp. 126, 128).

Banerjee, Satanjeev and Alon Lavie (2005). "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72 (cit. on p. 133).

Basu, Sabyasachi and Martin Meckesheimer (2007). "Automatic outlier detection for time series: an application to sensor data". In: *Knowledge and Information Systems* 11, pp. 137–154 (cit. on p. 11).

Baziotis, Christos, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos (June 2019). "SEQˆ3: Differentiable Sequence-to-Sequence-to-Sequence Autoencoder for Unsupervised Abstractive Sentence Compression". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 673–681 (cit. on pp. 128, 142, 143, 145, 146).

Beckman, R. J. and R. D. Cook (1983). "Outlier.....s". In: *Technometrics* 25.2, pp. 119–149 (cit. on p. 52).

Ben-Gal, Irad (2005). "Outlier Detection". en. In: *Data Mining and Knowledge Discovery Handbook*. Ed. by Oded Maimon and Lior Rokach. New York: Springer-Verlag, pp. 131–146 (cit. on p. 18).

Bengio, Y., A. Courville, and P. Vincent (2013). "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8, pp. 1798–1828 (cit. on pp. 10, 32).

Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent (2000). "A neural probabilistic language model". In: *Advances in neural information processing systems* 13 (cit. on p. 59).

Berry, Michael W., Murray Browne, Amy N. Langville, V. Paul Pauca, and Robert J. Plemmons (Sept. 2007). "Algorithms and applications for approximate nonnegative matrix factorization". In: *Computational Statistics & Data Analysis* 52.1, pp. 155–173 (cit. on p. 68).

Bhattarai, Bimal, Ole-Christoffer Granmo, and Lei Jiao (2020). "Measuring the Novelty of Natural Language Text Using the Conjunctive Clauses of a Tsetlin Machine Text Classifier". In: *arXiv:2011.08755 [cs]* (cit. on p. 55).

Bird, Steven, Ewan Klein, and Edward Loper (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc." (cit. on p. 111).

Bishop, Christopher M. (2006). *Pattern recognition and machine learning*. Information Science and Statistics. Springer New York. 738 pp. (cit. on pp. 11, 41).

Blazquez-Garcia, Ane, Angel Conde, Usue Mori, and Jose A Lozano (2021). "A review on outlier/anomaly detection in time series data". In: *ACM Computing Surveys (CSUR)* 54.3, pp. 1–33 (cit. on p. 11).

Bleiholder, Jens and Felix Naumann (2009). "Data fusion". In: *ACM computing surveys (CSUR)* 41.1, pp. 1–41 (cit. on p. 99).

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2016). "Enriching Word Vectors with Subword Information". In: *arXiv preprint arXiv:1607.04606* (cit. on p. 103).

Boukhaled, Mohamed Amine and Jean-Gabriel Ganascia (2014). "Probabilistic anomaly detection method for authorship verification". In: *International Conference on Statistical Language and Speech Processing*. Springer, pp. 211–219 (cit. on p. 65).

Breiman, Leo (1996). "Bagging predictors". In: *Machine Learning* 24.2, pp. 123–140 (cit. on p. 40).

Breunig, Markus M., Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander (2000). "LOF: identifying density-based local outliers". In: *ACM SIGMOD International Conference on Management of Data*. Dallas, Texas, United States: ACM Press, pp. 93–104 (cit. on pp. 31, 111).

Brown, Tom, Benjamin Mann, Nick Ryder, et al. (2020). "Language models are few-shot learners". In: *Advances in neural information processing systems* 33, pp. 1877–1901 (cit. on p. 61).

Campos, Guilherme O., Arthur Zimek, Jörg Sander, et al. (2016). "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study". In: *Data Mining and Knowledge Discovery* 30.4, pp. 891–927 (cit. on p. 152).

Cao, Ziqiang, Furu Wei, Wenjie Li, and Sujian Li (2018). "Faithful to the original: Fact aware neural abstractive summarization". In: *AAAI Conference on Artificial Intelligence* (cit. on pp. 139–141).

Celik, Mete, Filiz Dadaser-Celik, and Ahmet Sakir Dokuz (June 2011). "Anomaly detection in temperature data using DBSCAN algorithm". In: *2011 International Symposium on Innovations in Intelligent Systems and Applications*. 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA). Istanbul, Turkey: IEEE, pp. 91–95 (cit. on pp. 70, 74).

Chandola, Varun, Arindam Banerjee, and Vipin Kumar (July 2009). "Anomaly detection: A survey". en. In: *ACM Computing Surveys* 41.3, pp. 1–58 (cit. on pp. 15, 17, 19, 21, 22, 25, 47, 52).

Chang, Chia-Yang, Shie-Jue Lee, Chih-Hung Wu, Chih-Feng Liu, and Ching-Kuan Liu (2021). "Using word semantic concepts for plagiarism detection in text documents". In: *Information Retrieval Journal* (cit. on p. 55).

Chen, Jing and Yang Liu (2011). "Locally linear embedding: a survey". In: *Artificial Intelligence Review* 36, pp. 29–48 (cit. on p. 105).

Chen, Jinghui, Saket Sathe, Charu Aggarwal, and Deepak Turaga (2017). "Outlier detection with autoencoder ensembles". In: *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, pp. 90–98 (cit. on pp. 93, 100, 102–104, 107, 144).

Chen, Xiuying, Mingzhe Li, Xin Gao, and Xiangliang Zhang (2022). "Towards improving faithfulness in abstractive summarization". In: *Advances in Neural Information Processing Systems* 35, pp. 24516–24528 (cit. on p. 136).

Chen, Zhaomin, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau (2018). "Autoencoder-based network anomaly detection". In: *2018 Wireless Telecommunications Symposium (WTS)*, pp. 1–5 (cit. on p. 103).

Cheng, Jianpeng and Mirella Lapata (2016). "Neural Summarization by Extracting Sentences and Words". In: *Association for Computational Linguistics* (cit. on pp. 139–141).

Chikodili, Nwodo Benita, Mohammed D. Abdulmalik, Opeyemi A. Abisoye, and Sulaimon A. Bashir (2021). "Outlier Detection in Multivariate Time Series Data Using a Fusion of K-Medoid, Standardized Euclidean Distance and Z-Score". In: *Information and Communication Technology and Applications*. Ed. by Sanjay Misra and Bilkisu Muhammad-Bello. Vol. 1350. Cham: Springer International Publishing, pp. 259–271 (cit. on p. 27).

Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, et al. (2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734 (cit. on p. 125).

Chomsky, Noam (1956). "Three models for the description of language". In: *IRE Transactions on information theory* 2.3, pp. 113–124 (cit. on p. 64).

Clark, Kevin, Urvashi Khandelwal, Omer Levy, and Christopher D Manning (2019). "What does bert look at? an analysis of bert's attention". In: *arXiv preprint arXiv:1906.04341* (cit. on p. 119).

Clark, Kevin, Minh-Thang Luong, Quoc V Le, and Christopher D Manning (2020). "Electra: Pre-training text encoders as discriminators rather than generators". In: *arXiv preprint arXiv:2003.10555* (cit. on p. 73).

Collobert, Ronan and Jason Weston (2008). "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th international conference on Machine learning*, pp. 160–167 (cit. on p. 59).

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297 (cit. on p. 32).

Dang, Xuan Hong, Ira Assent, Raymond T Ng, Arthur Zimek, and Erich Schubert (2014). "Discriminative features for identifying and interpreting outliers". In: *2014 IEEE 30th international conference on data engineering*. IEEE, pp. 88–99 (cit. on p. 120).

Davidson, Ian and Selvan Suntiha Ravi (2020). "A framework for determining the fairness of outlier detection". In: *ECAI 2020*. IOS Press, pp. 2465–2472 (cit. on p. 120).

Deerwester, Scott, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman (1990). "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6, pp. 391–407 (cit. on p. 58).

Deng, Ailin and Bryan Hooi (2021). "Graph neural network-based anomaly detection in multivariate time series". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 5, pp. 4027–4035 (cit. on p. 151).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Conference of the North American Chapter of the Association for Computational Linguistics* (cit. on pp. 59, 101, 103, 119, 127).

Doddington, George (2002). "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics". In: *Conference on Human Language Technology Research* (cit. on p. 133).

Dohare, Shibhansh, Vivek Gupta, and Harish Karnick (2018). "Unsupervised semantic abstractive summarization". In: *Proceedings of ACL 2018, Student Research Workshop*, pp. 74–83 (cit. on p. 129).

Eisenstein, Jacob (2019). *Introduction to natural language processing*. Adaptive computation and machine learning. The MIT Press. 519 pp. (cit. on p. 9).

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *kdd*. Vol. 96. 34, pp. 226–231 (cit. on p. 31).

Fabbri, Alexander R, Wojciech Kryścinski, Bryan McCann, et al. (2021). "Summeval: Re-evaluating summarization evaluation". In: *Transactions of the Association for Computational Linguistics* 9, pp. 391–409 (cit. on p. 136).

Fevry, Thibault and Jason Phang (Oct. 2018). "Unsupervised Sentence Compression using Denoising Auto-Encoders". In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, pp. 413–422 (cit. on p. 128).

Fouché, Edouard, Yu Meng, Fang Guo, et al. (2020). "Mining text outliers in document directories". In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, pp. 152–161 (cit. on pp. 20, 24, 81, 90, 102).

Gao, Jing, Peng Li, Zhikui Chen, and Jianing Zhang (2020). "A survey on deep learning for multimodal data fusion". In: *Neural Computation* 32.5, pp. 829–864 (cit. on pp. 99, 100).

Gehrmann, Sebastian, Yuntian Deng, and Alexander M Rush (2018). "Bottom-up abstractive summarization". In: *arXiv preprint arXiv:1808.10792* (cit. on p. 127).

Gorokhov, Oleg, Mikhail Petrovskiy, and Igor Mashechkin (2017). "Convolutional Neural Networks for Unsupervised Anomaly Detection in Text Data". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2017*. Ed. by Hujun Yin, Yang Gao, Songcan Chen, et al. Vol. 10585. Cham: Springer International Publishing, pp. 500–507 (cit. on pp. 52, 72, 74).

Graves, Alex and Jürgen Schmidhuber (2005). "Framewise phoneme classification with bidirectional LSTM and other neural network architectures". In: *Neural networks* 18.5-6, pp. 602–610 (cit. on p. 60).

Grubbs, Frank E. (1969). "Procedures for Detecting Outlying Observations in Samples". en. In: *Technometrics* 11.1, pp. 1–21 (cit. on p. 15).

Guo, Han, Ramakanth Pasunuru, and Mohit Bansal (2018). "Soft Layer-Specific Multi-Task Summarization with Entailment and Question Generation". In: *Association for Computational Linguistics* (cit. on p. 135).

Gupta, Nikhil, Dhivya Eswaran, Neil Shah, Leman Akoglu, and Christos Faloutsos (2019). "Beyond outlier detection: Lookout for pictorial explanation". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*. Springer, pp. 122–138 (cit. on p. 120).

Guthrie, David, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks (2006). "A closer look at skip-gram modelling." In: *LREC*. Vol. 6, pp. 1222–1225 (cit. on p. 59).

Guthrie, David, Louise Guthrie, Ben Allison, and Yorick Wilks (2007). "Unsupervised anomaly detection". In: *Proceedings of the 20th international joint conference on Artifical intelligence*. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 1624–1628 (cit. on pp. 55, 100).

Hardy, Hardy, Shashi Narayan, and Andreas Vlachos (2019). "HighRES: Highlight-based Reference-less Evaluation of Summarization". In: *Association for Computational Linguistics* (cit. on p. 141).

Hastie, Trevor, Jerome Friedman, and Robert Tibshirani (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York (cit. on pp. 11, 33).

Hawkins, D. M (1980). *Identification of Outliers*. English. Dordrecht: Springer Netherlands (cit. on pp. 11, 13, 14, 17, 18, 22, 47, 52).

Hawkins, Douglas M., Dan Bradu, and Gordon V. Kass (1984). "Location of Several Outliers in Multiple-Regression Data Using Elemental Sets". In: *Technometrics* 26.3, pp. 197–208 (cit. on p. 12).

Hinton, G. E. and R. R. Salakhutdinov (2006). "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786, pp. 504–507 (cit. on p. 10).

Hochreiter, Sepp and Jurgen Schmidhuber (Dec. 1997). "Long Short-term Memory". In: *Neural computation* 9, pp. 1735–80 (cit. on p. 60).

Hodge, Victoria and Jim Austin (Oct. 2004). "A Survey of Outlier Detection Methodologies". en. In: *Artificial Intelligence Review* 22.2, pp. 85–126 (cit. on pp. 15, 17–19, 21, 22, 47, 52).

Honda, Ukyo, Tsutomu Hirao, and Masaaki Nagata (2018). "Pruning Basic Elements for Better Automatic Evaluation of Summaries". In: *Conference of the North American Chapter of the Association for Computational Linguistics)* (cit. on pp. 136, 137).

Hovy, Eduard H, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto (2006). "Automated Summarization Evaluation with Basic Elements." In: *LREC* (cit. on p. 138).

Hu, Yibo and Latifur Khan (2021). "Uncertainty-Aware Reliable Text Classification". In: *arXiv:2107.07114 [cs]* (cit. on p. 55).

Hutto, Clayton and Eric Gilbert (2014). "Vader: A parsimonious rule-based model for sentiment analysis of social media text". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 8. 1 (cit. on pp. 101, 109).

Idris, Ismaila, Ali Selamat, and Sigeru Omatu (2014). "Hybrid email spam detection model with negative selection algorithm and differential evolution". In: *Engineering Applications of Artificial Intelligence* 28, pp. 97–110 (cit. on p. 54).

Jain, Sarthak and Byron C Wallace (2019). "Attention is not explanation". In: *arXiv preprint arXiv:1902.10186* (cit. on p. 119).

Jean, Sébastien, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio (2015). "On Using Very Large Target Vocabulary for Neural Machine Translation". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (cit. on p. 125).

Jiang, Yichen and Mohit Bansal (2018). "Closed-Book Training to Improve Summarization Encoder Memory". In: *Empirical Methods in Natural Language Processing* (cit. on p. 127).

Kannan, Ramakrishnan, Hyenkyun Woo, Charu C. Aggarwal, and Haesun Park (2017). "Outlier Detection for Text Data". In: *SDM International Conference on Data Mining* 17, pp. 489–497 (cit. on pp. 52, 55, 68, 69, 74, 78, 81, 100, 103, 111).

Karim, Asif, Sami Azam, Bharanidharan Shanmugam, Krishnan Kannoorpatti, and Mamoun Alazab (2019). "A Comprehensive Survey for Intelligent Spam Email Detection". In: *IEEE Access* 7, pp. 168261–168295 (cit. on pp. 54, 65).

Kieu, Tung, Bin Yang, Chenjuan Guo, and Christian S. Jensen (July 2019). "Outlier Detection for Time Series with Recurrent Autoencoder Ensembles". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pp. 2725–2732 (cit. on p. 103).

Kim, Been, Rajiv Khanna, and Oluwasanmi O Koyejo (2016). "Examples are not enough, learn to criticize! criticism for interpretability". In: *Advances in neural information processing systems* 29 (cit. on p. 119).

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (cit. on p. 44).

Koppel, Moshe and Shachar Seidman (Oct. 2013). "Automatically Identifying Pseudepigraphic Texts". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2013. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1449–1454 (cit. on pp. 69, 74).

Koren, Yehuda, Robert Bell, and Chris Volinsky (2009). "Matrix Factorization Techniques for Recommender Systems". In: *Computer* 42.8, pp. 30–37 (cit. on p. 33).

Kowsari, Kamran, Donald E Brown, Mojtaba Heidarysafa, et al. (2017). "Hdltex: Hierarchical deep learning for text classification". In: *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, pp. 364–371 (cit. on p. 79).

Kriegel, Hans-Peter, Peer Kröger, Erich Schubert, and Arthur Zimek (2009a). "LoOP: local outlier probabilities". In: *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*. Proceeding of the 18th ACM conference. Hong Kong, China: ACM Press, p. 1649 (cit. on p. 31).

— (2009b). "Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu-Bao Ho. Vol. 5476. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 831–838 (cit. on pp. 35, 38).

Kriegel, Hans-Peter, Matthias S hubert, and Arthur Zimek (2008). "Angle-based outlier detection in high-dimensional data". In: *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08*. the 14th ACM SIGKDD international conference. Las Vegas, Nevada, USA: ACM Press, p. 444 (cit. on p. 37).

Kryścinski, Wojciech, Bryan McCann, Caiming Xiong, and Richard Socher (2019). "Evaluating the factual consistency of abstractive text summarization". In: *arXiv preprint arXiv:1910.12840* (cit. on p. 136).

Kryściński, Wojciech, Romain Paulus, Caiming Xiong, and Richard Socher (2018). "Improving Abstraction in Text Summarization". In: *arXiv:1808.07913 [cs]* (cit. on p. 55).

Lai, Chieh-Hsin, Dongmian Zou, and Gilad Lerman (2020). "Robust Subspace Recovery Layer for Unsupervised Anomaly Detection". In: *ICLR International Conference on Learning Representations* (cit. on pp. 44, 52, 55, 71–74, 78, 81, 88, 100, 103, 104, 107, 110, 111, 152).

Lample, Guillaume, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato (2017). "Unsupervised machine translation using monolingual corpora only". In: *In Proceedings of the International Conference on Learning Representations* (cit. on p. 128).

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, et al. (2019). "Albert: A lite bert for self-supervised learning of language representations". In: *arXiv preprint arXiv:1909.11942* (cit. on p. 60).

Landauer, Thomas K, Peter W Foltz, and Darrell Laham (1998). "An introduction to latent semantic analysis". In: *Discourse processes* 25.2-3, pp. 259–284 (cit. on p. 58).

Laorden, Carlos, Xabier Ugarte-Pedrero, Igor Santos, et al. (2014). "Study on the effectiveness of anomaly detection for spam filtering". In: *Information Sciences* 277, pp. 421–444 (cit. on p. 54).

Lazhar, Farek (2019). "Fuzzy clustering-based semi-supervised approach for outlier detection in big text data". In: *Progress in Artificial Intelligence* 8.1, pp. 123–132 (cit. on p. 55).

Le, Quoc V. (2013). "Building high-level features using large scale unsupervised learning". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. ICASSP 2013 - 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 8595–8598 (cit. on p. 10).

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *Nature* 521.7553, pp. 436–444 (cit. on p. 32).

Lee, Daniel and H. Sebastian Seung (2000). "Algorithms for Non-negative Matrix Factorization". In: *Advances in Neural Information Processing Systems*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. MIT Press (cit. on p. 33).

Lerman, Gilad and Tyler Maunu (2018). "An overview of robust subspace recovery". In: *Proceedings of the IEEE* 106.8, pp. 1380–1410 (cit. on pp. 44, 103).

Lin, Chin-Yew (July 2004). "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics (cit. on p. 132).

Liu, Bing (2012). "Sentiment Analysis and Opinion Mining". In: *Synthesis Lectures on Human Language Technologies* 5.1, pp. 1–167 (cit. on p. 99).

Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou (2008). "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422 (cit. on pp. 37, 38, 73, 81, 111).

Liu, Huawen, Xuelong Li, Jiuyong Li, and Shichao Zhang (2018). "Efficient Outlier Detection for High-Dimensional Data". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48.12, pp. 2451–2461 (cit. on p. 98).

Liu, Ninghao, Donghwa Shin, and Xia Hu (2017). "Contextual outlier interpretation". In: *arXiv preprint arXiv:1711.10589* (cit. on p. 120).

Liu, Yang and Mirella Lapata (2019). "Text Summarization with Pretrained Encoders". In: *Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing* (cit. on p. 140).

Liu, Yezheng, Zhe Li, Chong Zhou, et al. (2019). "Generative Adversarial Active Learning for Unsupervised Outlier Detection". In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1 (cit. on pp. 44, 144).

Liu, Yinhan, Myle Ott, Naman Goyal, et al. (2019). "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (cit. on pp. 60, 80, 103, 106).

Lu, Jie, Anjin Liu, Fan Dong, et al. (2018). "Learning under Concept Drift: A Review". In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1 (cit. on p. 55).

Lundberg, Scott M and Su-In Lee (2017). "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (cit. on pp. 119, 120).

Ma, Xiaoxiao, Jia Wu, Shan Xue, et al. (2021). "A comprehensive survey on graph anomaly detection with deep learning". In: *IEEE Transactions on Knowledge and Data Engineering* (cit. on p. 151).

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, et al. (2011). "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 142–150 (cit. on p. 79).

Macha, Meghanath and Leman Akoglu (2018). "Explaining anomalies in groups with characterizing subspace rules". In: *Data Mining and Knowledge Discovery* 32, pp. 1444–1480 (cit. on p. 120).

Mahapatra, Amogh, Nisheeth Srivastava, and Jaideep Srivastava (2012). "Contextual Anomaly Detection in Text Data". In: *Algorithms* 5.4, pp. 469–489 (cit. on pp. 55, 78, 100, 102).

Manevitz, Larry M. and Malik Yousef (2001). "One-Class SVMs for Document Classification". In: *Journal of Machine Learning Research* 2, pp. 139–154 (cit. on pp. 55, 71, 74, 81, 98, 103).

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to information retrieval*. New York: Cambridge University Press (cit. on p. 64).

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of statistical natural language processing*. Cambridge, Mass: MIT Press. 680 pp. (cit. on p. 64).

Manolache, Andrei, Florin Brad, and Elena Burceanu (2021). "DATE: Detecting Anomalies in Text via Self-Supervision of Transformers". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 267–277 (cit. on pp. 24, 73, 74, 82, 86, 103, 106, 119).

Markou, Markos and Sameer Singh (2003). "Novelty detection: a review—part 1: statistical approaches". In: *Signal Processing* 83.12, pp. 2481–2497 (cit. on p. 12).

Marques, Henrique O., Ricardo J. G. B. Campello, Jörg Sander, and Arthur Zimek (2020). "Internal Evaluation of Unsupervised Outlier Detection". In: *ACM Transactions on Knowledge Discovery from Data* 14.4, pp. 1–42 (cit. on p. 152).

Maunu, Tyler, Teng Zhang, and Gilad Lerman (2019). "A well-tempered landscape for non-convex robust subspace recovery". In: *Journal of Machine Learning Research* 20.37 (cit. on p. 105).

Medhat, Walaa, Ahmed Hassan, and Hoda Korashy (2014). "Sentiment analysis algorithms and applications: A survey". In: *Ain Shams Engineering Journal* 5.4, pp. 1093–1113 (cit. on pp. 99, 101).

Mei, Mei, Xinyu Guo, Belinda C. Williams, et al. (July 2018). "Using Semantic Clustering And Autoencoders For Detecting Novelty In Corpora Of Short Texts". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. 2018 International Joint Conference on Neural Networks (IJCNN). Rio de Janeiro: IEEE, pp. 1–8 (cit. on pp. 72, 74).

Meng, Tong, Xuyang Jing, Zheng Yan, and Witold Pedrycz (2020). "A survey on machine learning for data fusion". In: *Information Fusion* 57, pp. 115–129 (cit. on p. 99).

Mi, Haitao, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah (2016). "Coverage Embedding Models for Neural Machine Translation". In: *Empirical Methods in Natural Language Processing* (cit. on p. 126).

Miao, Yishu and Phil Blunsom (Nov. 2016). "Language as a Latent Variable: Discrete Generative Models for Sentence Compression". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 319–328 (cit. on p. 128).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc. (cit. on pp. 59, 125).

Mohotti, Wathsala Anupama and Richi Nayak (Oct. 6, 2020). "Efficient Outlier Detection in Text Corpus Using Rare Frequency and Ranking". In: *ACM Transactions on Knowledge Discovery from Data* 14.6, pp. 1–30 (cit. on pp. 69, 74).

Muhr, Markus, Mario Zechner, and Roman Kern (2009). "External and Intrinsic Plagiarism Detection Using Vector Space Models". In: *CEUR Workshop Proceedings* 502 (cit. on p. 55).

Nadkarni, Prakash M, Lucila Ohno-Machado, and Wendy W Chapman (2011). "Natural language processing: an introduction". In: *Journal of the American Medical Informatics Association* 18.5, pp. 544–551 (cit. on p. 64).

Nallapati, Ramesh, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang (2016). "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: *Conference on Computational Natural Language Learning* (cit. on pp. 125, 126, 128, 129, 138).

Narayan, Shashi, Shay B Cohen, and Mirella Lapata (2018). "Ranking Sentences for Extractive Summarization with Reinforcement Learning". In: *North American Chapter of the Association for Computational Linguistics* (cit. on pp. 129, 140).

Nenkova, Ani and Rebecca Passonneau (2004). "Evaluating content selection in summarization: The pyramid method". In: *Conference of the North American chapter of the Association for Computational Linguistics* (cit. on p. 135).

OpenAI (2023). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL] (cit. on p. 61).

Pantin, Jérémie, Christophe Marsala, and Marie-Jeanne Lesot (2022). "Analyse de Données Aberrantes pour le Texte: Taxonomie et Étude Expérimentale". In: *Extraction et Gestion des Connaissances-EGC'2022*, pp. 15–26 (cit. on pp. 76, 148).

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "BLEU: a method for automatic evaluation of machine translation". In: *Association for Computational Linguistics* (cit. on p. 130).

Parker, A. and J.O. Hamblen (1989). "Computer algorithms for plagiarism detection". In: *IEEE Transactions on Education* 32.2, pp. 94–99 (cit. on p. 55).

Pasunuru, Ramakanth and Mohit Bansal (2018). "Multi-Reward Reinforced Summarization with Saliency and Entailment". In: *Conference of the North American Chapter of the Association for Computational Linguistics* (cit. on p. 127).

Paulus, Romain, Caiming Xiong, and Richard Socher (2017). "A deep reinforced model for abstractive summarization". In: *arXiv preprint arXiv:1705.04304* (cit. on p. 127).

Pedregosa, F., G. Varoquaux, A. Gramfort, et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830 (cit. on p. 111).

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp. 1532–1543 (cit. on pp. 59, 103).

Peters, Matthew E., Mark Neumann, Mohit Iyyer, et al. (2018). "Deep contextualized word representations". In: *Proceedings of NAACL* (cit. on p. 60).

Pimentel, Marco A.F., David A. Clifton, Lei Clifton, and Lionel Tarassenko (2014). "A review of novelty detection". In: *Signal Processing* 99, pp. 215–249 (cit. on p. 12).

Potthast, Martin, Andreas Eiselt, Luis Alberto Barrón Cedeño, Benno Stein, and Paolo Rosso (2011). "Overview of the 3rd International Competition on Plagiarism Detection". In: *CEUR Workshop Proceedings*. Vol. 1177. CEUR Workshop Proceedings (cit. on p. 55).

Prastawa, Marcel, Elizabeth Bullitt, Sean Ho, and Guido Gerig (2004). "A brain tumor segmentation framework based on outlier detection". In: *Medical image analysis* 8.3, pp. 275–283 (cit. on p. 11).

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. (2018). "Improving language understanding by generative pre-training". In: (cit. on p. 60).

Radford, Alec, Jeffrey Wu, Rewon Child, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9 (cit. on p. 60).

Raffel, Colin, Noam Shazeer, Adam Roberts, et al. (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *The Journal of Machine Learning Research* 21.1, pp. 5485–5551 (cit. on p. 61).

Rahmani, Mostafa and George K. Atia (2017). "Randomized Robust Subspace Recovery and Outlier Detection for High Dimensional Data Matrices". In: *IEEE Transactions on Signal Processing* 65.6, pp. 1580–1594 (cit. on p. 103).

Ramadhani, Adyan Marendra and Hong Soon Goo (2017). "Twitter sentiment analysis using deep learning methods". In: *International Annual Engineering Seminar*. IEEE, pp. 1–4 (cit. on p. 101).

Ramaswamy, Sridhar, Rajeev Rastogi, and Kyuseok Shim (2000). "Efficient algorithms for mining outliers from large data sets". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*. the 2000 ACM SIGMOD international conference. Dallas, Texas, United States: ACM Press, pp. 427–438 (cit. on pp. 29, 30, 69, 81).

Reimers, Nils and Iryna Gurevych (Nov. 2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics (cit. on pp. 80, 106).

Ren, Pengjie, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou (2016). "A Redundancy-Aware Sentence Regression Framework for Extractive Summarization". In: *Conference on Computational Linguistics* (cit. on p. 141).

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). """ Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144 (cit. on pp. 119, 120).

Rosenblatt, Frank (1958). "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6, pp. 386–408 (cit. on p. 41).

Rousseeuw, Peter J and Mia Hubert (2011). "Robust statistics for outlier detection". In: *Wiley interdisciplinary reviews: Data mining and knowledge discovery* 1.1, pp. 73–79 (cit. on pp. 11, 27).

Rousseeuw, Peter J. and Annick M. Leroy (1987). *Robust Regression and Outlier Detection*. 1st ed. Wiley Series in Probability and Statistics. Wiley (cit. on p. 12).

Roweis, Sam T and Lawrence K Saul (2000). "Nonlinear dimensionality reduction by locally linear embedding". In: *science* 290.5500, pp. 2323–2326 (cit. on p. 105).

Ruff, Lukas, Jacob R. Kauffmann, Robert A. Vandermeulen, et al. (May 2021). "A Unifying Review of Deep and Shallow Anomaly Detection". In: *Proceedings of the IEEE* 109.5, pp. 756–795 (cit. on pp. 16, 17, 19–22, 24, 47, 48, 52, 55, 81, 98).

Ruff, Lukas, Robert Vandermeulen, Nico Goernitz, et al. (2018). "Deep one-class classification". In: *International conference on machine learning*. PMLR, pp. 4393–4402 (cit. on pp. 43, 81).

Ruff, Lukas, Yury Zemlyanskiy, Robert Vandermeulen, Thomas Schnake, and Marius Kloft (2019). "Self-Attentive, Multi-Context One-Class Classification for Unsupervised Anomaly Detection on Text". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4061–4071 (cit. on pp. 52, 55, 73, 74, 78, 81, 82, 86, 97, 103, 106, 119).

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1985). "Learning internal representations by error propagation". In: (cit. on p. 128).

Rush, Alexander M, Sumit Chopra, and Jason Weston (2015). "A Neural Attention Model for Abstractive Sentence Summarization". In: *Empirical Methods in Natural Language Processing* (cit. on pp. 129, 138, 143, 145).

Ruz, Gonzalo A, Pablo A Henríquez, and Aldo Mascareno (2020). "Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers". In: *Future Generation Computer Systems* 106, pp. 92–104 (cit. on p. 99).

Sathe, Saket and Charu C. Aggarwal (2016). "Subspace Outlier Detection in Linear Time with Randomized Hashing". In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 2016 IEEE 16th International Conference on Data Mining (ICDM). Barcelona, Spain: IEEE, pp. 459–468 (cit. on p. 26).

Savage, David, Xiuzhen Zhang, Xinghuo Yu, Pauline Chou, and Qingmai Wang (2014). "Anomaly detection in online social networks". In: *Social Networks* 39, pp. 62–70 (cit. on p. 101).

Schölkopf, Bernhard, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson (July 1, 2001). "Estimating the Support of a High-Dimensional Distribution". In: *Neural Computation* 13.7, pp. 1443–1471 (cit. on pp. 103, 111).

Schölkopf, Bernhard, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt (2000). "Support Vector Method for Novelty Detection". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press (cit. on p. 39).

Schubert, Erich, Arthur Zimek, and Hans-Peter Kriegel (Jan. 2014). "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection". In: *Data Mining and Knowledge Discovery* 28.1, pp. 190–237 (cit. on pp. 16, 31).

Schuster, Mike and Kuldip K Paliwal (1997). "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11, pp. 2673–2681 (cit. on p. 60).

See, Abigail, Peter J. Liu, and Christopher D. Manning (2017). "Get To The Point: Summarization with Pointer-Generator Networks". In: *Association for Computational Linguistics* (cit. on pp. 126, 127, 135).

ShafieiBavani, Elaheh, Mohammad Ebrahimi, Raymond Wong, and Fang Chen (2018a). "A Graph-theoretic Summary Evaluation for ROUGE". In: *Conference on Empirical Methods in Natural Language Processing* (cit. on p. 138).

— (2018b). "Summarization Evaluation in the Absence of Human Model Summaries Using the Compositionality of Word Embeddings". In: *Conference on Computational Linguistics* (cit. on pp. 136, 138).

Shanmugam, Mahalakshmi, Aayushi Agawane, Anchal Tiwari, and Rugved V Deolekar (2020). "Twitter Sentiment Analysis using Novelty Detection". In: *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*. Tirunelveli, India: IEEE, pp. 1258–1263 (cit. on p. 55).

Shekhar, Shubhranshu, Neil Shah, and Leman Akoglu (2021). "Fairod: Fairness-aware outlier detection". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 210–220 (cit. on p. 120).

Shravan Kumar, B. and Vadlamani Ravi (2017). "One-Class Text Document Classification with OCSVM and LSI". In: *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Ed. by Subhransu Sekhar Dash, K. Vijayakumar, Bijaya Ketan Panigrahi, and Swagatam Das. Vol. 517. Singapore: Springer Singapore, pp. 597–606 (cit. on pp. 71, 74).

Shyu, Mei-Ling, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang (2003). *A novel anomaly detection scheme based on principal component classifier*. Tech. rep. MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL and COMPUTER ENGINEERING (cit. on pp. 33, 34, 81).

Siddiqui, Md Amran, Alan Fern, Thomas G Dietterich, and Weng-Keen Wong (2019). "Sequential feature explanations for anomaly detection". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13.1, pp. 1–22 (cit. on p. 120).

Smits, Grégory, Marie-Jeanne Lesot, Véronne Yepmo Tchaghe, and Olivier Pivert (2022). "PANDA: Human-in-the-Loop Anomaly Detection and Explanation". In: *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, pp. 720–732 (cit. on p. 120).

Socher, Richard, Alex Perelygin, Jean Wu, et al. (Oct. 2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1631–1642 (cit. on p. 79).

Soleymani, Mohammad, David Garcia, Brendan Jou, et al. (2017). "A survey of multimodal sentiment analysis". In: *Image and Vision Computing* 65, pp. 3–14 (cit. on p. 99).

Spirin, Nikita and Jiawei Han (2012). "Survey on web spam detection: principles and algorithms". In: *ACM SIGKDD Explorations Newsletter* 13.2, pp. 50–64 (cit. on p. 54).

Srivastava, A.N. and B. Zane-Ulman (2005). "Discovering recurring anomalies in text reports regarding complex space systems". In: *2005 IEEE Aerospace Conference*. 2005 IEEE Aerospace Conference. Big Sky, MT, USA: IEEE, pp. 3853–3862 (cit. on pp. 70, 71, 74).

Stein, Benno and Sven Meyer zu Eissen (2007). "Intrinsic plagiarism analysis with meta learning". In: *PAN* (cit. on p. 55).

Stewart, Gilbert W (1993). "On the early history of the singular value decomposition". In: *SIAM review* 35.4, pp. 551–566 (cit. on p. 58).

Sutskever, Ilya, James Martens, and Geoffrey E Hinton (2011). "Generating text with recurrent neural networks". In: *Proceedings of the 28th international conference on machine learning*, pp. 1017–1024 (cit. on p. 59).

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems* (cit. on p. 125).

Talmy, Leonard (2000). *Toward a cognitive semantics*. Vol. 2. MIT press (cit. on p. 66).

Tang, Guanting, James Bailey, Jian Pei, and Guozhu Dong (2013). "Mining multidimensional contextual outliers from categorical relational data". In: *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, pp. 1–4 (cit. on p. 120).

Tax, David Martinus Johannes (2001). "One-class classification: concept-learning in the absence of counter-examples". English. PhD thesis. S.l.: s.n. (cit. on p. 8).

Tax, David MJ and Robert PW Duin (2004). "Support vector data description". In: *Machine learning* 54.1, pp. 45–66 (cit. on p. 43).

Tishby, Naftali, Fernando Pereira, and William Bialek (July 2001). "The Information Bottleneck Method". In: vol. 49 (cit. on pp. 128, 142).

Toutanova, Kristina, Francine Chen, Kris Popat, and Thomas Hofmann (2001). "Text classification in a hierarchical mixture model for small training sets". In: *Proceedings of the tenth international conference on Information and knowledge management*, pp. 105–113 (cit. on pp. 66, 76, 79, 80).

Tran Manh Thang and Juntae Kim (Apr. 2011). "The Anomaly Detection by Using DBSCAN Clustering with Multiple Parameters". In: *2011 International Conference on Information Science and Applications*. 2011 International Conference on Information Science and Applications (ICISA 2011). Jeju Island: IEEE, pp. 1–5 (cit. on pp. 70, 74).

Tu, Zhaopeng, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li (2016a). "Modeling Coverage for Neural Machine Translation". In: *Association for Computational Linguistics* (cit. on p. 126).

— (2016b). "Modeling coverage for neural machine translation". In: *arXiv preprint arXiv:1601.04811* (cit. on p. 126).

Urologin, Siddhaling (2018). "Sentiment analysis, visualization and classification of summarized news articles: a novel approach". In: *International Journal of Advanced Computer Science and Applications* 9.8, pp. 616–625 (cit. on p. 101).

Van Halteren, Hans and Simone Teufel (2003). "Examining the consensus between human summaries: initial experiments with factoid analysis". In: *Conference of the North American chapter of the Association for Computational Linguistics* (cit. on p. 136).

Vapnik, Vladimir N (1999). "An overview of statistical learning theory". In: *IEEE transactions on neural networks* (cit. on p. 137).

Vashishtha, Srishti and Seba Susan (2019). "Fuzzy rule based unsupervised sentiment analysis from social media posts". In: *Expert Systems with Applications* 138, p. 112834 (cit. on p. 99).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008 (cit. on pp. 60, 127).

Vig, Jesse and Yonatan Belinkov (2019). "Analyzing the structure of attention in a transformer language model". In: *arXiv preprint arXiv:1906.04284* (cit. on p. 119).

Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly (2015). "Pointer networks". In: *Advances in Neural Information Processing Systems* (cit. on p. 126).

Walkowiak, Tomasz, Szymon Datko, and Henryk Maciejewski (2020). "Utilizing Local Outlier Factor for Open-Set Classification in High-Dimensional Data - Case Study Applied for Text Documents". In: *Intelligent Systems and Applications*. Ed. by Yaxin Bi, Rahul Bhatia, and Supriya Kapoor. Vol. 1037. Cham: Springer International Publishing, pp. 408–418 (cit. on pp. 71, 74).

Wang, Fei, Robert J. Ross, and John D. Kelleher (2018). "Exploring Online Novelty Detection Using First Story Detection Models". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Ed. by Hujun Yin, David Camacho, Paulo Novais, and Antonio J. Tallón-Ballesteros. Vol. 11314. Cham: Springer International Publishing, pp. 107–116 (cit. on pp. 55, 69, 74).

Wang, Siqi, Yijie Zeng, Xinwang Liu, et al. (2019). "Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network". In: *Advances in neural information processing systems* 32 (cit. on p. 73).

Wang, Yaushian and Hung-Yi Lee (2018). "Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4187–4195 (cit. on pp. 143, 145).

Wawre, Suchita V and Sachin N Deshmukh (2016). "Sentiment classification using machine learning techniques". In: *International Journal of Science and Research (IJSR)* 5.4, pp. 819–821 (cit. on p. 101).

West, Peter, Ari Holtzman, Jan Buys, and Yejin Choi (2019). "BottleSum: Unsupervised and Self-supervised Sentence Summarization using the Information Bottleneck Principle". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3743–3752 (cit. on p. 128).

Wu, Tingmin, Sheng Wen, Yang Xiang, and Wanlei Zhou (2018). "Twitter spam detection: Survey of new approaches and comparative study". In: *Computers & Security* 76, pp. 265–284 (cit. on p. 55).

Xu, Hongzuo, Yijie Wang, Songlei Jian, et al. (2021). "Beyond outlier detection: Outlier interpretation by attention-guided triplet deviation network". In: *Proceedings of the Web Conference 2021*, pp. 1328–1339 (cit. on p. 120).

Yang, Zhilin, Zihang Dai, Yiming Yang, et al. (2019). "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems* 32 (cit. on p. 60).

Yepmo, Véronne, Grégory Smits, and Olivier Pivert (2022). "Anomaly explanation: A review". In: *Data & Knowledge Engineering* 137, p. 101946 (cit. on p. 120).

You, Lan, Qingxi Peng, Zenggang Xiong, et al. (2020). "Integrating aspect analysis and local outlier factor for intelligent review spam detection". In: *Future Generation Computer Systems* 102, pp. 163–172 (cit. on p. 54).

Yue, Lin, Weitong Chen, Xue Li, Wanli Zuo, and Minghao Yin (2019). "A survey of sentiment analysis in social media". In: *Knowledge and Information Systems* 60.2, pp. 617–663 (cit. on p. 99).

Zhang, Ji (Feb. 2013). "Advancements of Outlier Detection: A Survey". en. In: *ICST Transactions on Scalable Information Systems* 13.1 (cit. on pp. 15, 17, 20–22, 25, 26, 47, 52).

Zhang, Ji, Qigang Gao, and Hai Wang (2006). "A Novel Method for Detecting Outlying Subspaces in High-dimensional Databases Using Genetic Algorithm". In: *Sixth International Conference on Data Mining (ICDM'06)*. Sixth International Conference on Data Mining (ICDM'06). Hong Kong, China: IEEE, pp. 731–740 (cit. on p. 26).

Zhang, Ji and Hai Wang (2006). "Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance". In: *Knowledge and Information Systems* 10.3, pp. 333–355 (cit. on p. 26).

Zhang, Wenbin and Steven Skiena (2010). "Trading strategies to exploit blog and news sentiment". In: *International AAAI Conference on Weblogs and Social Media*. Vol. 4. 1, pp. 375–378 (cit. on p. 99).

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2015). "Character-level convolutional networks for text classification". In: *Advances in neural information processing systems* 28 (cit. on p. 79).

Zhang, Zhihao, Xinnian Liang, Yuan Zuo, and Zhoujun Li (2023). "Unsupervised abstractive summarization via sentence rewriting". In: *Computer Speech & Language* 78, p. 101467 (cit. on p. 129).

Zhao, Yue, Zain Nasrullah, Maciej K Hryniewicki, and Zheng Li (2019). "LSCP: Locally selective combination in parallel outlier ensembles". In: *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, pp. 585–593 (cit. on pp. 100, 102, 103).

Zhao, Yue, Zain Nasrullah, and Zheng Li (2019). "PyOD: A Python Toolbox for Scalable Outlier Detection". In: *Journal of Machine Learning Research* 20.96, pp. 1–7 (cit. on pp. 24, 81, 111).

Zhou, Chong and Randy C. Paffenroth (2017). "Anomaly Detection with Robust Deep Autoencoders". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. New York, NY, USA: Association for Computing Machinery, pp. 665–674 (cit. on p. 103).

Zhou, Qingyu, Nan Yang, Furu Wei, Shaohan Huang, et al. (2018). "Neural Document Summarization by Jointly Learning to Score and Select Sentences". In: *Association for Computational Linguistics* (cit. on pp. 127, 139–141).

Zhou, Qingyu, Nan Yang, Furu Wei, and Ming Zhou (July 2017). "Selective Encoding for Abstractive Sentence Summarization". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1095–1104 (cit. on p. 124).

Zhuang, Haojie, Wei Emma Zhang, Jian Yang, et al. (2022). "Learning From the Source Document: Unsupervised Abstractive Summarization". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 4194–4205 (cit. on pp. 129, 143, 145).

Zimek, Arthur, Ricardo JGB Campello, and Jörg Sander (2014). "Ensembles for unsupervised outlier detection: challenges and research questions a position paper". In: *Acm Sigkdd Explorations Newsletter* 15.1, pp. 11–22 (cit. on pp. 100, 102, 103).

Zimek, Arthur, Erich Schubert, and Hans-Peter Kriegel (2012). "A survey on unsupervised outlier detection in high-dimensional numerical data". In: *Statistical Analysis and Data Mining* 5.5, pp. 363–387 (cit. on p. 12).