Context-based detection of sensitive data in structured datasets

Kužina, Vjeko

Sveučilište u Zagrebu

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Vjeko Kužina

CONTEXT-BASED DETECTION OF SENSITIVE DATA IN STRUCTURED DATASETS

DOCTORAL THESIS

Zagreb, 2024

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Vjeko Kužina

CONTEXT-BASED DETECTION OF SENSITIVE DATA IN STRUCTURED DATASETS

DOCTORAL THESIS

Supervisor: Associate Professor Alan Jović, PhD

Zagreb, 2024

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Vjeko Kužina

OTKRIVANJE OSJETLJIVIH PODATAKA U STRUKTURIRANIM SKUPOVIMA PODATAKA ZASNOVANO NA KONTEKSTU

DOKTORSKI RAD

Mentor: izv. prof. dr. sc. Alan Jović

Zagreb, 2024

The doctoral thesis was prepared at the University of Zagreb, Faculty of Electrical Engineering and Computer Science, at the Department of Electronics, Microelectronics, Computer and Intelligent Systems.

Supervisor: Associate Professor Alan Jović

The doctoral thesis has: 73 pages

Doctoral thesis nr.: _____

About the Supervisor

Alan Jović was born in Zagreb in 1982. He received his Dipl. Ing. and PhD degrees in computer science from the University of Zagreb Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia, in 2006 and 2012, respectively. From September 2006 till March 2007, he worked as expert associate at the Ruđer Bošković Institute. Since April 2007, he has been working at the Department of Electronics, Microelectronics, Computer and Intelligent Systems at FER. In June 2016, he became assistant professor, and in April 2020, associate professor at FER. He is the author of 26 published papers in A-category scientific journals, 3 scientific book chapters, 2 papers in other scientific journals, and 44 papers in proceedings of international conferences. His works were cited more than 1000 times according to the Web of Science database (h-index 14). He is Editor-in-Chief of the scientific journal "CIT. Journal of Computing and Information Technology". He was a program committee member for 15 international conferences and held several invited lectures. He is a program committee member of the international ICT convention MIPRO as well as conference chair of MIPRO's "Artificial Intelligence Systems" (AIS) conference. He received several national and international awards and acknowledgments for his scientific work. He was principal investigator on four scientific and industrial national projects and collaborator on seven national and international projects. His professional areas of interest include data mining, application of computing in medicine, and knowledge representation in computer systems. He is a member of IEEE, EMBS and MIPRO organizations.

O mentoru

Alan Jović rođen je u Zagrebu 1982. godine. Diplomirao je i doktorirao računarstvo na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva (FER), Zagreb, Hrvatska, 2006. odnosno 2012. godine. Od rujna 2006. do ožujka 2007. radio je kao stručni suradnik na Institutu Ruđer Bošković. Od travnja 2007. radi na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave na FER-u. U lipnju 2016. postaje docent, a u travnju 2020. izvanredni profesor na FER-u. Autor je 26 radova objavljenih u znanstvenim časopisima A-kategorije, 3 poglavlja u znanstvenim knjigama, 2 rada u drugim znanstvenim časopisima i 44 rada u zbornicima međunarodnih konferencija. Njegovi radovi citirani su više od 1000 puta prema bazi podataka Web of Science (h-indeks 14). Glavni je urednik znanstvenog časopisa "CIT. Journal of Computing and Information Technology". Bio je član programskog odbora na 15 međunarodnih konferencija i održao je nekoliko pozvanih predavanja. Član je programskog odbora međunarodne ICT konferencije MIPRO, kao i voditelj MIPRO-ovog savjetovanja "Sustavi umjetne inteligencije" (AIS). Za svoj znanstveni rad dobio je nekoliko nacionalnih i međunarodnih nagrada i priznanja. Bio je glavni istraživač na četiri znanstvena i industrijska nacionalna projekta i suradnik na sedam nacionalnih i međunarodnih projekata. Njegova profesionalna područja od interesa uključuju dubinsku analizu podataka, primjenu računarstva u medicini i predstavljanje znanja u računalnim sustavima. Član je organizacija IEEE, EMBS i MIPRO.

Abstract

The digital landscape's growth has underscored the importance of addressing data privacy and security challenges. One such challenge, the detection of sensitive data in structured datasets, has become increasingly relevant due to regulatory frameworks like the GDPR in the EU. While the topic of sensitive data detection in unstructured data has been extensively researched, the problem of sensitive data detection in structured data has not been adequately addressed in related work. This research seeks to fill this gap by leveraging techniques utilized in similar problems and establishing a comparative analysis to determine the most appropriate approach. An extensive overview of methods in this field has been provided, detailing the progression of methodologies and examining existing approaches. Central to this research is the development of a novel method for structured sensitive data detection. This method integrates an active context-based approach with traditional rule-based heuristics to classify database table columns into specific labels. The active context approach leverages surrounding information, enhancing the classification process's accuracy. This method's design ensures that features of the input are preserved for contextual embedding, which is crucial for addressing the intricacies of natural language within table cells. A novel dataset has been introduced, specifically tailored for the problem of sensitive data detection in structured datasets. Alternative approaches that modify the input methods or the architecture of the proposed method have been explored. These alternatives test additional hypotheses and further investigate the problem, broadening the research scope. The motivations behind these variations and their potential advantages are discussed in detail. Experiments with the proposed method have been conducted with great success on the newly created dataset as well as on datasets from related scientific literature.

Keywords: Machine Learning, Artificial Intelligence, Structured data, Sensitive data, Natural Language Processing

Sažetak

Doktorska disertacija strukturirana je u sedam poglavlja: uvod, srodna literatura, specifikacija problema, metodologija, eksperimenti te zaključci, pružajući jasan put kroz provedeno istraživanje i njegove implikacije za budući rad u ovom području.

Prvo poglavlje uvodi sve veći izazov detekcije osjetljivih podataka zbog masovnog povećanja podataka na internetu. Ovaj porast digitalnih informacija, posebice u posljednjem desetljeću, doveo je do značajnih napredaka u različitim područjima kao što su zdravstvo i financije. Međutim, to je također povećalo zabrinutost za privatnost i sigurnost podataka, najviše u pogledu osjetljivih informacija kao što su osobno identificirajuće informacije (engl. *Personally Identifiable Information*, PII). Razjašnjavaju se razlike između osobnih podataka, osjetljivih podataka i PII, te se ističe potreba za zaštitom osjetljivih informacija od neovlaštenog pristupa. Također, ističe se globalni zakonodavni odgovor na ove probleme privatnosti.

Fokusirajući se na detekciju strukturiranih podataka, u prvom poglavlju ustanovljava se ovo područje kao nedovoljno istraženo u usporedbi s područjem detekcije u nestrukturiranim podacima. Predlaže se nova metoda koja koristi mješavinu modela strojnog učenja temeljenih na aktivnom kontekstu kao i pristupe zasnovane na pravilima kako bi poboljšala točnost detekcije u strukturiranim podacima. Opisuju se doprinosi, uključujući novu metodu detekcije, referentni skup podataka za detekciju strukturiranih podataka, istraživanje alternativnih pristupa i sveobuhvatni eksperimenti za vrednovanje predloženog rješenja.

Drugo poglavlje konstatira potrebu za automatiziranim metodama za pronalazak i zaštitu osjetljivih podataka. Ta potreba je porasla zbog stalnog rasta količine digitalnih podataka, dok se pravila o čuvanju tih podataka neprestano mijenjaju. Ručna provjera svih tih podataka više nije izvediva. Veliki dio pronalaženja osjetljivih podataka u strukturiranim podacima (kao što su tablice baza podataka sa stupcima i retcima) je utvrđivanje koji stupci sadrže osjetljive informacije te, ako ih sadrže, koja vrsta osjetljivih podataka se u njima nalazi. To je slično širem zadatku nazvanom semantičko označavanje stupaca, gdje je cilj sortirati podatke u općenite kategorije (ne samo u osjetljive podatke). Budući da je pronalaženje osjetljivih podataka u strukturiranim podacima novije područje istraživanja, ovo poglavlje također istražuje postojeće pristupe rješavanju problema semantičkog označavanja stupaca kako bi se pronašli načini za poboljšanje metode predložene u ovoj disertaciji.

U istraživanju rješenja za detekciju osjetljivih podataka, drugo poglavlje pregledava različite metode, dijeleći ih na dvije glavne kategorije: metode zasnovane na pravilima i metode zasnovane na strojnom učenju. Svaki od ovih pristupa ima svoj skup prednosti i izazova kada se primjenjuju na zadatak ustanovljavanja osjetljivih podataka unutar strukturiranih skupova podataka. Metode zasnovane na pravilima oslanjaju se na unaprijed definiran skup pravila ili uzoraka za identifikaciju osjetljivih podataka. Ta pravila su obično razvijena na temelju stručnog

znanja i vrlo su specifična u tome što čini osjetljive informacije. Na primjer, uzorci poput brojeva socijalnog osiguranja ili e-mail adresa mogu se izravno identificirati kroz specifične, dobro definirane formate. Iako su metode zasnovane na pravilima jednostavne i jednostavno ih je interpretirati, glavno ograničenje im je nedostatak fleksibilnosti te nemogućnost definiranja ograničavajućih pravila za neke od kategorija.

S druge strane, metode zasnovane na strojnom učenju nude dinamičniji pristup detekciji osjetljivih podataka. Za razliku od metoda zasnovanih na pravilima, algoritmi strojnog učenja uče iz primjeraka podataka. To znači da potencijalno mogu identificirati nove vrste osjetljivih informacija bez izričitog programiranja. Pristupi strojnom učenju kreću se od tradicionalnih algoritama poput stabala odlučivanja i strojeva s potpornim vektorima do naprednijih modela neuronskih mreža, uključujući rekurentne neuronske mreže (RNN), mreže dugog kratkotrajnog pamćenja (LSTM) i transformere poput BERT-a. Ti modeli mogu otkriti složene obrasce u podacima, čineći ih posebno učinkovitima za detekciju osjetljivih informacija u nestrukturiranim podacima ili u slučajevima kada je važan kontekst oko podatka.

Drugo poglavlje također daje uvid u različite metode predstavljanja ulaza vektorskim reprezentacijama (engl. *embeddings*) te različitih metoda za njihovu izgradnju. Te metode pomažu pretvoriti složene podatke u jednostavniji oblik koji potencijalno može razumjeti semantičko značenje prirodnog jezika. Detaljno se objašnjava zašto su te različite metode ugrađivanja važne i kako se mogu koristiti za razumijevanje podataka. To postavlja temelje za specifičan pristup predložen u ovoj disertaciji za detekciju osjetljivih podataka u strukturiranim skupovima podataka. Nakon toga, slijedi uključivanje tabličnog konteksta (različite ćelije iz iste baze podataka) i koji su trenutačni pristupi takvom problemu. Prikazuje da se trenutačno koristi statičan način uključivanja konteksta, otvarajući novi pravac za istraživanje aktivnijih pristupa.

Poglavlje tri opisuje razvoj novog skupa podataka, nazvanog DeSSI, dizajniranog za izazov detekcije osjetljivih podataka unutar strukturiranih skupova podataka. Adresirajući ograničenja postojećih resursa, ovaj skup podataka kombinira sintetičke podatke s pseudo-anonimiziranim stvarnim podacima kako bi ponudio sveobuhvatan alat za učenje i evaluaciju metoda detekcije osjetljivih podataka. Skup podataka ima za cilj obuhvatiti razne vrste podataka, uključujući razne vrtse osobnih podataka kako bi odražavao složenost detekcije osjetljivih podataka. Stvaranje ovog skupa podataka uključivalo je prevladavanje nekoliko izazova, a posebice zabrinutost za privatnost podataka i potrebu za raznolikom reprezentacijom vrsta osjetljivih podataka. Sintetički dio skupa podataka generiran je kako bi obuhvatio širok raspon scenarija osjetljivih podataka, dok su stvarni podaci, dobiveni iz javno dostupnih skupova podataka, pažljivo anonimizirani kako bi se osigurala privatnost. Ovaj hibridni pristup ne samo da poboljšava relevantnost skupa podataka za zadatke detekcije osjetljivih podataka već i zaobilazi ograničenja povezana s korištenjem isključivo sintetičkih ili stvarnih podataka. Ovo poglavlje konstatira važnost izrade skupa podataka koji odražava složenost i varijabilnost osjetljivih informacija u

stvarnim scenarijima. DeSSI uključuje preko 31.000 stupaca baze podataka, sadrži mješavinu različith vrsta podataka i uključuje i sintetičke i pseudo-anonimizirane stvarne podatke. Kako bi simulirao stvarne izazove, skup podataka uključuje postojeća ili odsutna zaglavlja stupaca, te time osigurava korisnost u nesavršenim uvjetima mogućim u stvarnim situacijama.

Poglavlje četiri uvodi novu metodu razvijenu za rješavanje izazova detekcije osjetljivih podataka unutar strukturiranih skupova podataka. Prepoznata su ograničenja postojećih pristupa, koji se kreću od sustava zasnovanih na pravilima do jednostavnijih modela strojnog učenja. Ovo poglavlje opisuje razvoj rješenja zasnovanog na strojnom učenju, posebice uz korištenje mogućnosti kontekstualiziranih vektorskih reprezentacija kako bi se poboljšala sposobnost modela za razumijevanje i obradu prirodnog jezika pronađenog unutar ćelija baze podataka. Začetak ove metode bio je pod utjecajem spoznaje da ni sustavi zasnovani na pravilima ni jednostavni modeli strojnog učenja ne hvataju dovoljno nijansirane odnose između značajki podataka i oznaka potrebnih za učinkovitu detekciju osjetljivih podataka. To je dovelo do usvajanja naprednije strategije, usmjerene na korištenje BERT-a (bidirekcijske reprezentacije kodera iz transformatora) za generiranje kontekstualiziranih vektorskih reprezentacija. Tako se nudi dinamička reprezentacija riječi u kontekstu, te se omogućuje dublje razumijevanje prirodnog jezika i značajno poboljšava sposobnost modela da točno ustanovi osjetljive podatke.

Da bi se prilagodila jedinstvenoj strukturi stupaca baze podataka i raznolikosti podataka koje sadrže, ova metoda predlaže novu tehniku formiranja ulaza. Ta tehnika kombinira zaglavlje stupca s vrijednostima ćelija, odvojenim posebnim znakovima za odvajanje, u jedan ulazni niz. Ovaj pristup osigurava da BERT može istovremeno razmatrati više vrijednosti ćelija, uključujući i unutarćelijski i međućelijski kontekst u svojoj analizi. U disertaciji se takav pristup opisuje kao aktivni pogled na kontekst koji sadrži jednu razinu apstrakcije manje nego dotadašnji statički pogled te time omogućuje modelu direktan uvid u ulazne podatke pri stvaranju vektorskih reprezentacija stupaca. Metoda transformira cijeli stupac baze podataka u ulazni niz za model, te demonstrira sposobnost metode da sačuva i iskoristi prirodni kontekst podataka.

Značajan izazov kojim se ova metoda bavi je ograničenje nametnuto maksimalnim brojem ulaznih tokena BERT-a. Kako bi se izbjegao potencijalni gubitak informacija zbog skraćivanja ulaza, metoda koristi strategiju dijeljenja i zasebne obrade dijelova stupaca koji premašuju ograničenje broja tokena. To osigurava sveobuhvatno pokrivanje podataka uz očuvanje integriteta kontekstualnih informacija. Osim osnovnog pristupa zasnovanog na strojnom učenju, metoda uključuje tehniku zasnovanu na pravilima u fazi naknadne obrade za određene vrste osjetljivih podataka. Ovaj hibridni pristup omogućava preciznu identifikaciju vrsta osjetljivih podataka koji se pridržavaju specifičnih formata ili pripadaju poznatim podskupovima, kao što su brojevi kreditnih kartica ili nacionalni identifikacijski brojevi. Integracijom logike zasnovanoj na pravilima model dobiva dodatni sloj specifičnosti koji poboljšava njegovu ukupnu točnost u

detekciji osjetljivih podataka.

Peto poglavlje produbljuje alternativne arhitekturalne pristupe za model detekcije osjetljivih podataka, što je motivirano težnjom za poboljšanjem i optimizacijom njegove izvedbe. Budući da hiperparametri strojnog učenja značajno utječu na njegovo ponašanje i ishode, ovo poglavlje istražuje više od same prilagodbe hiperparametara, te predlaže izmjene temeljne strukture samog modela. Takve arhitekturalne promjene ne ciljaju samo na prilagodbu hiperparametara već na temeljitu promjenu pristupa modela obradi i analizi podataka. Ovo istraživanje uključuje i pojednostavnjenja i unaprjeđenja inicijalno predloženog modela. Alternative se kreću od jednostavne metode koja izostavlja konkatenaciju stupaca do složenijih strategija koje aktivno uključuju podatke iz susjednih stupaca, bilo statički ili dinamički. Svaki pristup procjenjuje se s obzirom na njegov potencijal da postigne ravnotežu između računalne učinkovitosti i dubine kontekstualnog razumijevanja.

Prva opisana alternativa razmatra arhitekturu modela bez konkatenacije podataka stupaca, čime pojednostavljuje unos tako da on stane unutar BERT-ovog ograničenja broja tokena, bilo sekvencijalno bilo nasumično. Ovaj pristup ima za cilj smanjenje vremena izvršavanja ograničavanjem modela na obradu jednog unosa za svaki stupac, čime se smanjuju računalni zahtjevi. Međutim, ova jednostavnost donosi rizik da se zanemare složeni ili višestruki tipovi osjetljivih podataka unutar stupca zbog ograničenog opsega podataka koji se uzimaju u obzir u analizi. Proširujući kontekstualni opseg modela, sljedeće alternative istražuju uključivanje podataka iz susjednih stupaca. Te metode pretpostavljaju da susjedni stupci mogu ponuditi vrijedan kontekst koji bi mogao poboljšati sposobnost modela za detekciju osjetljivih informacija. Predlažu se dvije metode: jedna u kojoj se vektorska reprezentacija susjednih stupaca generira unaprijed i statički se uključuje, i druga koja aktivno integrira podatke susjednih stupaca na ulazu u metodu strojnog učenja. Iako ove metode teže iskorištavanju šireg konteksta podataka, suočavaju se i s povećanom složenošću modela i inherentnim izazovom učinkovitog učenja na skupovima podataka s varijabilnim brojem stupaca.

Poglavlje šest produbljuje eksperimente provedene za vrednovanje učinkovitosti predložene metode za detekciju osjetljivih podataka u strukturiranim skupovima podataka. Koristi se biblioteka FLAIR i ispituju se različite konfiguracije kako bi se pronašle optimalne postavke za hiperparametre stope učenja, epohe, veličine grupe podataka za učenje i više, a pritom se koristi destilirana verzija BERT-a i optimizator AdamW za učinkovito učenje. U poglavlju se raspravlja o metrikama vrednovanja koje se koriste za procjenu učinkovitosti modela te se ističe važnost mjere prisjećanja (engl. *recall*) za detekciju osjetljivih podataka. Naime, važno je minimizirati lažno negativne rezultate, gdje se osjetljivi podaci pogrešno ne identificiraju, zbog potencijalnih ozbiljnih kršenja privatnosti. Nasuprot tome, lažno pozitivni primjerci, iako manje poželjni, smatraju se manje štetnima jer samo rezultiraju nepotrebnim zamagljivanjem podataka. Ravnoteža između mjera preciznosti i prisjećanja ispravlja se korištenjem mjere F1,

čime se daje holistička mjera učinkovitosti modela.

Nekoliko eksperimentalnih podsekcija ovog poglavlja predstavlja usporedbe predložene metode s alternativnim pristupima, modelima iz srodnih radova i izvedbom na specifičnim skupovima podataka, uključujući razvijeni skup podataka DeSSI. Svaki eksperiment ima za cilj pokazati mogućnosti modela i njegovu prilagodljivost različitim vrstama podataka i zadacima detekcije. Daljnje usporedbe s modelima iz srodnih radova naglašavaju superiornu izvedbu predložene metode na različitim skupovima podataka, uključujući one koji nisu specifično prilagođeni za detekciju osjetljivih podataka. Ti rezultati potvrđuju robustnost modela i njegov potencijal za širu primjenjivost u zadacima semantičkog označavanja stupaca i identifikacije osjetljivih podataka.

Sedmo pogljavlje sažima doprinose disertacije o detekciji osjetljivih podataka u strukturiranim skupovima podataka i semantičkom označavanju stupaca. Istraživanje je uvelo novu metodu koja kombinira klasifikaciju zasnovanu na aktivnom kontekstu s heuristikama zasnovanim na pravilima, čime je značajno poboljšana detekcija osjetljivih podataka u odnosu na ranije pristupe. Razvijen je referentni skup podataka za vrednovanje i usporedbu metoda detekcije, čime je ispunjen ključni jaz u području. Istraživani su alternativni pristupi kako bi se vrednovale različite moguće hipoteze što je dovelo do sveobuhvatnog razumijevanja prednosti i ograničenja različitih arhitektura modela na ovom problemu. Predložena metoda demonstrirala je superiornu izvedbu u odnosu na osnovni model i postojeće modele iz srodnih radova čime je istaknuta visoka učinkovitost pristupa zasnovanog na aktivnom kontekstu.

Ključne riječi: Strojno učenje, Umjetna inteligencija, Strukturirani podatci, Osjetljivi podatci, Obrada prirodnog jezika

Contents

Chapter 1

Introduction

1.1 Motivation

Since the emergence of the Internet, and even more so in the last decade, the amount of data available to everyone has increased dramatically. The ability to share and process vast amounts of data is an excellent catalyst for the world's research and development, and the availability of this data has led to significant advancements in a wide range of fields, including healthcare, finance, and scientific research. However, with the increased availability of digital data, concerns about data privacy and security have also emerged. Sensitive data, such as personally identifiable information (PII) [1] and confidential business information are vulnerable to unauthorized access and misuse, resulting in a significant threat to individuals and organizations.

While the terms Personal data, Sensitive data, and Personally identifiable information are often intertwined, they represent different concepts.

Personal data refers to any information that can be used to identify an individual, either directly or indirectly. This can include a wide range of information such as name, address, email, phone number, date of birth, gender, nationality, and more.

Sensitive data, on the other hand, refers to any information that, if accessed or disclosed by unauthorized parties, could lead to harm or negative consequences. This can include social security numbers, driver's license numbers, passport numbers, and other forms of identification that can be used to impersonate or steal someone's identity. Sensitive data can also include financial information, medical records, and other types of confidential or proprietary information.

Personally Identifiable Information (PII) is a specific type of Personal data that directly identifies an individual, it might be a single data point or a combination of multiple data points. While a Social Security number is PII on its own, the name of a person itself is not (because multiple people might have the same name). But the combination of a name and an address might be.

While all three of these terms refer to intrinsically private information, the field is mainly

concerned with the term 'Sensitive data' and its detection process.

In addition to the moral reasons for the protection of sensitive data, regulations have been set in place around the world in the past 10 years. The General Data Protection Regulation within EU [2] sparked not only the widespread debate on data privacy and protection but also the creation of many different privacy acts and regulations worldwide, such as the California Consumer Privacy Act [3] in California, or the General Personal Data Protection Law [4] in Brasil. All of these regulations aim to give individuals a relatively high degree of control over their personal data and impose strict guidelines on the data collectors. As a consequence of the application of these regulations, the need for sensitive data detection and privacy protection has increased [5] and the need for the automation of such processes has arisen, driven by the ever-increasing amounts of digital data available and the ever-changing and evolving nature of what data is considered sensitive.

The described problems are collectively widely known as the sensitive data de-identification problem, which consists of two parts. In the first part, the sensitive data must be detected, and in the second part, it must be removed, encrypted, anonymized, or pseudo-anonymized. Therefore, Sensitive data detection is an essential area of research that focuses on the identification and protection of sensitive data in digital environments. The detection of sensitive data is critical for ensuring the security and privacy of individuals and organizations. There are several challenges associated with sensitive data detection, including the sheer volume of data and the complexity of identifying sensitive data in unstructured data sources such as emails and social media posts as well as in structured data such as database tables or spreadsheets. In addition to these two main data formats, sensitive data detection can also be performed on images, audio, or video, or in short multimedia [6]. This thesis addresses the first of the two parts of the problem, namely the detection process for sensitive data.

Since sensitive data can be found in a variety of data formats, and approaches to creating algorithms, models, and solutions for those problems vary because of it, the fields of research for structured, unstructured, and multimedia data have diverged. The task of sensitive data detection in unstructured text is essentially a specific case of the Named Entity Recognition (NER) [7] task, which relies on the context of Natural Language Processing [8] to better define and detect certain entities, which in this specific task would represent types of sensitive data. On the other hand, in structured data, such as a database, while there can be natural language connections present inside a single cell, there is no clear connection with other cells besides in which column or row they occur. That causes the solutions to mostly differ in the text embedding methods [9]. As for multimedia data, most approaches employ either Optical Character Recognition (OCR) [10] for images and video, or Speech-to-Text (STT) [11] recognition for audio, to get those formats into textual formats and employ either the methods from structured or unstructured data.

While the topic of sensitive data detection in unstructured data is thoroughly researched, the problem of sensitive data detection in structured data has not been addressed adequately in related work. Therefore, the topic of the dissertation follows the problem of sensitive data detection on structured datasets and fields related to it. Given the scarcity of research on this topic, it is imperative to leverage techniques utilized in similar problems and establish a comparative analysis to determine the most appropriate approach.

1.2 Hypothesis and contributions

The contributions of this work encompass various aspects aimed at advancing the field of structured sensitive data detection. Specifically, the following contributions have been made:

1. Proposal of a novel method: This work introduces a novel method for structured sensitive data detection. The proposed method utilizes an active context-based approach in combination with traditional rule-based heuristics to classify database table columns into one or more labels. By incorporating active context, the method leverages the surrounding information to improve the accuracy and effectiveness of the classification process. The method outperforms methods from related work on both their dataset and the newly proposed dataset.
2. Creation of a benchmark dataset: A new dataset is developed specifically for addressing the problem of structured sensitive data detection. This dataset fills a gap in the existing literature by providing a standard benchmark for evaluating and comparing different methods in this domain. Researchers and practitioners can utilize this dataset to assess the performance of their approaches and establish a baseline for future research.
3. Exploration of alternative approaches: The thesis explores alternative approaches that involve slight modifications to the model architecture and the notion of context. By proposing these alternative approaches, the work investigates different hypotheses and explores the potential benefits and limitations of various design choices. This exploration broadens the understanding of the problem space and provides insights into the effectiveness of different strategies.
4. Conducting comprehensive experiments: The proposed method undergoes rigorous experimentation to identify the optimal parameters and to evaluate its performance. Through systematic experimentation, the work aims to fine-tune the proposed approach and understand its strengths and weaknesses. Additionally, the alternative approaches are compared and analyzed to provide a comprehensive understanding of their performance characteristics.

Overall, this work makes significant contributions to the field of structured sensitive data detection by introducing a novel method, providing a benchmark dataset, exploring alternative

approaches, conducting comprehensive experiments, and showcasing the superior performance of the proposed method. These contributions advance the understanding and capabilities in this domain, paving the way for further research and development in the field.

1.3 Thesis organization

The thesis is organized into 7 chapters.

Chapter 1 serves as the introductory chapter, providing readers with an overview of the thesis structure and briefly summarizing the content of each chapter. It lays out the motivation for the problem addressed in the thesis and highlights the contributions it aims to make. By setting the stage for the subsequent chapters, Chapter 1 creates a roadmap for the reader, ensuring a clear understanding of what to expect.

Chapter 2 delves into the background knowledge necessary to comprehend the thesis and its contributions. It provides a comprehensive overview of the history and evolution of methods used in the field, tracing their development over time. The chapter also explores related work, examining how other researchers have tackled similar problems and discussing the strengths and limitations of their approaches. By establishing this foundation of knowledge, Chapter 2 contextualizes the thesis and enables readers to grasp the significance of its contributions.

Chapter 3 introduces a novel dataset that has been specifically created for the problem of sensitive data detection in structured datasets. The chapter provides detailed insights into the dataset's construction, outlining the rationale behind its creation and the considerations taken into account. It highlights the unique features of the dataset and explains how it addresses the limitations of existing datasets. By presenting this new resource, Chapter 3 enhances the thesis's practical relevance and demonstrates its potential impact on the field.

Chapter 4 focuses on the main proposed method, providing a comprehensive description of its design and architecture. It explains the reasoning behind each aspect of the method, including the selection of specific techniques and methodologies. The chapter highlights the unique characteristics of the proposed method and explains how it addresses the challenges identified in the earlier chapters. By presenting a detailed overview of the method, Chapter 4 equips readers with the necessary knowledge to understand and evaluate its effectiveness.

Chapter 5 explores alternative approaches to the initially proposed approach that slightly modify the input methods or architecture of the proposed method. It presents these alternative approaches as potential solutions to test additional hypotheses and further investigate the problem. The chapter discusses the motivations behind these variations and the potential advantages they may offer. By considering alternative approaches, Chapter 5 broadens the discussion and encourages critical analysis of different strategies.

Chapter 6 focuses on the conducted experiments, presenting the results obtained from dif-

ferent angles. It begins by comparing the performance of different parameter values on the main method, providing insights into the impact of these choices on the model's effectiveness. The chapter then compares the main method to the alternative approaches discussed in Chapter 5, highlighting the strengths and weaknesses of each approach. It further compares the proposed method to existing methods from related work on their respective datasets. Lastly, the chapter presents the results of the comparison on the new dataset created as part of the thesis. By providing a thorough analysis of the experimental results, Chapter 6 substantiates the claims and contributions of the thesis.

Chapter 7 serves as the concluding chapter, summarizing the main findings and contributions of the thesis. It reflects on the limitations encountered during the research and discusses potential avenues for future work and improvements. The chapter also emphasizes the broader significance and implications of the thesis's contributions. By providing a comprehensive conclusion, Chapter 7 brings the thesis to a close, offering a synthesis of the research journey and highlighting its broader implications for the field.

Chapter 2

Background and Related Work

In the past decade, the need for the automation of sensitive data detection has increased because of laws and policies set in place for the need for protection and regulation. Moreover, the amount of digital data has increased drastically [12] and without automated processes, the detection and protection process is not feasible. The regulations about what is considered sensitive data are also always adapting and can be changed with time [13].

In the structured sensitive data detection task, the data consists of one or more database tables, that consist of cells that are ordered into columns and rows. A single cell can be in a format ranging from a single character, number, or word to something much more complex like a sentence or multiple sentences. The rows of a database table mostly represent a single observation or example, and the columns represent a feature or variable that describes the observation. Since values inside of the same column in the table represent the same feature, they are mostly very similar, or at least of the same type.

Thus, the problem of structured sensitive data detection is going through the columns in a database table and determining for each of them whether they contain sensitive data or not, as well as what type(s) of sensitive data they are. The formulation of the problem can be viewed as a subtype of the semantic column labeling problem [14], which is a more general problem, that requires the classification of data into one or more of several categories that are very broad, and not as specific as sensitive data. The semantic column labeling problem has strong similarities to the named entity recognition problem [15] since named entities must be detected in the data.

Since the topic of sensitive data detection, especially for structured data, is not thoroughly researched, this thesis will also look at the broader problem of semantic column labeling and present possible solutions for that problem, as well as compare the proposed method with other solutions on that task.

When it comes to machine learning, there are many different approaches that can be used to solve the same task [16]. Some of these approaches may be more suited to certain types of data or problem domains, while others may be more general-purpose or flexible. For example, some

machine learning algorithms are designed to work with structured data, such as tabular data in a spreadsheet or database, while others are designed to work with unstructured data, such as text, images, or audio.

While some data can be present in a format that is easily understandable to computers, and with which they can easily work, like an array of features represented by numbers, some data comes in a much more complex form, such as natural language text or image pixels. Those formats are high-dimensional and sparse, which makes them difficult to use in machine learning models.

An algorithm is needed to determine in what form that input will be interpreted, transformed, and forwarded to the model. The transformed features are called embeddings [17]. For example, a common embedding of a word consists of a vector of real-valued numbers that efficiently represent the word. Since the values inside of cells in structured data are most often in textual format, they first need to be transformed into textual embeddings before a machine learning model tries to work with them.

In this chapter, firstly, natural language embeddings will be covered, explaining what they are and their different types. After that, we'll look at rule-based and machine learning-based models in general, as well as focusing on how ML models handle context in structured data. This overview will help set the stage for introducing our proposed method and addressing gaps and challenges in the current approaches to sensitive data detection.

2.1 Natural language embeddings

Embeddings are a technique for learning a lower-dimensional representation of high-dimensional data [18]. The goal of an embedding is to transform the input data into a more compact, lower-dimensional representation that captures the essential features of the data. While embeddings for text need to encapsulate the meaning of words or even the entire language, image embeddings need to capture the important parts of the picture. Although simpler inputs (an array of features for example) for simpler machine learning models or rule-based methods might not need any special embeddings, more complex tasks might need complex machine learning models to even create understandable embeddings which will later on be used by other machine learning models to interpret them.

The choice of the embedding algorithm can have a significant impact on the performance of machine learning models [19], because the quality of the embeddings directly affects the ability of the model to capture the essential relationships and structure of the data.

There are several efficient ways for textual data to be transformed into a numerical form that a computer and machine learning methods can work with. These are, by and large, models trained to perform a mapping of words or phrases into a real-valued vector of a fixed size such

that some desirable semantic properties and linguistic word relations are satisfied. Word embedding algorithms are the standard for improving model performance in numerous NLP tasks. The algorithm takes a variable-length word, phrase, or sentence and transforms it into a vector representation of fixed dimensionality that consists of floating point numbers. The algorithm can either take the whole text at the same time or parts of the text separately. Depending on what information from the input text the textual embeddings use, and how they use it, the embedding methods are separated into one of several categories [20]. Some of these methods also use combinations of categories to reap the benefits of each of them. In the continuation of this section, the embedding methods categories are described in more detail.

2.1.1 Character-level embedding models

Since sentences are essentially sequences of words separated by spaces, and words are sequences of characters, character-level embedding algorithms use character-level information to represent words as vectors [21]. There are a few reasons why this type of embedding is used. The most obvious one is the simplicity, while there exists an infinite amount of sentences and an almost infinite amount of words, there exists only a finite and small collection of characters. This allows the model to embed accurately every set of possible characters, whereas if taken as words, some problems could occur if the words haven't been seen before. Another advantage is that languages like Chinese do not use the alphabet, they use a logographic system of characters [22]. This means that words are made up of symbols instead of letters, and each symbol can have its own meaning, and taking them into account as symbols or a set of symbols could prove useful.

The most often used approach for the creation of character-level embeddings are Convolutional Neural Networks (CNN) [23]. Convolutional neural networks are a famously proven network architecture for computer vision because they can learn spatial features, eliminating the need for feature extraction steps. The same advantage can also be applied to word representation tasks. Namely, one can view a word or phrase as a one-dimensional image. For example, when one sees a date or an address, one immediately knows what it is by just looking at it. The fact that CNNs are very good at learning internal structures can be extended to learning complicated morphological structures that words may have and learning semantic relationships between different character sequences. The textual data is first preprocessed by converting it into a sequence of one-hot encoded characters [24]. One-hot encoded vectors are binary vectors where only one element is "hot" (set to 1), and all other elements are "cold" (set to 0). They are used to represent categorical data, where each category is uniquely represented by a single element in the vector. This sequence of characters is then passed through a CNN model, which consists of a series of convolutional layers that filter important parts of the input, and pooling layers that pick out the most important filtered features and reduce the dimensionality

by removing the non-important features. The combination of convolutional and pooling layers enables the CNN to learn useful representations of the input data, by capturing important local features in the data and combining them into a global representation.

2.1.2 Subword-level embedding models

Subword-level embeddings are a type of word embedding method that represent words as a sequence of smaller subword units, rather than as a single atomic unit like a word or a character [25]. Subword-level embeddings can handle words that are not present in the training data, by breaking them down into subword units that are already represented in the vocabulary. They are typically learned using unsupervised methods that break words down into smaller units, such as character n-grams or byte pairs. These subword units are then used to generate a unique representation for each word based on its constituent subword units. The resulting embeddings capture the morphology and structure of words mostly better than character-level embeddings.

One of the most popular subword-level embedding methods is Byte Pair Encoding (BPE) [26], which generates subword-level embeddings by iteratively merging the most frequent pairs of consecutive characters in a corpus until a desired vocabulary size is reached. The resulting subword units are used to generate embeddings for each word based on the frequency of their constituent subword units.

Another popular subword-level embedding method is FastText [27], which generates subword-level embeddings by first breaking each word down into a set of character n-grams (e.g., "ing", "eat", "piz"), and then learning a vector representation for each subword unit. The embeddings for each word are then computed as the weighted sum of its constituent subword vectors.

2.1.3 Context-level embedding models

Context-level embedding methods were developed on the basis of famously articulated Firth's 1957 notion "you shall know a word by the company it keeps." [28], thereby defining each word with the words it most often co-occurs with. This approach does not put emphasis on the syntactical meaning of words, but rather on the semantical meaning of words. Meaning, character-level and subword-level embeddings rely on the character composition of the words, while context-level embeddings rely on their meaning. Context-level embeddings regard cat and dog as very similar words because they are often present in the same context of for instance a domesticated pet, although they consist of entirely different characters. This allows for the capturing of many semantic properties that a word might have, and also concepts like synonyms or antonyms which the previously mentioned approaches can not comprehend properly. One of the main drawbacks of context-level embeddings are out-of-vocabulary (OOV) words, which do not appear in the training data. Since embeddings are learned from the distributional properties

of the training corpus, if a word does not appear in the training data, there is no contextual information available to learn its embedding. Therefore, the model cannot assign a meaningful vector representation to the word, whereas methods that work with subwords or characters can work with any word.

The most prominent context-level embedding methods are Word2Vec and GloVE.

Word2Vec [29] is a neural network-based method that learns distributed representations of words based on the distributional hypothesis which states that words that appear in similar contexts tend to have similar meanings. Word2Vec takes a large corpus of text as input and learns to predict the context in which each word appears, given its neighboring words or phrases. It does so in two ways, the first one, called Skip-gram, predicts the surrounding words given a target word, while the second one, called CBOW (Continuous Bag Of Words), predicts a target word given its surrounding words. The resulting embeddings capture the semantic and syntactic relationships between words.

GloVe [30], which stands for Global Vectors for Word Representation, is also based on the distributional hypothesis but uses a different approach to learn the embeddings. Instead of predicting the context in which each word appears, GloVe constructs a co-occurrence matrix that counts the number of times each word appears with every other word in the corpus. The matrix is then factorized to obtain a low-dimensional vector representation for each word that captures its global co-occurrence statistics.

2.1.4 Contextualized word embeddings

While context-level embeddings are good at capturing the general semantic meaning of words in a vector space, they were created on a large training set and capture some meaning of all the possible occurrences of the word in the contexts seen in the training data. They cannot take the current context of the word that is present into account and derive a specific embedding for the word in the current context. Contextualized embeddings, on the other hand, try to circumvent that deficiency by creating embeddings based on the context the words appear at the time of inference. Contextual embeddings are typically learned using deep neural networks, such as recurrent neural networks (RNNs) or transformer models. These models are trained on large corpora of text to learn the distributional properties of words and phrases in context. During training, the model is presented with a sequence of words, and it learns to predict the probability of each word given its surrounding context. Contextualized models allow the creation of **active embeddings**, and not only statically pre-generated, **passive embeddings** for words. Active embeddings are more accurate for the current occurrence, and the model is able to generate contextual embeddings for any text input, even if the words or phrases have never been seen before, thereby avoiding the problem of OOV words.

The most popular contextualized embedding models are ELMo [31], BERT [32] and GPT

[33], which will be explained and discussed later, in Section 2.3.

2.2 Rule-based methods

The most basic methods of automation are rule-based methods, which are techniques that rely on explicit, predefined rules or logical statements to make decisions or perform tasks. Rule-based methods can be used for a variety of tasks, including decision-making, pattern recognition, and natural language processing.

In rule-based methods, rules are typically defined using a set of if-then statements, where the input data is evaluated against a set of conditions, and a decision or action is taken based on the outcome of the evaluation.

The key advantages of rule-based methods are [34] [35] [36]:

- Limited data: Rule-based methods can be effective in situations where the available data is limited or noisy. By defining a set of rules that capture domain-specific knowledge or heuristics, it is possible to make accurate predictions or decisions with only domain knowledge, without needing any data.
- Interpretability: Rule-based methods are mostly interpretable, while most machine learning methods, such as deep learning, are not. This can be important in situations where it is necessary to understand how the model is making its predictions or decisions, or where the model needs to be auditable.
- Domain-specific knowledge: Rule-based methods can be effective in situations where domain-specific knowledge or heuristics are available. By incorporating this knowledge into the rules, it is possible to improve the accuracy and effectiveness of the model.
- Regulatory compliance: Rule-based methods can be useful in situations where regulatory compliance is important. For example, in industries such as healthcare or finance, it may be necessary to use rules to ensure that decisions are made in compliance with legal or ethical requirements.

Because rule-based methods rely on explicit, deterministic rules to make decisions or predictions, they are best used on types of data that come in a strictly defined format and are highly structured.

The most often used rule-based methods include regular expressions, lookup tables, and expert systems.

2.2.1 Regular expressions

Regular expressions are a pattern-matching technique that uses a set of rules to identify and extract specific patterns or sequences of characters from textual data. Regular expressions are

hand-crafted by domain specialists and consist of a sequence of characters that represent a pattern of interest [37]. They decisively judge if the desired pattern is present in the data or not. For example, in natural language processing, regular expressions can be crafted to locate the occurrence of data types that come in specifically defined formats like email addresses.

2.2.2 Lookup tables

Lookup tables are a type of data structure used in computer science and machine learning to map input values to output values based on predefined rules [38]. The structure consists of a table or matrix that stores the output values for a set of input values, along with a set of rules that define how the output values are generated. They are hand-crafted by domain experts and require no data to be created.

The process of using a lookup table typically involves taking an input value and searching the table for the corresponding output value. If the input value matches a value in the table, the corresponding output value is returned. If the input value does not match any value in the table, the output value is typically interpolated based on the values in the table.

For example, in natural language processing, a lookup table might be used to map words to their corresponding classes or part-of-speech tags.

2.2.3 Expert systems

Expert systems [39] are a type of rule-based method that uses a knowledge base and a set of inference rules to solve problems or make decisions in a specific domain. Expert systems are designed to mimic the decision-making ability of a human expert, and they use a set of rules and heuristics to reason about complex problems and provide advice or recommendations.

Expert systems are typically composed of two main components: a knowledge base [40] and an inference engine [41]. They are often constructed by open systems such as Clips [42] and Drools [43]. The knowledge base contains a set of rules, facts, and heuristics that are specific to the domain and are hand-crafted for the specific task, while the inference engine applies the rules and heuristics to the input data to generate a solution or recommendation.

Expert systems are often used in domains where there is a high degree of expertise and domain-specific knowledge.

2.3 Machine learning methods

Unlike most rule-based methods that rely on manually defined rules for detecting patterns, machine learning methods employ algorithms to automatically recognize patterns without explicit

communication. Since basic machine learning approaches like decision trees or SVM-s (Support Vector Machines) are not equipped to adequately capture the nuanced and intricate relationships within natural language, more advanced algorithms that effectively capture the contextual nuances of the entire input are necessary. In continuation, some of the commonly used machine learning methods in NLP are described.

2.3.1 Hidden Markov model

Hidden Markov models (HMM) [44] are categorized as generative models that use latent variables (hidden states) representing entities (outputs) to predict observable variables (inputs). The hidden states are interconnected and have probabilities of transitions from one to another as well as probabilities of producing a particular input. The model maximizes the joint probability for the entire sequence of tags along with the entire input sequence, rather than for a single tag, because in this way previous words and tags change the classification of subsequent tags. Chen et al. [45] used HMMs on data introduced by Stubbs et al. [46]. In the preprocessing part, each word was embedded into a vector and this embedding was further given as input to the model. The model was allowed to use as many hidden states as the data itself dictates by using the latent Dirichlet process [47]. This allowed the model to capture variations in the data and thus create more distinct categories. For example, the word “a" by itself is not a sensitive word and usually suggests that the word to come is not sensitive either, but if the words “works" and “as" are present before the word “a" then it suggests that the next word will be a sensitive word that would represent an occupation.

2.3.2 Conditional random fields

Conditional random fields (CRFs) [48] are a generalization of HMMs. They follow the same idea of hidden states, except that the states are undirected, which allows the model to use information from both previous and subsequent inputs as well as possibly other features represented as hidden states. The most important difference between CRFs and HMMs is that CRFs are discriminative rather than generative models, because they maximize the conditional probability of outputs given inputs, whereas HMMs maximize the joint probability of inputs and outputs co-occurring. These differences allow the CRF model to create arbitrary features that need not be statistically independent and are not restricted to modeling dependencies of hidden states and their associated observations. These arbitrary features are often handcrafted and specific to the domain. In the field of sensitive data detection, they are often created by rule-based methods since the rule-based methods are better at detecting certain kinds of sensitive data [16].

Implementations of CRFs for the task of de-identification use different features to try to predict the most likely labels. For example, Berg and Dalianis [49] used lemmas, the first few

and last few letters of words, and binary and integer indicators, among others. If the word consists only of numbers, the binary indicator would be a “1", and if not, then a “0". Similarly, the integer indicator could indicate how many letters are in the word. Liu et al. [50] used one CRF that focused on various word-level features such as Bag of Word and orthographic features, such as word length, capitalization punctuation, etc. They also used another CRF that focused on character-level features, such as Bag of Characters, consisting of used unigrams, bigrams, and trigrams, as well as relevant sentence information. By using both CRFs they aimed to incorporate both character-level and word-level information, as well as some hand-crafted rules.

2.3.3 Recurrent neural networks

Recurrent neural networks (RNNs) [51] [52] are types of neural networks that contain an internal state (latent variable) that is modified by inputs and produces outputs. The state thus acts as a kind of memory that allows past words to influence future output decisions. Like the HMM, RNNs also model the distribution of a sequence of observations from latent variables, but RNNs have one latent variable that is changed by each input that comes to it, while HMMs have multiple latent variables that are not changed by the inputs, instead, only transitions between each other using previous states and the current input are changed. Srivastava et al. [53] used two types of RNNs on sensitive data detection tasks, the first of which generated the new internal state from the previous internal state and the current input, while the second RNN used the output of the previous internal state along with the input of the current state to generate the new internal state. The RNN’s input was an embedding of the target word and its surrounding words to better capture short-term temporal dependencies.

2.3.4 Long short-term memory

Long short-term memory (LSTM) [54] is a modification of an RNN that facilitates recall of previous input, and solves some of the problems RNNs have faced when processing long sequences, such as vanishing or exploding gradients when training the neural network. LSTM has the same general architecture as an RNN, the only difference is that the internal state (memory) is more complex. It uses several matrices represented in the form of gates. The first gate decides which part of the input modifies the memory, the second decides which parts of the memory are forgotten, and the third gate decides which parts of the memory are used to generate the output. Implementations of LSTMs for the de-identification task mostly use a Bi-LSTM consisting of two LSTMs, the first of which trains on the sequence as it normally is, and the second on a sequence with a reversed order of words. This means that at each time step, a Bi-LSTM considers not only the preceding context (past information) but also the following context (future

information).

Richter-Pechanski et al. [55] used a BI-LSTM with a concatenation of character-level word embeddings and embeddings obtained from ELMO (Embeddings from Language Models [31]), a word representation model trained on large amounts of unlabeled data. Madan et al. [56] also used a Bi-LSTM, but with character-level embeddings concatenated with POS tag embeddings. Both approaches were used for the detection and de-identification of medical data, and significantly outperformed traditional approaches.

2.3.5 BERT

BERT [32] is a more recent deep learning model that uses attention through bidirectional transformers [57] to capture important features in natural language. It allows the model to consider the entire input while predicting each output, and the model trains itself on which part of the input to pay the most attention to. The model is pre-trained on huge amounts of unlabeled data using a masked language model, laying a good foundation for transfer learning to a variety of different domains.

Garcia-Pablos et al. [58] and Johnson et al. [59] used BERT for the task of sensitive data detection and de-identification. They tokenized their sentences as inputs to a pre-trained BERT and refined it with a fully connected linear layer that has the outputs of BERT as inputs, and the log-likelihood of the classes as outputs. These approaches outperformed traditional approaches as well as more recent machine learning models.

2.4 Embedding context for structured data

For the task of sensitive data detection in structured data, for each cell in a database table, one can use any of the previously mentioned approaches to extract information. However, the information from the other cells might also be of significant importance, especially the data from the other cells in the same rows or columns as the currently looked-at cell. These are important, since the row describes the same observation, and the column has the same feature. In a recent work [60], it has been shown that the solutions to the structured sensitive data detection could benefit from using NLP and leveraging the context inside of and between cells in the table to deduce embeddings of each cell. When referring to the context of other words inside of the same cell, the term *intra-cellular context* will be used, and when referring to the context of other cells, *inter-cellular context* will be used.

When creating an embedding of a cell, column, or row of a database table, the context of adjacent cells, columns, or rows might also be of significant importance, and providing them as an additional input might result in better embedding. The first question that may come to

mind is: why not use the whole table as the context? It certainly is a valid proposal, since the more context one provides, the better the result might be. While the idea seems sound at first, on second thought, three problems present themselves:

- The first problem is that, by creating embeddings, one is trying to formulate a lower-dimensionality representation of our input, that ought to capture the most important features, which implies that it should not be cluttered with possibly useless information from distant rows and columns in the database table. A counter-argument could be that a good embedding method would ignore unimportant information from distant places in the table, anyway.
- The second problem comes from the variable input size. Since the database table does not have a strictly defined amount of rows and columns, this limits the choice of the embedding method that can be used.
- The third problem comes from the computational complexity. Depending on the embedding method and how the context is used, the computational complexity rises with the increase of the context size, and since the number of columns can be in the thousands, and the number of rows in millions, it is simply too complex for the currently available computational resources, which might change in the future.

Therefore, certain compromises in the amount of context need to be made. Multiple approaches have been proposed and used by related work for the task of semantic column labeling, or more generally, in any tasks that work with structured data.

In Table 2.1 an overview of the related work is provided, along with the methods used and the limitations observed.

Some commercial approaches like Trifacta [61], Power BI [62], and IBM [71] mostly use rule-based methods, while other approaches such as Cloud DLP [63], PII Catcher [64], Nightfall [72], Presidio [73], and Gretel AI [74] use both rule-based and machine learning methods on database cells. These approaches fall under the simplest category 1) in Figure 2.1 *, with none of them taking any inter-cellular context into account.

SIMON [65] attempts to solve the semantic column labeling problem by employing a character-level convolutional neural network (CNN) together with a long short-term memory (LSTM) network [54] to produce embeddings of individual cells, combine them into a singular embedding and subsequently classify the column into one of the possible labels. SIMON, however, does not use any kind of context and only averages the cell values of each column to predict the final label.

Sherlock [66] aims to consider relations between cells and formulates more complex concepts of context. To this end, Sherlock uses a deep neural network architecture that does not only use the current cell in the table to generate its features but also incorporates context by

*License -https://www.sciencedirect.com/science/article/pii/S0957417423004256

Description of related work approaches and limitations		
Related work	Methods	Limitations
Trifacta [61], Power-BI [62]	Rule-based methods	No natural language understanding No context No adaptability
Cloud DLP [63], PII catcher [64]	Rule-based methods Basic machine learning methods on singular cells	No intra-cellular context No inter-cellular context
SIMON [65]	Character level embeddings of individual cells	No intra-cellular context No inter-cellular context
Sherlock [66]	Single cell embeddings Static column embeddings Statistical column features	No intra-cellular context No active inter-cellular context
SATO [67]	Deep neural networks Static context of other columns and statistics	No intra-cellular context understanding No inter-cellular context
SeLaB [14]	BERT for intra-cellular context Static inter-cellular column context	No active inter-cellular context
TaBERT [68]	BERT for intra-cellular context Active inter-cellular context of entire row	Created for question answering task No context of other values in column
TABBIE [69]	BERT for intra-cellular context Static inter-cellular context from row and column	Created for corrupt cell detection Not looking at multiple cells while creating embedding with natural language

Table 2.1: The description of related work with the methods they are using, and what in our mind are the limiting factors or shortcomings [70].

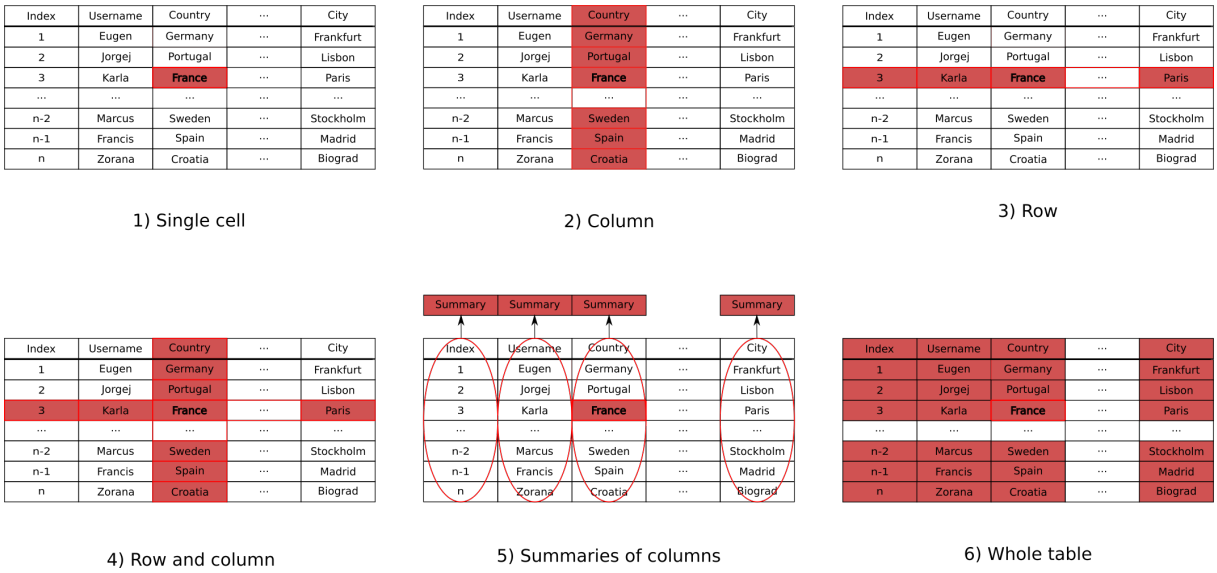


Figure 2.1: Approaches to inter-cellular context representation. In all cases, the cell with the value 'France' is the currently looked-at cell. 1) Only the current cell is taken into account, there is no context. 2) The current cell and the column of that cell are taken as context. 3) The current cell and the row of the cell are taken as context. 4) The current cell, the row, and the column are taken as context. 5) Summaries for all columns are created a priori and they are given as context together with the current cell. 6) All the cells in the table are given as context [70].

considering all other cells in the same column through a modified version of Paragraph Vectors [75] which are generated beforehand. This approach is shown under 2) in Figure 2.1, and includes taking into account only the summary of the current cell's column. Together with the paragraph vectors, Sherlock also employs statistical features of the current column such as character distributions and average cell lengths.

SATO [67] extends on Sherlock [66] by incorporating a Topic Prediction model which works on the entire table. In such a way, SATO also uses other columns in the database to extend the context of a cell, which can be seen under 5) in Figure 2.1. While both SATO and Sherlock consider table context, they do so in a passive way, meaning that the context is generated beforehand and the model is not allowed to actively look at other cell values while creating an embedding for the current cell (see also Section 2.1.4).

SeLaB [14] approaches the semantic column labeling problem using a two-step processing. In the first step, the method uses BERT to generate embeddings and classify each cell individually, and subsequently calculate the most likely label from these. In the second step, SeLaB repeats the first step with additional information of other predicted column labels from the first step, thereby using the context of all other columns to determine the label of each column. This approach can be seen as a variation of the approach under 5) in Figure 2.1.

TaBERT [68] tackles a slightly different problem as it aims to answer the question in the form of a sentence and predict which cells from the table give the best answer to the question. Its value for the current problem is in giving a glimpse into a possible way to incorporate

dynamic context. TaBERT attempts to solve the question-answering problem by calculating the similarity of the sentence to a row in the table and then feeding both the sentence and the most similar row into BERT [32], as shown under 3) in Figure 2.1, thus allowing the model to consider all cells inside of the same row at the same time, effectively allowing the model to learn what it needs to pay attention to in its context.

TABBIE [69] is a recent method that first creates an embedding for each cell separately with BERT. Afterward, individual embeddings of cells in the same row are taken as input into the transformer. The process is repeated for each individual cell in the column. In the end, the outputs are averaged to obtain the embedding, which takes into account the context of both the current row and column. This is a strategy depicted under 4) in Figure 2.1.

Chapter 3

The Proposed Dataset

3.1 Motivation

Semantic column labeling is a broad problem in natural language processing that involves assigning labels to columns in structured datasets based on their semantic meaning. There are several methods that have been developed to tackle this problem, including rule-based methods, machine learning-based methods, and hybrid approaches that combine the two. However, while these methods can be effective at solving general semantic labeling tasks, they may not be suitable for specific challenges such as sensitive data detection in structured datasets. This is because sensitive data often has unique characteristics and requires specialized approaches for accurate detection.

To address this challenge, a novel, context-full approach is proposed in this thesis. However, one of the main challenges faced in addressing the challenge was the lack of a relevant dataset for training and evaluating the method. To overcome this challenge, a hybrid dataset was created that consists of both synthetic and pseudo-anonymized real-world data. The synthetic data was generated using a data generation tool that allows one to create datasets with a wide array of sensitive data types, while the real-world data was sourced from publicly available datasets and then anonymized to protect privacy.

This novel, hybrid dataset allows for training and evaluation of the proposed method on a diverse set of data types, including personally identifiable information (PII), financial data, and health data. In addition, it allows for overcoming the limitations of using only synthetic or only real-world data for training and evaluation. In the continuation of this section, the specific data challenges that were encountered during the creation of the dataset are described, including the issues related to data quality, data privacy, and data diversity. Furthermore, a detailed description of the dataset itself is provided, including its size, composition, and labeling scheme.

3.2 Challenges in sensitive data detection datasets

In the area of machine learning, building a model to recognize sensitive data is crucial for various applications, ranging from healthcare to finance. However, obtaining real-world datasets that contain personal information is a challenging task due to privacy concerns. Furthermore, the use of real-world datasets for training and testing machine learning models can lead to potential security issues, such as the extraction of sensitive data from the classifier itself [76] or from the neural networks through statistical inference [77].

These considerations necessitate the creation of synthetic datasets. Synthetic datasets are commonly used in disciplines where privacy is a concern, such as medicine [78] and computer vision [79]. Synthetic data can be used both to train the model and to augment a real-world dataset. In related work, synthetic datasets were created by either a direct use of tools such as Faker [80] or by taking columns from other datasets, such as VizNet [81], and refining them to take a subset of columns matched to DBpedia column types. While these datasets can be decent as a comparison for machine learning models in general, they do not represent sensitive data types and therefore are not the best possible representation of data to handle the problem at hand. This is because they do not accurately represent the variety of sensitive data types that can exist in real-world datasets. These synthetic datasets may not contain sensitive data types such as credit card numbers, social security numbers, or other personally identifiable information that is essential for sensitive data detection tasks. Furthermore, such synthetic datasets may not be able to capture the intricacies and complexities of real-world datasets, making them less effective at identifying sensitive data.

Therefore, using these datasets as a basis for comparison for machine learning models, in general, may not be an accurate representation of the performance of these models when working with sensitive data. Instead, the creation of a new, more comprehensive dataset that includes a wide range of sensitive data types is necessary to accurately train and evaluate machine learning models for sensitive data detection tasks.

Furthermore, the datasets from related work do not seem to incorporate column headers for various reasons. Most of the related work, such as Sherlock [66] or SIMON [65], either state that column headers are not reliable sources of information or they use column header data to generate the column labels. While column headers can sometimes be empty, provide useless or even misguiding information, and models should not rely solely on them, the notion that they are useless in most cases can not be considered correct. Instead, it is a common case that column headers carry substantial information about the column data type. Additionally, column headers are almost always present in real-world data in some form and should be incorporated into synthetic datasets and used by models for context creation.

Column headers in structured data refer to the names or labels given to the columns in a

dataset, which can provide a useful context for the data contained in each column. While some related work in the field of semantic column labeling, such as Sherlock [66] or SIMON [65], ignore column headers altogether or use them only to generate column labels, column headers are an important aspect of data that should not be overlooked. Column headers can provide a wealth of information about the data contained in each column, such as the type of data, format, or domain-specific terminology, which can be critical for accurately labeling sensitive data.

Furthermore, column headers are almost always present in real-world data in some form and, therefore, they should be incorporated into synthetic datasets used for training machine learning models. By doing so, the models can learn to recognize the patterns in the column headers and use them to inform their predictions. While it is true that column headers can sometimes be misleading or provide little information, they should not be discounted entirely. Instead, models should be trained to take into account the context provided by column headers, in addition to the data contained in each column.

Attention should also be given to the problem of datasets given in [81] and [80], where each column is assigned only one label, which is not always reflective of real-world data. In practice, it is not uncommon for a single column to contain multiple types of sensitive data, such as a column in a database table that stores both credit card numbers and email addresses. In such cases, a single label for the column is insufficient for accurately identifying and protecting sensitive information. This highlights the need for more nuanced and granular labeling schemes in order to better represent the complexities of real-world data and create more robust machine learning models for sensitive data detection.

Lastly, there are multiple other challenges in real-world structured data, which can potentially limit the efficiencies of machine learning models and which require careful data preparation [82]. Some of these challenges include input errors, inconsistent column headers, human errors, the use of shorthand or business-specific labels for columns, etc. A common issue is input errors, where data is not entered correctly or completely, leading to missing or incorrect values. Another challenge is inconsistent column headers, where the same column may be labeled differently across different datasets, making it difficult to integrate the data. Human errors also play a role, where input errors can lead to incorrect data being entered into the system. Additionally, there may be a lack of standardization of the labels used to describe the data, with the use of shorthand or business-specific terms leading to confusion when trying to compare and analyze data. These challenges highlight the importance of careful data preparation. They need to be considered to some degree to ensure that the machine learning models can effectively process and analyze the data. Finally, there is a need for domain expertise and collaboration between subject matter experts and data scientists to accurately identify and address these challenges in real-world structured data.

3.3 The proposed dataset description

To tackle the problem of building machine learning models for sensitive data detection in relational databases, it is necessary to have appropriate data for training and testing. However, as discussed earlier, the availability of real-world datasets containing personal information is limited due to privacy concerns.

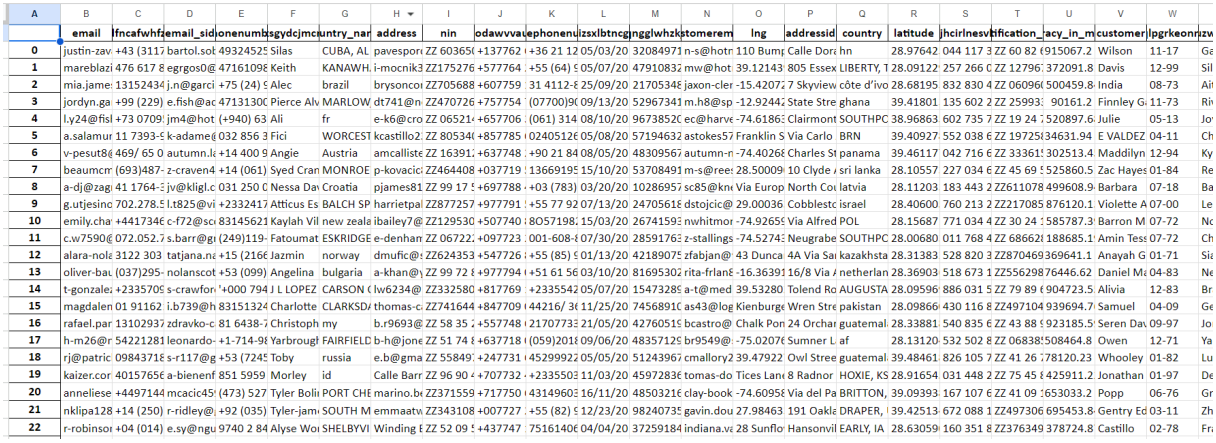
Therefore, new relational data models were created to train and test the proposed method. These models, incorporated in a dataset, are designed to simulate the real-world environment and provide a diverse set of data types that can be encountered in real-world scenarios.

One of the main challenges in real-world structured data is the problem of missing or misleading column headers. In the proposed dataset, this challenge was simulated to ensure that the proposed model would not rely solely on column headers for context creation. By incorporating missing and misleading column names, it was possible to provide a dataset that is more representative of real-world scenarios and can help improve the performance of machine learning models in sensitive data detection tasks.

The proposed dataset consists of snippets of personal data aggregated from various open-source datasets, such as Kaggle, as well as synthetic data generated by Python packages such as Faker [80]. Additionally, pseudo-anonymized real-world data were incorporated that were provided by an organization under a strict third-party confidentiality agreement. By removing any identifying information from the real-world data and replacing it with artificial identifiers (pseudonyms), any potential breach of confidentiality was avoided. This approach allowed for incorporating real-world data into the dataset without compromising the privacy and security of the individuals involved. This hybrid dataset incorporates a wide array of sensitive data types, such as financial information, health data, and personally identifiable information, to provide a more comprehensive representation of the data encountered in real-world scenarios.

The proposed dataset, named DeSSI (**D**atas**e**t for **S**tructured **S**ensitive **I**nformation) [83] is a dataset created specifically for the purpose of developing and evaluating machine learning models for sensitive data detection in structured datasets. It contains over 31,000 database columns with 100 rows in a single database table while incorporating column headers. The dataset includes sensitive data types, such as Social Security Numbers (SSN), phone numbers, and email addresses, which are labeled as such, while all other non-sensitive information is labeled as "Other data". All the data types can be seen in Table 3.1.

The true labels are located in a separate file indicating which column of the original dataset file is of what label. A clip of the database table is shown in Figure 3.1.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
		email	lfncafwhfz	email_sid	onenumb	sgydcjmc	untry_nar	address	nin	odawvvau	ephonenu	zsxlbtncg	nggjwhzk	stomerem	lng	addressid	country	latitude	jhcirlnesv	tification_	acy_in_m	customer	lpgrkeonr
0		justin-zav	+43 (3117	bartol.sot	49324525	Silas	CUBA, AL	pavespor	ZZ 60365(	+137762	+36 21 12	05/03/20	32084971	n-s@hotr	110 Bump	Calle Dora	hn	28.97642	044 117 3	ZZ 60 82 (	915067.2	Wilson	11-17
1		mareblazi	476 617 8	egrgos0@	47161098	Keith	KANAWH	i-mocnik3	ZZ175276	+577764	+55 (64) 9	05/07/20	47910832	mw@hot	39.12143	805 Essex	LIBERTY, T	28.09122	257 266 0	ZZ 12796	372091.8	Davis	12-99
2		mia.jame	13152434	j.n@garci	+75 (24) 9	Alec	brazil	brysonco	ZZ705688	+607759	31 4112-8	25/09/20	21705348	jaxon-cler	-15.4207	7 Skyview	côte d'ivo	28.68195	832 830 4	ZZ 06096(	500459.8	India	08-73
3		jordyn.ga	+99 (229)	e.fish@ac	47131300	Pierce Alv	MARLOW	dt741@n	ZZ470726	+757754	(07700)9(	09/13/20	52967341	m.h8@sp	-12.9244	State Stre	ghana	39.41801	135 602 2	ZZ 25993	90161.2	Finnley G	11-73
4		l.y24@fisl	+73 0709	jm4@hot	(+940) 63	Ali	fr	e-k6@cro	ZZ 06521	+657706	(061) 314	08/10/20	96738526	ec@harve	-74.6186	Clairmont	SOUTHPC	38.96863	602 735 7	ZZ 19 24 7	520897.6	Julie	05-13
5		a.salamur	11 7393-9	k-adame@	032 856 3	Fici	WORCEST	kcastillo2	ZZ 80534(	+857785	02405126	05/08/20	57194632	astokes57	Franklin S	Via Carlo	BRN	39.40927	552 038 6	ZZ 19725(	34631.94	E VALDEZ	04-11
6		v-pesut8(	469/ 65 0	autumn.la	+14 400 9	Angie	Austria	amcallist	ZZ 16391	+637748	+90 21 84	08/05/20	48309567	autumn-n	-74.4026	Charles St	panama	39.46117	042 716 6	ZZ 33361	302513.4	Maddilyn	12-94
7		beaumcm	(693)487-	z-craven4	+14 (061)	Syed Cran	MONROE	p-kovacic	ZZ464408	+037719	13669195	15/10/20	53708491	m-s@ree	28.50009	10 Clyde /	sri lanka	28.10557	227 034 6	ZZ 45 69 9	525860.5	Zac Hayes	01-84
8		a-dj@zag	41 1764-3	jv@kligl.c	031 250 0	Nessa Dav	Croatia	pjames81	ZZ 99 17 9	+697788	+03 (783)	03/20/20	10286957	sc85@kn	Via Europ	North Cot	latvia	28.11203	183 443 2	ZZ611078	499608.9	Barbara	07-18
9		g.utjesinc	702.278.5	l.t825@vi	+2332417	Atticus Es	BALCH SP	harrietpal	ZZ877257	+977791	+55 77 92	07/13/20	24705618	dstojcic@	29.00036	Cobblestc	israel	28.40600	760 213 2	ZZ217085	876120.1	Violette A	07-00
10		emily.cha	+441734(	c-f72@sc	83145621	Kaylah Vil	new zeala	ibailey7@	ZZ129530	+507740	8O57198(	15/03/20	26741595	nwhitmor	-74.9265	Via Alfred	POL	28.15687	771 034 4	ZZ 30 24 1	585787.3	Barron M	07-72
11		c.w7590(	072.052.7	s.barr@gi	(249)119-	Fatoumat	ESKRIDGE	e-denhan	ZZ 06722(	+097723	001-608-8	07/30/20	28591763	z-stallings	-74.5274	Neugrabe	SOUTHPC	28.00680	011 768 4	ZZ 68662(	188685.1	Amin Tess	07-72
12		alara-nole	3122 303	tatjana.na	+15 (216(	Jazmin	norway	dmufic@s	ZZ624353	+547726	+55 (85) 9	01/13/20	42189075	zfabjan@	43 Dunca	4A Via Sa	kazakhsta	28.31383	528 820 3	ZZ870469	369641.1	Anayah G	01-71
13		oliver-bal	(037)295-	nolanscot	+53 (099)	Angelina	bulgaria	a-khan@y	ZZ 99 72 8	+977794	+51 61 56	03/10/20	81695302	rita-frlan	-16.3639	16/8 Via /	netherlan	28.36903	518 673 1	ZZ556298	76446.62	Daniel M	04-83
14		t-gonzale	+2335705	s-crawfor	+000 794	J L LOPEZ	CARSON (	lw6234@	ZZ332580	+817769	+2335542	05/07/20	15473285	a-t@med	39.53280	Tolend Ro	AUGUSTA	28.09596	886 031 5	ZZ 79 89 (	904723.5	Alivia	12-83
15		magdalen	01 91162	i.b739@h	83151324	Charlotte	CLARKSDA	thomas-c	ZZ741644	+847709	44216/ 36	11/25/20	74568915	as43@log	Kienburg	Wren Stre	pakistan	28.09866	430 116 8	ZZ497104	939694.7	Samuel	04-09
16		rafael.par	13102937	zdravko-c	81 6438-7	Christoph	my	b.r9693@	ZZ 58 35 2	+557748	21707733	21/05/20	42760519	bcastro@	Chalk Pon	24 Orchar	guatemal	28.33881	540 835 6	ZZ 43 88 9	923185.5	Seren Da	09-97
17		h-m26@r	54221281	leonardo-	+1-714-9(	Yarbrough	FAIRFIELD	b-h@jone	ZZ 51 74 8	+637718	(059)2018	09/06/20	48357125	br9549@	-75.0207	Sumner L	af	28.13120	532 502 8	ZZ 06838	508464.8	Owen	12-71
18		rj@patric	09843718	s-r117@g	+53 (7245	Toby	russia	e.b@gma	ZZ 55849	+247731	45299922	05/05/20	51243967	cmallory2	39.47922	Owl Stree	guatemal	39.48461	826 105 7	ZZ 41 26 7	78120.23	Whooley	01-82
19		kaizer.cor	4015765(	a-bienenf	851 5959	Morley	id	Calle Barr	ZZ 96 90 4	+707732	+2335503	11/03/20	45972836	tomas-do	Tices Lan	8 Radnor	HOXIE, KS	28.91654	031 448 2	ZZ 75 45 8	425911.2	Jonathan	01-97
20		anneliese	+4497144	mcacic45	(473) 527	Tyler Boli	PORT CHE	marino.b	ZZ371559	+717750	43149603	16/11/20	48503216	clay-book	-74.6095	Via del Pa	BRITTON,	39.09393	167 107 6	ZZ 41 09 2	653033.2	Popp	06-76
21		nklipa128	+14 (250)	r-ridley@	+92 (035)	Tyler-jam	SOUTH M	emmaatw	ZZ343108	+007727	+55 (82) 9	12/23/20	98240735	gavin.dou	27.98463	191 Oakla	DRAPER,	39.42513	672 088 1	ZZ497306	695453.8	Gentry Ed	03-11
22		r-robinso	+04 (014)	e.sy@ngu	9740 2 84	Alyse Woo	SHELBYVI	Winding E	ZZ 52 09 5	+437747	75161406	04/04/20	37259184	indiana.va	28 Sunflo	Hansonvil	EARLY, IA	28.63059	160 351 8	ZZ376349	378724.8	Castillo	02-78

Figure 3.1: An example of a part of the database, each row represents one instance, while each column is a potentially sensitive data type, and the column header can be anonymized or missing.

To ensure proper training and evaluation of machine learning models, the dataset was randomly split into training, validation, and test datasets in a ratio of 60/20/20 percent. The data is presented in the format of a comma-separated values (CSV) file and is available for public use on the Kaggle platform. With the creation of DeSSI, the real-world environment was simulated by incorporating data challenges such as missing or misleading column headers, while also ensuring that no sensitive information was put at risk through the use of pseudo-anonymized real-world data. Overall, DeSSI provides a valuable resource for researchers and practitioners in the field of sensitive data detection in structured datasets.

Each column is labeled with either one or multiple sensitive data types because, in many real-world datasets, it is common for columns to contain multiple types of sensitive data labels. For example, a single column may contain both names and addresses. In order to accurately represent this type of data, multi-label columns were created in the dataset.

Class	Description	Instance count
Other data	Everything that does not fall under any sensitive data type	6561
Phone number	Various supported formats of landline and mobile phone numbers	3547
Address	Multiple formats, can contain the street name, street number, and postal code	2966
Person	Name, Surname, can contain multiple of each or a combination	2319
Email	Email address	1533
NIN	National Identification Number	1706
Date	Various date formats	2319
Organization	Various types, supports extensions, i.e LLC, DD...	1607
GPE	Geopolitical entities such as states, cities, countries, etc.	2172
Geolocation	Longitude and Latitude	1635
SWIFT/BIC	Business identification codes for both financial and non-financial institutions	240
IBAN	International bank account number	213
Passport	Passport numbers	108
Religion	Religions and members of such organizations	93
CCN	Credit card numbers with all their supported formats	109
ID Card	Numbers of identification cards	68
Sexuality	Various types of sexualities	92
Gender	Various types of genders	94
Nationality	Nationalities based on countries	91
Race	Various descriptions of races	58

Table 3.1: Types of sensitive in the dataset. There also exist instances of combinations of classes.

In the dataset, the number of possible labels was limited to sensitive data types only (and 'Other data'). This decision was made because the ultimate goal of the machine learning task in the context of this work is to detect sensitive data and not all possible data types. This constraint results in a smaller number of labels compared to other similar datasets [81] [84] that contain general semantic types, not necessarily sensitive data. Due to this restriction, the number of possible labels in the DeSSI dataset is smaller and mostly easier to detect than the general-purpose labels present in the related work datasets. The restricted number of labels and

the less broad label types of sensitive data also lead to higher accuracy and, in general, better performance scores. Additionally, the dataset's focus on sensitive data types allows for more targeted development of machine learning models specific to detecting sensitive data, which is highly relevant in privacy-sensitive industries such as healthcare, finance, and government.

DeSSI is a significant contribution to the field of sensitive data detection in structured data, as it is the first widely available dataset to focus specifically on this problem. While other available datasets exist, they were created to solve a broader problem and are therefore less specific to sensitive data types. The specificity of DeSSI makes it a valuable resource for researchers and developers who want to train and test their models on real-world data that more closely resembles the type of data they will be working with within their applications.

While DeSSI is a valuable dataset, there is certainly room for improvement and expansion in the future. For example, the dataset could be expanded to include additional sensitive data types or include data from different geographic regions or cultural contexts. Additionally, the dataset could be modified to incorporate more challenging data scenarios, such as data with inconsistent or incomplete column headers, to better represent the challenges faced in real-world applications.

Despite these potential improvements, the use of DeSSI as a standard dataset for comparison in the field of sensitive data detection in structured data is encouraged. By using a standardized dataset, researchers and developers can more easily compare the performance of their models and techniques and gain insights into the strengths and weaknesses of different approaches. In this way, DeSSI can serve as a valuable resource for advancing the state-of-the-art in sensitive data detection in structured data.

Chapter 4

The Proposed Method

4.1 Introduction

Semantic column labeling and sensitive data detection are complex tasks that require a thorough understanding of natural language processing and machine learning techniques. Over the years, researchers have proposed various approaches to tackle these tasks, ranging from simple rule-based models to complex language models. After examining the existing literature and observing the recent trends, it became clear that an approach based solely on rules would not suffice and that the basis of the approach would need to be a machine-learning model. Such an approach can learn from the data and automatically discover the patterns and relationships between the features and labels that a rule-based approach cannot do on its own.

Furthermore, machine learning models are capable of handling large and complex datasets, making them suitable for tasks such as semantic column labeling and sensitive data detection. It also became clear that simpler machine learning and embedding models would not be sufficient for achieving satisfying performance in these tasks. It was apparent that a more advanced, active approach based on contextualized embeddings was necessary to capture the nuances of natural language present in the database cells.

As elaborated in Section 2.1.4, contextualized embeddings provide a way to represent words in the context of an entire sentence or paragraph, rather than just as standalone entities. This allows the model to capture complex relationships and meaning between words in the text, which is crucial for accurately detecting sensitive information in structured data. The research also indicated that an approach with actively created contextual embeddings has more potential to better capture the nuances of natural language than the passive, context-level embedding models.

The main idea of the proposed solution to the problem of structured sensitive data detection through the lens of NLP was inspired in part by TABBIE's [69] idea of using whole columns as context for BERT and by TaBERT's [68] idea to concatenate cells as input and give them

straight to BERT in their tokenized format, without creating embeddings for individual cells first, which is an approach depicted in Figure 2.1 under 2). This resulted in the removal of an intermediary step in the creation of the single-cell embedding. By removing that step, the features of the input are entirely preserved for BERT and its contextual embedding method. This is an active inclusion of context and is the main contribution of the proposed method with respect to related work.

The proposed method attempts to account for the nuances of natural language which often occur within cells, as table cells may contain multiple words, or in some cases, even multiple sentences. To this end, it utilizes BERT – the well-known language representation model that has extensive capabilities for the detection of relations between words. When testing the other existing models mentioned in the related work section, satisfactory results could not be obtained on the generated tabular data. Therefore, the focus was shifted to the approach that worked well on unstructured data, namely treating a column as a quasi-natural sentence and using context-sensitive transformers.

The driving idea behind this method was the assumption that cells in a column should be similar to each other in some way and that allowing the model to look at all cell values in a column at the same time might give more information than looking at each cell individually and then averaging. In a large portion of cases, one can also rely on the column header as an indication of what purpose these cells should have. However, it was decided against considering the column header as the most important feature, as column header naming conventions can include shorthands, or even be missing or misleading in real-world enterprise databases. The rest of this section will address the pipeline of the initially proposed method and some implementation details. The model is openly available for use on the GitHub platform [85].

4.2 Initialization and input

To take advantage of the aforementioned assumptions and conclusions, the model constructs the input to BERT from the column header together with multiple cell values from the same column, separated by delimiters, as shown in Figure 4.1 *. In such a way, BERT can take multiple cell values at the same time into account, thereby jointly incorporating both the context inside of a single cell, as well as the context between cell values.

*Licence - https://www.sciencedirect.com/science/article/pii/S0957417423004256

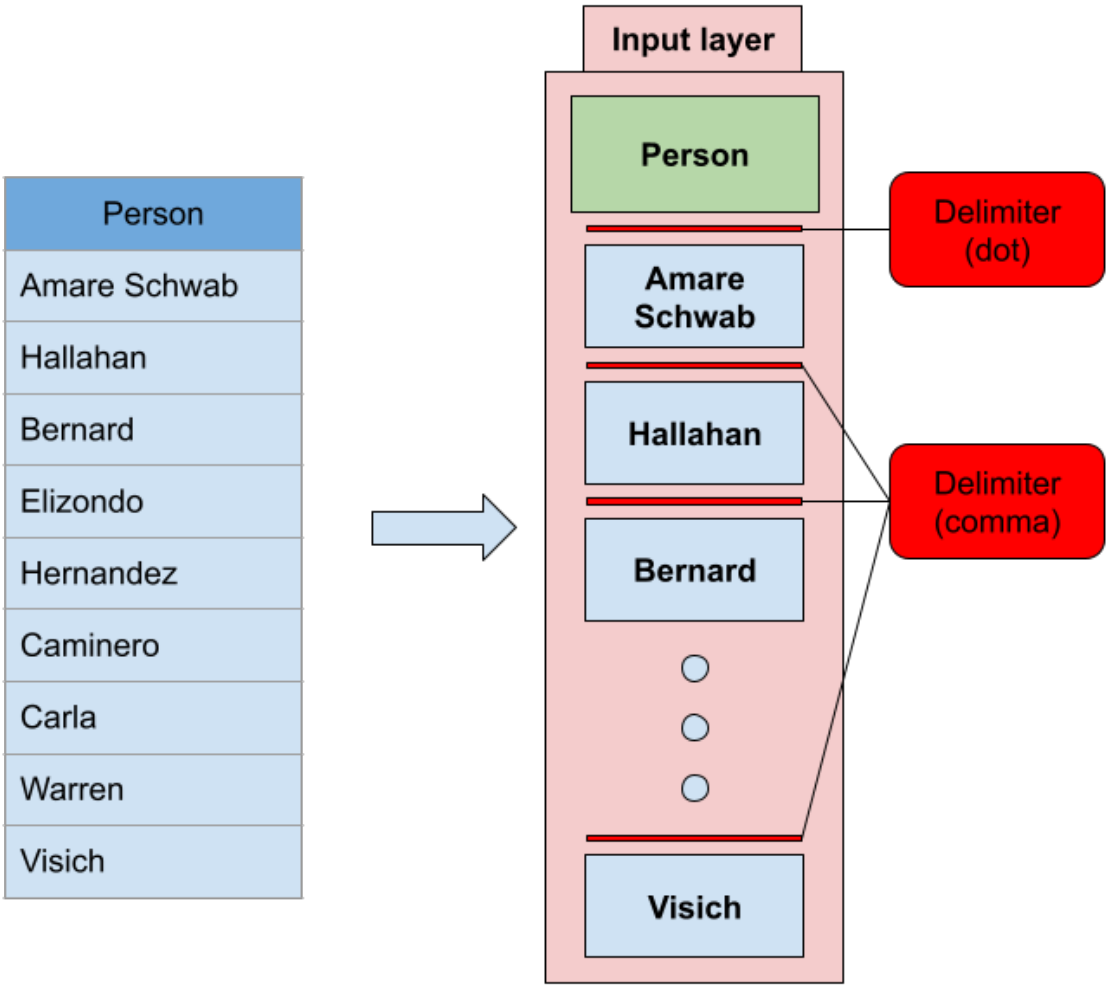


Figure 4.1: An example of a column turned into an input for the model. The values of cells are separated by a comma, while the name of the column is at the front, separated by a dot [70].

The proposed method for sensitive data detection in structured data sources using NLP leverages the strengths of contextualized embeddings and a unique input construction method to maximize the contextual understanding of the model. To achieve this, the model inputs are constructed from the column header and multiple cell values from the same column, separated by delimiters. This approach enables BERT to take multiple cell values into account at the same time, thereby capturing the context within a single cell as well as the context between cell values. The construction of the input is illustrated in Figure 4.1, where each column contains cell values that are concatenated together, separated by delimiters. This concatenated input is then tokenized and fed into BERT to generate contextual embeddings.

By using this input construction method, the model can better capture the nuances of natural language and the context within and between cells. This approach is different from other related work, which often focused on creating embeddings for individual cells or using whole columns as context for BERT. The advantage of using the proposed input construction method is that it preserves the features of the input for BERT's contextual embedding method which generates

contextual embeddings that are more accurate in representing the underlying meaning of the input. Furthermore, this approach reduces the complexity of the model and eliminates the need for an intermediary step of single-cell embedding creation.

During research, various types of delimiters were tested, such as semicolons, slashes, hyphens, and underscores, to separate the column headers and cell values. The experiments showed that the choice of delimiter did not significantly affect the performance of our proposed method. However, it was observed that using different delimiters to separate the column headers and cell values led to an increase in the model's performance. This is because the different delimiters provided additional information to the model, allowing it to better distinguish between the column headers and cell values. For example, a hyphen could be used to separate the column header and cell value, while an underscore could be used to separate different words within the cell value. This approach helped to make the input more interpretable and provided better context to the model, which ultimately improved its performance.

4.2.1 Concatenating columns

The limited number of tokens that can be entered into BERT is a significant challenge for the proposed method. In the default version of BERT, the maximum number of tokens in a sequence is limited to 512, and while alternative versions of BERT exist with larger input sizes, whatever the maximum input size is, this number can easily be exceeded when database cells contain multiple sentences or when tables contain a large number of rows. There are different ways to tackle this problem, but one common approach is to truncate the input to the maximum token length. However, this could result in a loss of valuable information, particularly when the values of cells are lengthy and contain dozens of tokens, as this would make the input consist of only a few cell values. Moreover, some of the multiple labels in the column might not be detected due to the truncated input.

To address these issues, the proposed method does not truncate the input but instead splits the information from the entire column into multiple inputs if the token count exceeds the maximum token count that the currently used version of BERT has. In this way, each column part is sent to BERT separately, and the BERT output is then averaged to produce a final prediction for the column. This technique enables the model to capture as much information as possible while avoiding the truncation of valuable data. Additionally, it increases the chances of detecting multiple labels in the column, which is particularly important for sensitive data detection.

Although the input splitting approach may not be entirely in line with the initial idea of active context, it still provides a reasonable trade-off between active context and the need to handle long cell values and tables with a large number of rows. In this way, the model can take a larger input and still preserve the valuable information in the input, ensuring that multiple labels in the column are not missed due to truncation.

4.2.2 Data processing

To expand on the given prompt, after applying the proposed approach and constructing the input to BERT, the model employs BERT on a batch of column embeddings. BERT's decoder then produces a non-normalized prediction (logit) for each label, indicating the probability of the column containing that particular sensitive data type. These logits are averaged over all column parts that the column was separated into providing a single set of averaged logits for the whole column. The averages are created in a weighted manner, taking into account how many rows were present inside of each column part.

After the averaged logits are calculated, a sigmoid function is applied to each of the logits to produce normalized probabilities for each class individually. A sigmoid function is used instead of a softmax function because the task is to find multiple labels if they are present. The use of a softmax function might cause the majority presence of one label to diminish the presence of another label. In contrast, a sigmoid function considers all labels separately from each other, and the probability of each label is calculated independently. The functions are depicted in formulas 4.1 and 4.2.

Sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{4.1}$$

Softmax function:

$$f_i(x) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{4.2}$$

Once the probabilities for each label are calculated, a threshold value is set to classify a column as containing a sensitive data type or not. The threshold value is determined using the validation set and is chosen to optimize the F1-score. If the probability of a label for a given column exceeds the threshold, that column is classified as containing the sensitive data type associated with that label. Otherwise, it is classified as not containing the sensitive data type.

After the probabilities for each label are obtained through the sigmoid function, they need to be compared to a predefined threshold value to determine if a label is present or not. This threshold value is determined based on the desired trade-off between precision and recall of the model. A higher threshold will result in higher precision, but lower recall, meaning that the model will correctly identify fewer instances of a label, but the ones it does identify are more likely to be correct. Conversely, a lower threshold will result in higher recall, but lower precision, meaning that the model will correctly identify more instances of a label, but there will be more false positives. Finding the optimal threshold is a key step in training a sensitive data detection model.

4.2.3 Data post-processing

As previously mentioned, while rule-based approaches lack the possibility of natural language understanding, they can be of significant use for the detection of certain data types. Specifically, sensitive data types that have a specific and strict form that is known beforehand, or when sensitive data only occurs in a subset of values, most of which are known, are suitable for rule-based approaches.

While machine learning methods are efficient for general sensitive data type detection purposes, they usually struggle with very specific cases, such as deciding if a 16-digit cell value is a social security, credit card, telephone, or some other kind of number. The differences among these sensitive data types are subtle and it can be very hard even for a human to discern them [86].

The proposed approach for sensitive data detection in structured data, in addition to the machine learning base part, utilizes rule-based methods in post-processing for certain sensitive data types. While the machine learning model can accurately detect and classify many sensitive data types, there are certain data types that require a more specific and strict approach. The regular expression approach is effective for sensitive data types such as emails, SSNs, or credit card numbers because these data types have specific formats that are well-established and known beforehand. Regular expressions are designed to match a specific pattern of characters, and instances of false positives are rare, if they occur at all. For instance, a regular expression can be used to match the 16-digit pattern of a credit card number or the format of an email address.

For sensitive data types that belong to a subset of possible values, the method uses lookup tables. Classes such as gender, nationality, religion, ethnicity, and other categories have a finite and well-defined set of options. The method uses lookup tables to compare the cell value to the set of options and identify the corresponding label. This is effective because all possible values are known in advance and can be listed in the lookup table. From the method's architectural standpoint, in addition to the main flow of classification through BERT, a side flow is created. In the side flow, each input column is passed through rule-based methods that produce a probability for the column to be of that label by looking at the ratio of the number of detected labels inside of the column and the total amount of cells in the column. The generated ratio is treated as a probability of the whole column representing that label type. After the main flow of the model classifies the column into one or multiple classes, the side flow then compares the probabilities it generated and also compares them to the threshold, altering the classification if it exceeds the threshold.

The depiction of the whole method is shown in Figure 4.2 [†].

[†]Licence - https://www.sciencedirect.com/science/article/pii/S0957417423004256

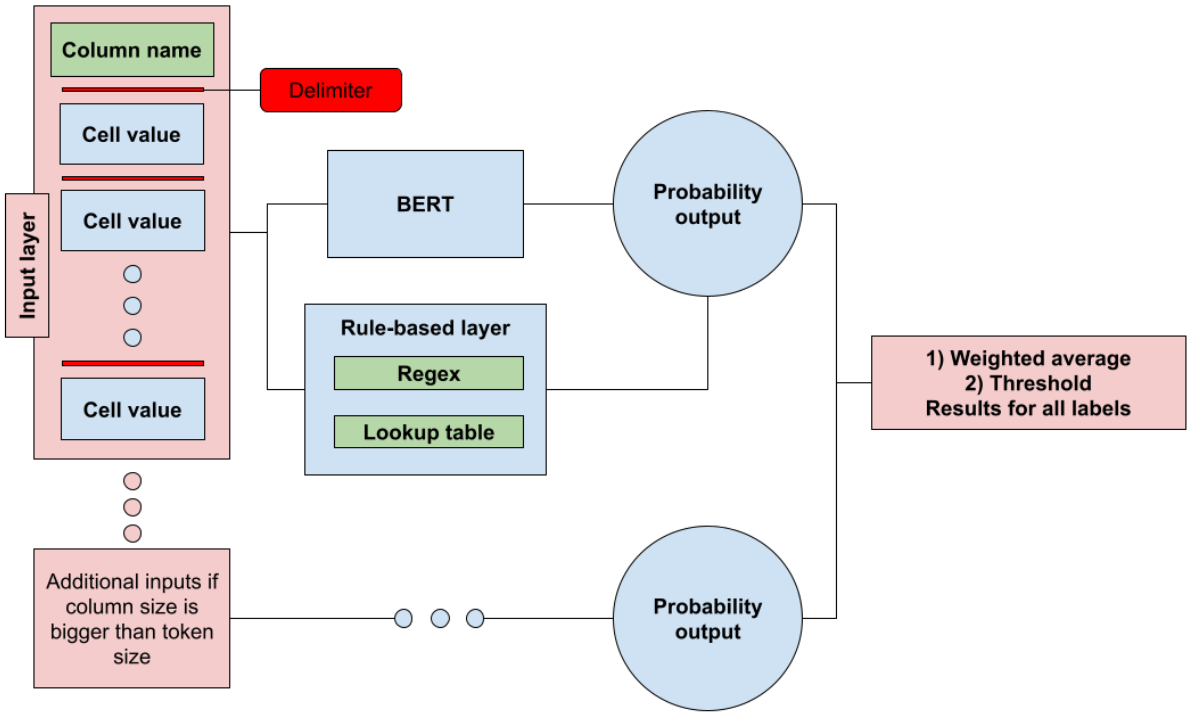


Figure 4.2: Method overview – the column is represented as an input and forwarded to both BERT and a rule-based layer separately, after which the probabilities for each sensitive data type are generated and creating a weighted average over all the column parts. If the averages exceed a certain threshold, then the column is classified as the according type, allowing multiple labels [70]

4.2.4 The baseline model

Before developing the proposed method, traditional methods for structured data were initially explored in the quest to tackle the problem of sensitive data detection. However, the results obtained from these conventional methods did not meet expectations and did not provide satisfactory performance. This realization led to the exploration of more unconventional and innovative approaches in order to improve the performance and effectiveness of the solution. This approach enabled establishing a performance benchmark against which one could measure the efficacy of the proposed method.

This benchmark, the baseline model, is a machine learning model that uses simpler conventional methods for NLP which do not consider or include possible contextual relations inside or between cells. This model was implemented separately and serves as a reference point to show how the inclusion of context improves classification results. Initially, the method creates word embeddings using GloVe [30] for each word. The embeddings are then averaged over the whole cell value. Aside from the word embeddings, it also creates character embeddings using a one-dimensional CNN, as recommended in [20]. The character embeddings are concatenated to the averaged word embeddings.

Once the final embedding is formed, it is fed into a fully connected neural network. The

neural network utilizes the learned features from the embeddings to predict the label of the cell. The output logits from the neural network are passed through a sigmoid function to obtain normalized probabilities for each label independently. In order to derive the final classification result, the predictions from all individual cells within a column are taken into consideration and averaged. This aggregation ensures a comprehensive evaluation of the column as a whole, reflecting the collective information provided by the cells.

By comparing the performance of the proposed method against the results obtained from the baseline model, it is possible to assess the significance of incorporating contextual information and highlight the improved classification accuracy achieved through the novel approach. Experiments and comparisons with related work, alternative approaches, and the baseline model will be shown in the following chapters.

Chapter 5

Alternative approaches

5.1 Motivation

The hyperparameters of a machine learning model play a crucial role in determining its performance and behavior. Various approaches can be adopted to alter and experiment with their values, leading to different outcomes and results. While tuning hyperparameters is a primary focus of the next chapter, the current chapter introduces a different perspective by proposing architectural changes to the model.

Architectural changes involve modifying the fundamental structure and design of the model, going beyond just adjusting the hyperparameters. These changes not only impact the values of the hyperparameters but also alter the underlying principles and ideas driving the solution process. By exploring different architectural approaches, we aim to enhance the overall effectiveness and capabilities of the model in addressing the specific challenges of sensitive data detection.

The alternative approaches include some simpler and some more complex architectures. The first approach is similar to the initially proposed approach but without column concatenation, the second and third approaches use also the adjacent columns to the currently observed column, where the second approach pre-generates the embeddings of the adjacent columns statically while the third approach uses them actively.

By experimenting with alternative approaches and evaluating their performance, it is possible to assess the impact and determine the trade-off between computational complexity and enhanced or reduced contextual understanding.

5.2 No column concatenation

The first alternative approach explored here is a simplified method that involves using a limited number of input cells, which are truncated to fit within the maximum token count limit. The

approach allows for selecting input cells either in their original order or at random. The aim of this approach is to streamline the model and simplify the execution process. The simplified model architecture is shown in Figure 5.1.

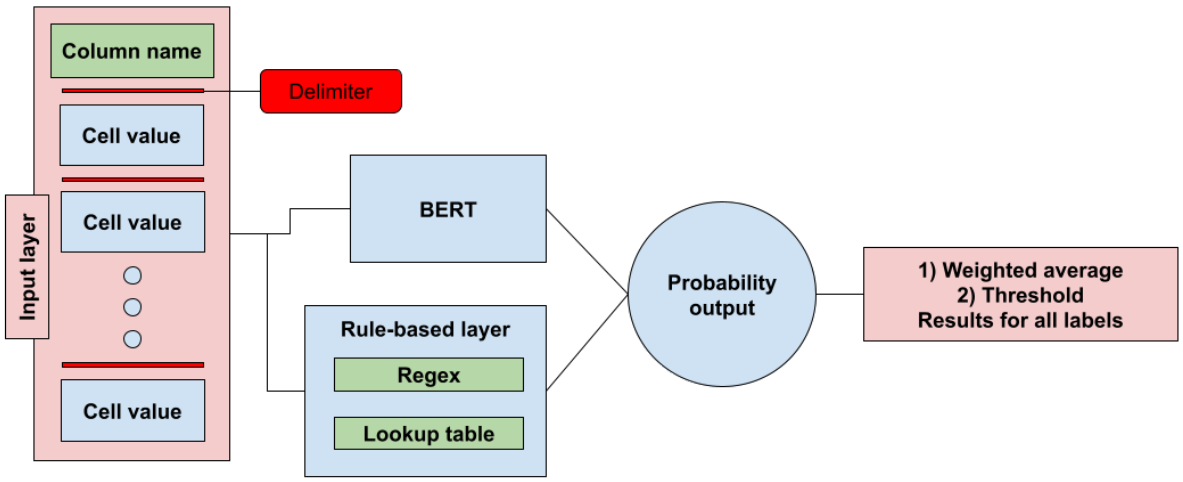


Figure 5.1: Method overview – the column is represented as input and forwarded to both BERT and a rule-based layer separately, after which the probabilities for each sensitive data type are generated. If the averages exceed a certain threshold, then the column is classified as the according type, allowing multiple labels.

Compared to the main approach that involves averaging multiple inputs, this alternative method significantly reduces execution time. With only one input created for each column, the number of representations generated and forward passes through the model are greatly minimized. Consequently, the computational burden is alleviated, enabling faster inference and reducing overall processing time.

This simplified approach also serves as a baseline for evaluating the effectiveness of using multiple inputs for a single column. By utilizing only a subset of data within the column, there is a possibility of not capturing certain sensitive data types, particularly in cases where the column has multiple labels.

5.3 Adjacent columns

While the main proposed method actively considers the values within the current column, it does not incorporate information from adjacent columns. However, there is a potential benefit

in leveraging as much context as possible, including the values from neighboring columns. This broader context can provide additional insights and enhance the model's understanding of the relationships between columns.

Despite this, incorporating information from adjacent columns comes with certain challenges. It significantly increases the computational complexity of the model and introduces the risk of cluttering the model with potentially irrelevant features. Furthermore, the number of columns in a database table can vary greatly, making it difficult to consistently train the model on variable column counts. To address these challenges, alternative approaches are explored that consider a fixed number of adjacent columns. Specifically, experiments are conducted with models that incorporate the values from 1 or 2 columns prior to and after the current column. By limiting the scope of adjacent columns, the evaluation strikes a balance between capturing contextual information and managing computational complexity.

These alternative approaches are based on the assumption that there could be some relationship or correlation between the current column and its adjacent columns, which could enhance the understanding of the values within the current column. Yet, it is important to note that this assumption may not hold true in all datasets, including the proposed dataset. Namely, in the case of the proposed dataset, where the order of the columns was randomly generated, the adjacent columns may not necessarily provide meaningful context or information directly related to the current column. Therefore, the assumption of the relationship between adjacent columns and the current column may not be applicable.

Nevertheless, despite the potential mismatch in the relationship between adjacent columns and the current column in our specific dataset, these alternative approaches still offer valuable perspectives and possibilities. They present different strategies that can be explored and tested in various tasks and datasets where there may be inherent relationships between adjacent columns or specific contextual dependencies. While the order of columns in a database table is arbitrary, the assumption was made that if some columns in the database have relevant context for the currently looked-at column and we cannot inherently know in which position relevant to the current column they would be, taking the adjacent column would be the best guess.

5.3.1 Static representations of adjacent columns

The second alternative method takes a different approach to address the incorporation of adjacent column information. Similar to the initially proposed approach, it begins by creating independent representations of each column through the embedding process. However, it goes a step further by expanding the neural network that follows the embedding creation process to incorporate the features from the adjacent columns as shown in Figure 5.2.

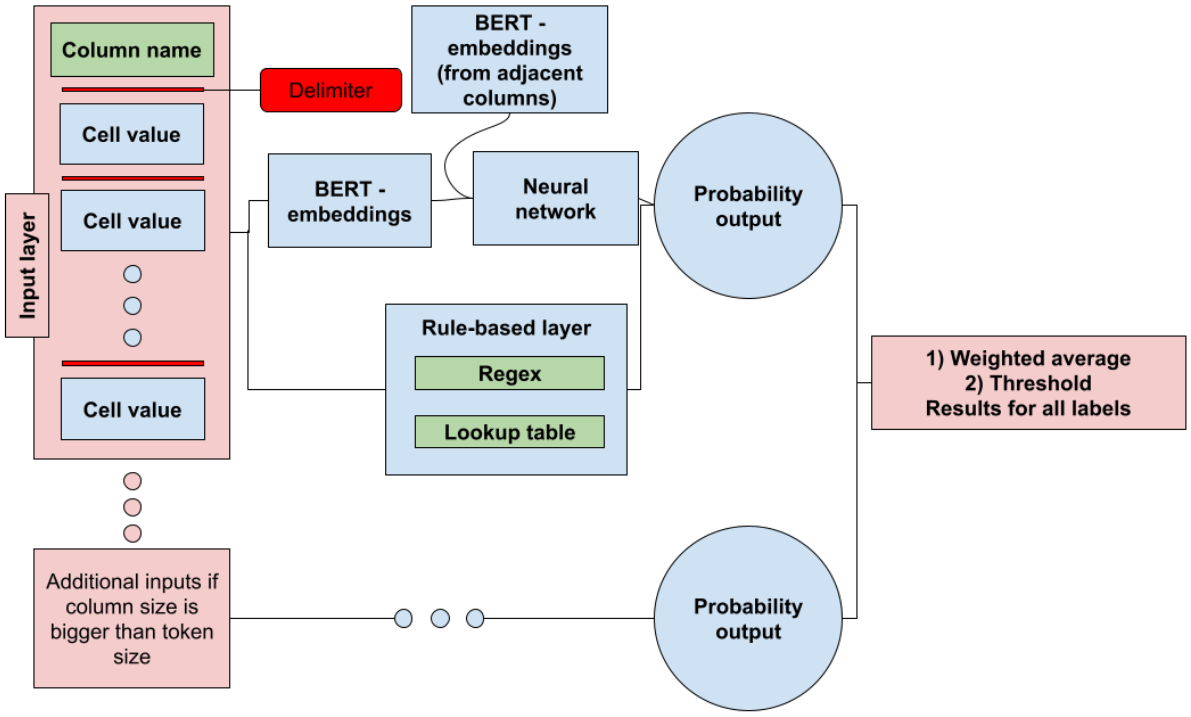


Figure 5.2: Static adjacent columns method overview – the column is represented as input and forwarded to BERT-s embedding model rule-based layer separately. The created embeddings are passed to the downstream neural network of BERT together with pre-generated embeddings from adjacent columns, after which the probabilities for each sensitive data type are generated. If the averages exceed a certain threshold, then the column is classified as the according type, allowing multiple labels.

This approach recognizes the importance of leveraging context from neighboring columns without compromising the information captured by the embedding method. By expanding the neural network architecture, the model can utilize the additional features from adjacent columns to potentially provide a richer context for sensitive data detection.

It is important to note that this expansion in architecture does come at the cost of increased computational complexity. The increased computational complexity comes from the fact that embeddings from adjacent columns need to be given as an input as well as the embeddings of the current column, which results in a larger neural network which in itself has more parameters because it has more inputs. The complexity is increased by a factor of how many adjacent columns are added. If only one column before and one after are added then the complexity increases by a factor of 3, and if 2 columns before and 2 columns after are added then by a factor of 5. However, the benefit of incorporating more context could possibly justify this additional complexity. By considering the values of adjacent columns, the model could gain a deeper understanding of the relationships and patterns within the dataset, potentially leading to improved performance in sensitive data detection.

5.3.2 Active representation of adjacent columns

The third alternative approach takes a different approach to incorporate the context from adjacent columns while maintaining the active inclusion of a single column. Instead of expanding the neural network architecture or increasing computational complexity, this method focuses on modifying the input method.

In this approach, the number of tokens from the current column is reduced proportionally to the amount of other column data that is added, thereby creating space for the inclusion of tokens from adjacent columns within the same neural network size as the initial approach. The same amount of input rows will be taken from all columns, which means that the amount of tokens from each column changes dynamically based on how many tokens are in each cell. By replacing some of the tokens from the current column with tokens from adjacent columns, the model can capture the context and information from neighboring columns without the need for additional computational resources.

The created input is divided into several parts, each part representing a whole or part of a column from the database table. Each part is basically a smaller input size from the original input creation method in Figure 4.1, each separated by an additional delimiter. Figure 5.3 depicts such an input, in this example consisting of three parts. The process ultimately makes use of three separate delimiter types. The first one is for the separation of column headers and cell values, the second one is for the separation of cell values from each other, and the third is for the separation of columns.

NIN	Person	Born
032206132	Hallahan	02/29/2004
528926767	Amare Schwab	07/21/1976
222288986	Bernard	03/21/1981
574156803	Elizondo	02/12/1970
665182619	Hernandez	07/01/1986
762054976	Caminero	08/06/1959
533401577	Carla	01/28/1979
482660817	Warren	09/03/1981
532037567	Visich	05/08/1961

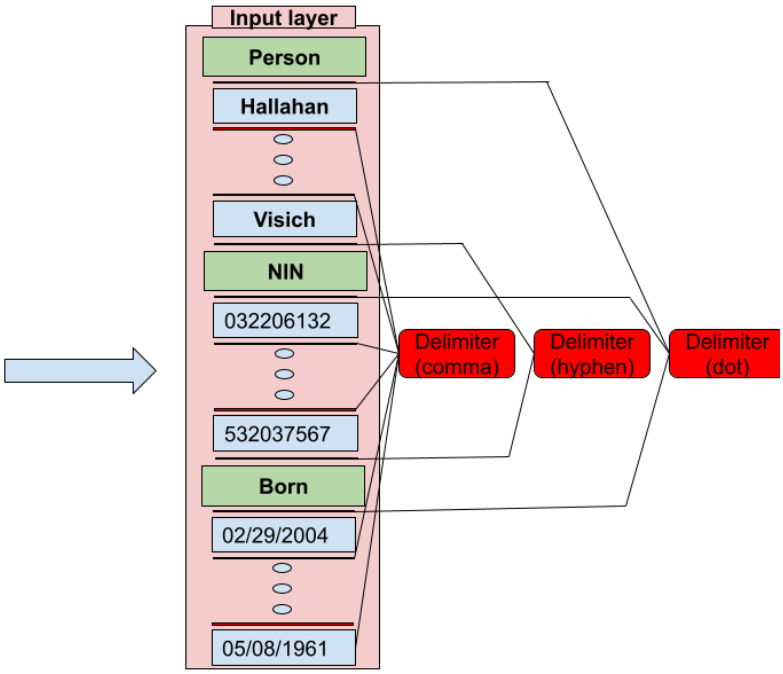


Figure 5.3: Input with adjacent columns: the input consists of tokens created from the column headers, and cell values of the current column and adjacent columns. The currently represented column is the middle one with the column header "Person". The method uses three types of delimiters, namely the first one for the separation of column headers and cell values, the second one for the separation of cell values from each other, and the third one to separate columns.

It is important to note that, while this approach does not directly increase computational complexity, it does have implications when considering the number of tokens from each column to be used. If the same number of tokens from each column as in the initial approach is desired, multiple inputs need to be created and averaged, which increases computational complexity proportionally.

While using the rule-based methods, it is important to only forward the tokens from the current column, since the adjacent columns could have misleading information for such a simpler and more strict model that does not have the capabilities to learn through iteration that those adjacent columns are only there for help and do not represent the actual input. The overview of the whole approach is represented in Figure 5.4.

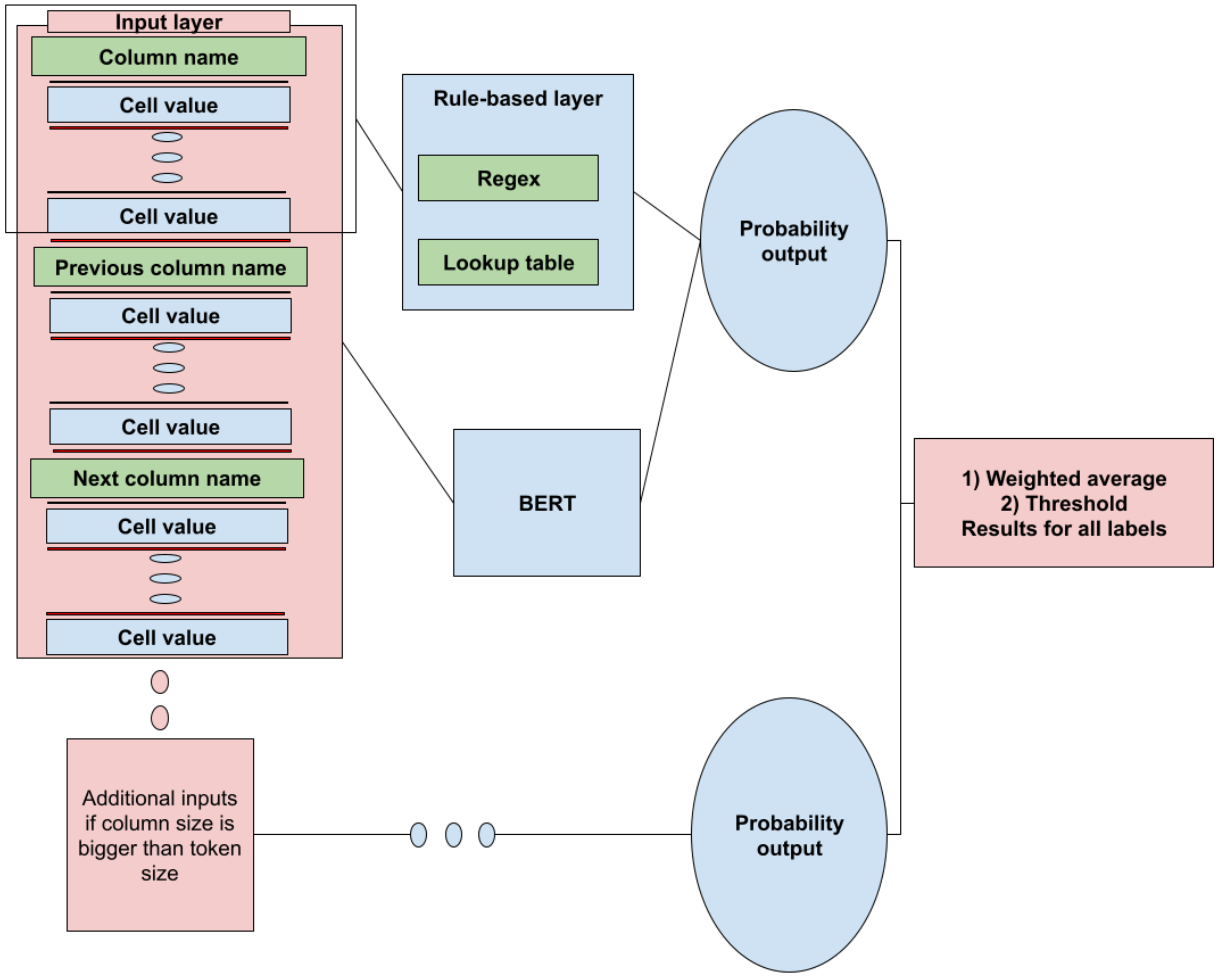


Figure 5.4: Active adjacent columns method overview – the current column and its neighboring columns are represented as input and forwarded to BERT, while only the representation part of the current column is forwarded to the rule-based layer. Thereafter, the probabilities for each sensitive data type are generated. If the averages exceed a certain threshold, then the column is classified as the according type, allowing multiple labels.

Chapter 6

Experiments

6.1 Calculation

The models created as part of this dissertation were developed using the FLAIR [87] library, a Pytorch-based framework for NLP. The models made use of a pre-trained distilled and uncased version of BERT [88] together with the AdamW optimizer [89]. Several configurations of learning rates, amount of epochs, and batch sizes were tested and experimented with. The standard values used for the comparison of the developed models to models of related work were a learning rate of $5 \cdot 10^{-5}$, 10 epochs, and a mini-batch size of 16 on an NVIDIA RTX 3090 GPU. The training process on the proposed dataset with one set of hyperparameters took close to 5 hours.

6.2 Evaluation metrics

To assess the effectiveness of our model, F1 score was used [90], which is calculated from the confusion matrix. With multiple possible classes for column labeling, the F1 score provides a comprehensive evaluation metric that takes into account both precision and recall.

In the case of detecting sensitive data, the emphasis is placed on recall rather than precision for all labels except 'Other data'. This distinction arises from the potential consequences associated with false negatives and false positives. A false negative occurs when the model fails to identify sensitive data, leading to subsequent tasks not adequately removing or de-identifying that sensitive information. Such false negatives can have serious implications for data privacy and security, as sensitive data may remain exposed. Thus, it is crucial to prioritize recall to minimize the occurrence of false negatives and ensure proper handling of sensitive data.

On the other hand, false positives, where the model erroneously detects non-sensitive data as sensitive, are considered less detrimental in the context of sensitive data detection. While false positives may result in the unnecessary removal or de-identification of data in downstream tasks,

the potential harm is relatively lower compared to false negatives. It is generally preferable to err on the side of caution and flag more data as potentially sensitive, rather than risk exposing sensitive information.

For semantic column labeling, where the objective is to assign appropriate labels to columns based on their semantics, the importance of recall and precision is relatively balanced. In this scenario, the F1 score, which combines both recall and precision, is considered the most suitable metric for evaluating the overall performance of the model. By considering the specific requirements of each task, such as sensitive data detection and semantic column labeling, and understanding the implications of false negatives and false positives, it is possible to select the appropriate evaluation metrics to effectively assess the performance of the model and ensure the accurate identification and labeling of data [91].

In both the related work and this thesis, three different types of measures are utilized: micro, macro, and weighted averages [66] [14] [65]. These measures determine how the overall performance is calculated by considering the results across all samples and classes.

In the micro average approach, each individual example contributes equally to the final result. This means that the performance metrics, such as precision, recall, and F1 score, are calculated by aggregating the counts of true positives, false positives, and false negatives across all samples. The micro average treats each sample with equal importance, regardless of the class to which it belongs.

In contrast, the macro average approach calculates the class results independently. Instead of considering the contribution of each individual sample, the macro average treats each class as equally important. This means that each class contributes the same weight to the final result, irrespective of the size of the class. The performance metrics for each class are calculated independently and then averaged to obtain the overall macro average result.

Lastly, the weighted average approach takes into account both the individual class results and the class sizes. Similar to the macro average, the class results are calculated independently. However, in the weighted average, the contribution of each class to the final result is proportional to the number of examples belonging to that class. This means that larger classes have a greater impact on the weighted average result compared to smaller classes.

6.3 Comparisons

In this section, a series of experiments were conducted to evaluate the performance of the proposed method and compare it with alternative approaches and models from related work. The experimental analysis is presented in five distinct subsections, each focusing on specific aspects of the evaluation.

The first subsection investigates the performance of the initially proposed model by sys-

tematically varying its hyperparameters and examining the impact of different assumptions. This comprehensive exploration allows for gaining insights into the sensitivity of the model to various settings and assessing its robustness. By analyzing the results obtained through these experiments, the optimal configuration can be determined and the effectiveness of the proposed method can be validated. All results are displayed based on the weighted F1 measure, explained above.

Moving to the second subsection, a comparative analysis is performed to evaluate the performance of the initially proposed method against alternative approaches. This comparison aims to assess whether the proposed approach outperforms or offers advantages over the other developed methods. Thereby, it is possible to highlight the strengths and weaknesses of each approach, providing a comprehensive understanding of their relative performance.

In the third subsection, the best-performing model is further evaluated and compared to other models from related work on datasets from related work, which tackle the problem of semantic column labeling. This comparison allows one to assess the generalization capability and effectiveness of the proposed model on different datasets and determine its performance in comparison to the state-of-the-art approaches. By analyzing the results in this context, it is possible to gain insights into the strengths and limitations of the proposed method in relation to the other existing solutions.

The fourth subsection focuses on evaluating the proposed method's performance on the proposed dataset, specifically designed for sensitive data detection. By comparing the proposed model to models from related work on this dataset, it is possible to assess its ability to accurately detect sensitive data types and identify its strengths in addressing the unique challenges posed by our dataset. This comparison provides valuable insights into the effectiveness of the method in handling sensitive data detection tasks.

The fifth subsection brings the rule-based system into place and uses real-world datasets for testing and comparison.

Overall, these experimental subsections provide a comprehensive evaluation of the proposed method, its performance in comparison to alternative approaches, and its generalizability across different datasets. Through rigorous experimentation and analysis, the aim is to validate the effectiveness of the proposed method and contribute to the advancement of the field of semantic column labeling and sensitive data detection.

6.3.1 Hyperparameters and assumptions

While the testing of individual hyperparameters has been conducted, it is important to acknowledge that machine learning model hyperparameters are often interconnected and can exhibit complex interactions. Modifying one hyperparameter may have an impact on the behavior and performance of other hyperparameters. However, due to resource limitations and the compu-

tational costs associated with conducting an extensive n-dimensional grid search, it was not feasible to explore the entire hyperparameter space comprehensively.

The process of tuning hyperparameters involves finding the optimal combination of hyperparameter values that maximize the model's performance [92]. In an ideal scenario, a grid search or randomized search can be performed to systematically explore various hyperparameter combinations. However, such an exhaustive search becomes increasingly challenging as the number of hyperparameters and their possible values increases. It requires significant computational resources and time.

In situations where resource constraints prohibit an exhaustive search, it becomes necessary to make informed decisions and prioritize certain parameters based on their expected impact. Domain knowledge, prior experience, and insights from preliminary experiments can help guide the selection of hyperparameters for further investigation. It is crucial to focus on hyperparameters that are likely to have the most substantial impact on the model's performance and prioritize their optimization.

Moreover, techniques such as manual tuning, intuition-driven adjustments, and leveraging insights from related work can provide valuable guidance in narrowing down the hyperparameter search space. Careful consideration of hyperparameter interactions and their potential implications is necessary when making adjustments based on limited testing.

While the lack of a comprehensive exploration of all hyperparameter interactions is a limitation of this work, the constrained hyperparameter testing approach aims to strike a balance between computational efficiency and achieving satisfactory performance. By focusing on key parameters and making informed adjustments, it is possible to identify reasonably effective hyperparameter configurations within the given constraints. It is important to acknowledge the limitations and potential for further improvement when interpreting the results obtained from hyperparameter testing in the context of the overall model performance. In continuation of this subsection, the influence of individual hyperparameters on the proposed method is explored.

The experiments were conducted so that all hyperparameters besides the currently looked-at hyperparameter were locked at a predefined value. The predefined values are arbitrarily chosen educated guesses. The chosen values are a Learning rate of 0.5, a batch size of 16, delimiters dot and comma, a number of inputs of 5, and a multi-label threshold of 0.1.

Learning rate

The learning rate is a hyperparameter used by machine learning models to determine the rate at which the internal parameters, i.e., weights between neurons, of the model change [24]. It plays a crucial role in the optimization process, as it controls the step size taken during parameter updates.

In the proposed approach, the Adam optimizer [93] is utilized, which dynamically adjusts

the learning rate based on past gradients. This adaptive learning rate allows the model to effectively navigate the optimization landscape and make appropriate adjustments to the parameters. The Adam optimizer considers both the first-order moments (the average of the gradients) and the second-order moments (the average of the squared gradients) to calculate the learning rate for each parameter. However, while the Adam optimizer can handle the learning rate adaptation during training, it is still important to determine a suitable initial learning rate. If the initial learning rate is too low, the model may converge slowly and get stuck in local optima, preventing it from finding better solutions. On the other hand, if the initial learning rate is too high, the model may experience large fluctuations in the parameter updates, leading to instability and hindered convergence.

To determine the right balance for the starting learning rate, experimentation and the observance of the model's behavior were tested. The results are shown in 6.1. The model performed decently with several different learning rates, showing significantly worse performance only when drastically decreasing or increasing the learning rate while achieving the best solution at the learning rate of 0.4.

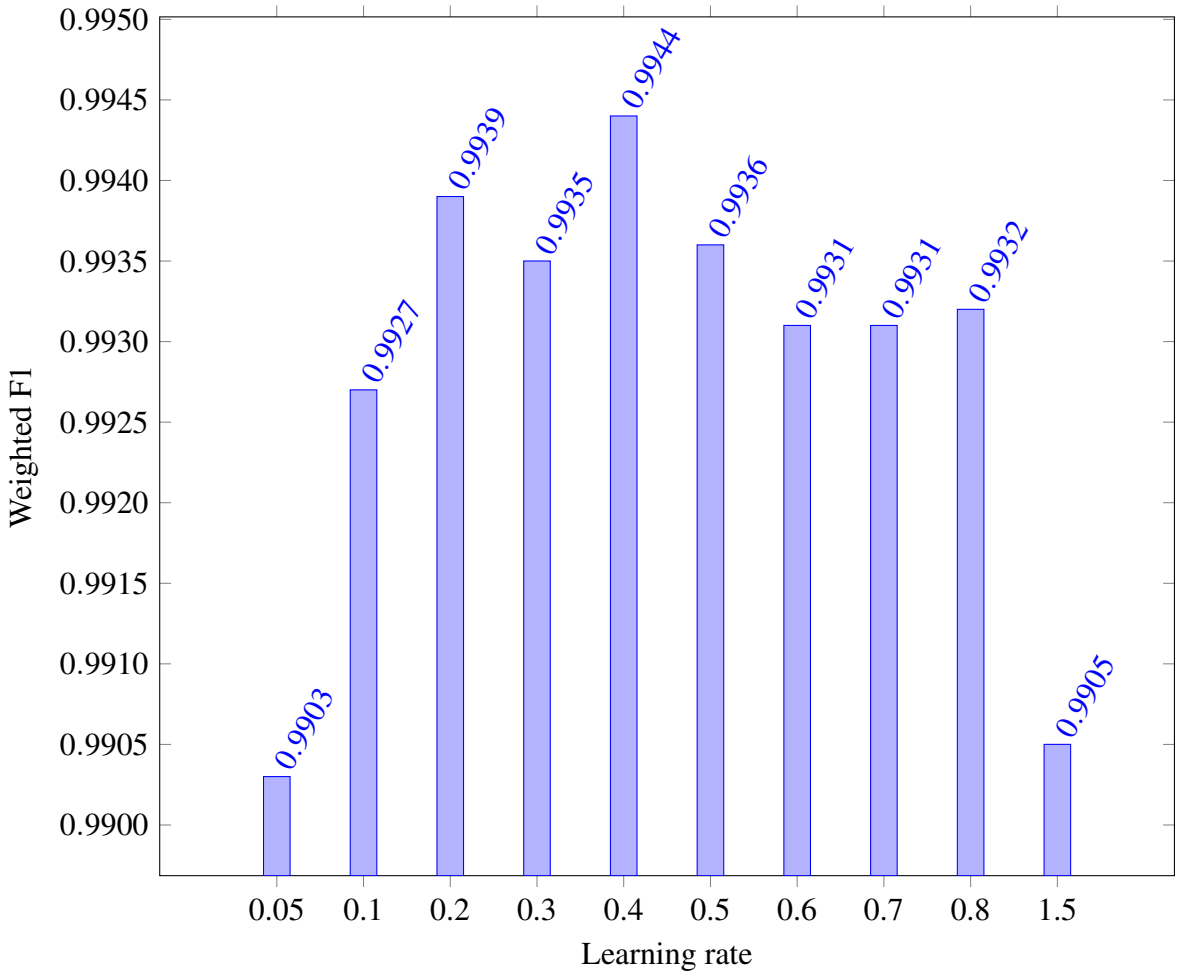


Figure 6.1: Learning rates of the proposed method on the proposed dataset with other hyperparameters standardized

Number of epochs

The number of epochs is a crucial hyperparameter in training machine learning models as it determines the number of times the model iterates over the training and validation datasets. Selecting an appropriate number of epochs is essential for achieving optimal model performance. If the number of epochs is too small, the model may not have enough iterations to converge to the optimal solution and may underfit the data. On the other hand, if the number of epochs is too large, the model may start to overfit the training data, resulting in poor generalization to unseen data.

To address this challenge, the proposed method incorporates the technique of early stopping [94]. Early stopping monitors the performance of the model on a separate validation set during training. If the performance on the validation set fails to improve or stagnates for a predefined number of consecutive epochs, the training process is halted. This mechanism allows for automatically determining the optimal number of epochs without the need for exhaustive testing of different epoch values.

Early stopping effectively prevents overfitting and avoids the redundant testing of various epoch values. The model is trained until the point where it achieves the best performance on the validation set, as indicated by a significant improvement in the evaluation metric or the absence of further improvement within a certain threshold. This approach strikes a balance between training the model for a sufficient number of epochs to capture meaningful patterns in the data and preventing it from overfitting by terminating training when performance plateaus. By incorporating early stopping, the proposed method streamlines the process of determining the appropriate number of epochs, ensuring that the model achieves the best possible performance while minimizing the risk of overfitting or underfitting.

Batch size

The batch size is an important hyperparameter [95] in the training process of machine learning models, as it determines the number of samples processed before updating the internal model parameters. It plays a significant role in the efficiency, accuracy, and convergence of the model during training. When using a larger batch size, more samples are processed simultaneously before updating the model's parameters. This results in more accurate estimates of the gradients, as the gradients are computed based on a larger number of samples. With accurate gradients, the updates to the model's parameters are more precise and aligned with the underlying patterns in the data. This can lead to faster convergence and improved accuracy of the model. On the other hand, when using a smaller batch size, fewer samples are processed at a time, resulting in more frequent updates of the model's parameters. This can allow the model to make smaller,

incremental adjustments and explore the parameter space in finer detail. However, the estimates of the gradients with smaller batch sizes can be noisier and less accurate compared to larger batch sizes. As a result, the convergence of the model may be slower, and the updates to the parameters may exhibit more fluctuations.

The choice of batch size depends on various factors [96] [97], including the available computational resources, the size of the dataset, and the complexity of the model. Larger batch sizes are generally preferred when computational resources allow, as they tend to provide more stable and accurate updates to the model's parameters. However, using very large batch sizes may consume excessive memory and computational power, making it impractical for certain setups. Conversely, smaller batch sizes are useful in scenarios where memory is limited or when dealing with large datasets. They can also be beneficial when the training data exhibits significant variations or when the model has a high capacity and benefits from more frequent updates. It is worth noting that the choice of batch size is a trade-off between computational efficiency and the quality of parameter updates. It is often necessary to experiment with different batch sizes to find the optimal balance that leads to good model performance.

The performance of the proposed method with varying batch sizes can be seen in Figure 6.2. The performance of the model exhibits a consistent upward trend as the batch size increases, reaching its peak at a batch size of 32. Due to resource limitations, it was not possible to test the model with a batch size of 64 as it would consume excessive memory on the available graphics card. However, in an attempt to explore an intermediate batch size, the model's performance was evaluated with a batch size of 48, only to observe a noticeable decline in performance. This suggests that 32 emerges as the optimal batch size for the given problem, striking a balance between computational efficiency and model effectiveness.

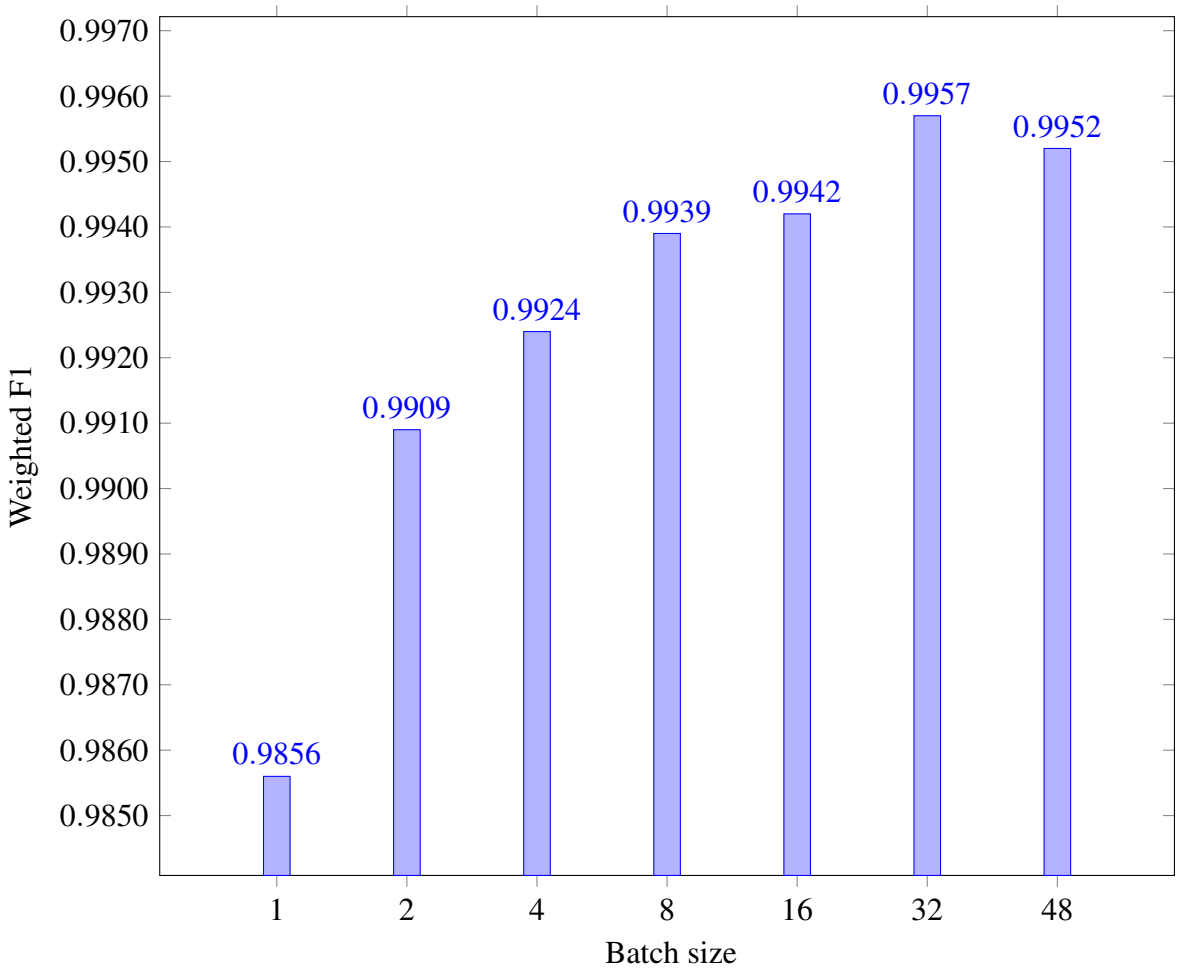


Figure 6.2: Batch sizes of the proposed method on the proposed dataset with other hyperparameters standardized

Delimiter types

In the proposed approach, the generation of inputs to BERT involves the conversion of cell values into tokens separated by delimiters. The choice of delimiters is an important hyperparameter that affects how the input data is structured and processed by the model. To determine the optimal delimiters for the proposed method, experiments were conducted with various types of delimiters and configurations. The results of these experiments are presented in Figure 6.3, which illustrates the performance of the model under different delimiter settings.

One explored aspect was the use of separate delimiters for column headers and cell values versus using the same delimiters for both. By using separate delimiters, the aim was to provide distinct cues to the model for distinguishing between column headers and cell values. This can be particularly useful when the structure of the data is complex or when there is a need to capture specific patterns or relationships between columns and their respective headers.

As illustrated in Figure 6.3, the performance of the model shows minimal variation when different delimiters are employed. Changing between various delimiters does not have a significant impact on the overall performance of the model. However, an interesting finding emerges when introducing two types of delimiters to differentiate between different parts of the text. This modification leads to a notable increase in performance, indicating that the utilization of distinct delimiters for column headers and cell values enhances the model's ability to capture contextual information and improve classification accuracy. This observation suggests that the choice of delimiters can indeed play an important role in optimizing the performance of the model.

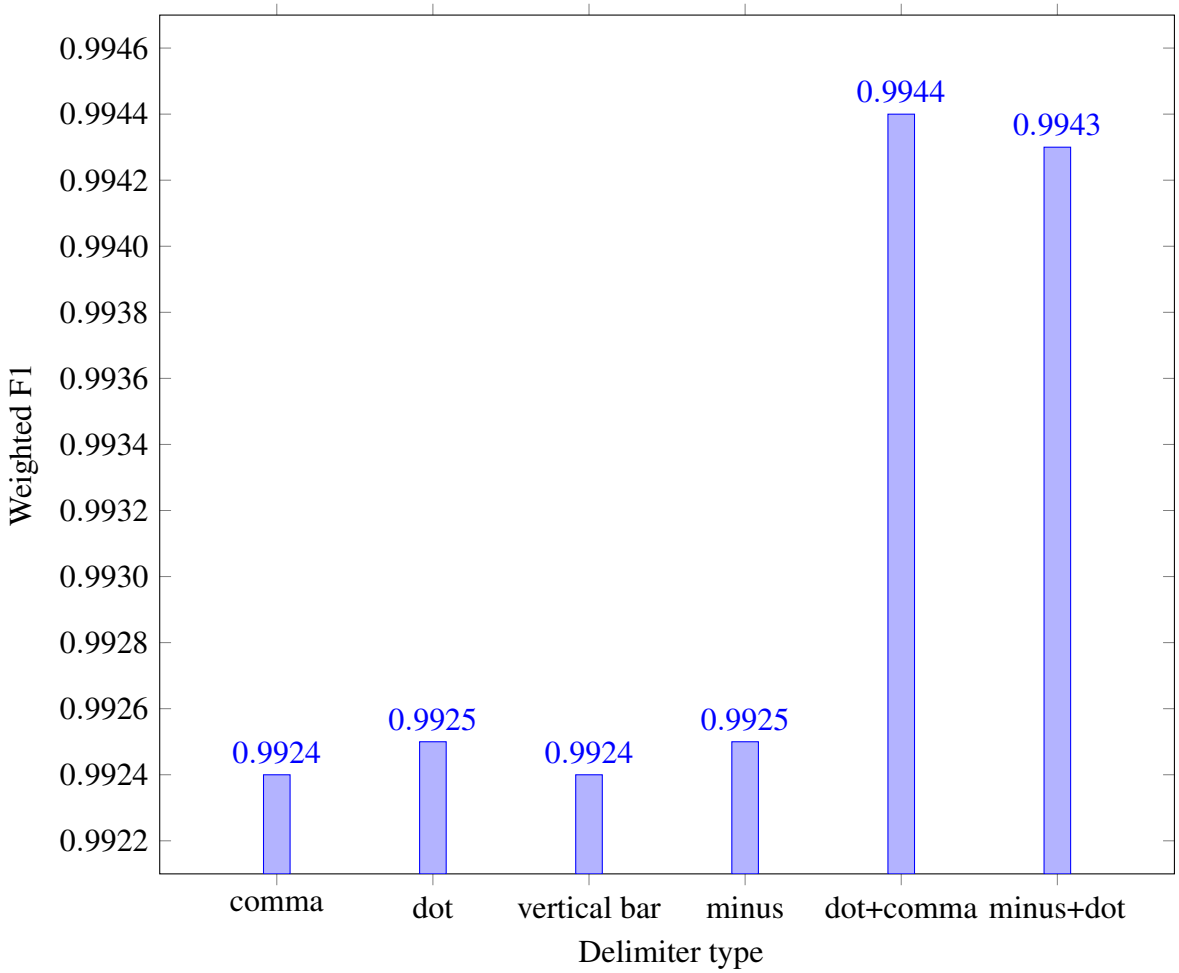


Figure 6.3: Delimiter types

Extended input averaging

Since the input size of the model has the potential to exceed the maximum token limit for BERT's input, the proposed method addresses this limitation by creating multiple inputs and

averaging their results. By increasing the number of inputs, the aim is to capture as much information as possible while still ensuring that the overall input size remains manageable. The computational complexity of the model increases proportionally with the number of inputs, as each input requires additional processing and forward passes through the model. However, this trade-off is necessary to incorporate a sufficient amount of context and improve the model's performance.

The experimental results, depicted in Figure 6.4, demonstrate the impact of varying the number of inputs on the model's performance. The results indicate that the performance of the model improves as the number of inputs increases, up until a certain threshold. In our dataset, this threshold is observed to be at 4 inputs. Beyond this point, the performance shows diminishing returns, suggesting that either most columns contain enough information to fill up to 4 inputs or that the relevant information is predominantly captured in the initial inputs.

This finding highlights the importance of incorporating multiple inputs to capture contextual information effectively. By averaging the results of multiple inputs, the model benefits from the collective knowledge contained within the different parts of the column. However, there is a point of diminishing returns, beyond which the additional inputs do not contribute significantly to the model's performance.

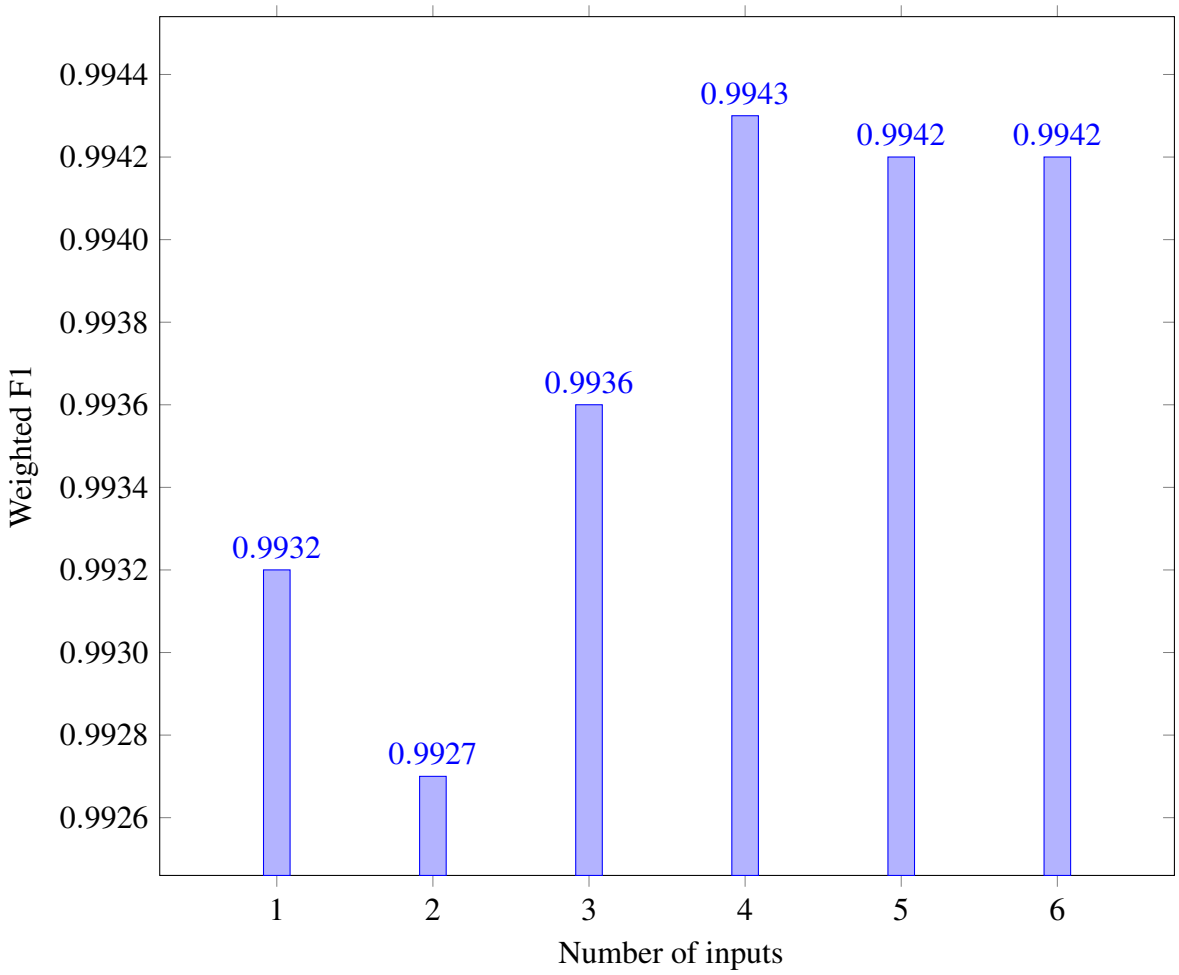


Figure 6.4: Number of inputs of the proposed method on the proposed dataset with other hyperparameters standardized

Multi-label threshold

The multi-label threshold [98] is a hyperparameter that plays a significant role in the classification process of multi-label models. In scenarios where a model can be assigned multiple labels simultaneously, the multi-label threshold serves as a determining factor for classifying a given output value as belonging to a specific class. When making predictions in a multi-label setting, the model produces output values for each potential class. These output values are typically probabilities or confidence scores associated with each class. The multi-label threshold acts as a decision boundary, setting a minimum value that the output must exceed in order to be considered as belonging to a particular class.

By setting an appropriate threshold, a stable measure for classification is established, ensuring that only output values above the threshold are assigned to their respective classes. This threshold is often determined based on the specific requirements of the task and the desired

trade-off between precision and recall. A higher threshold value leads to a stricter classification criterion, requiring the model to be more confident in its prediction before assigning a label. This approach can help reduce false positives, as only highly confident predictions are accepted. However, it may also increase the likelihood of false negatives, where true positive instances are incorrectly classified as negative. Conversely, a lower threshold value results in a more loose classification criterion, allowing for a wider range of predictions to be assigned as positive. This approach increases the recall of positive instances but may also lead to more false positives.

Selecting the optimal multi-label threshold involves considering the specific requirements of the task and the relative importance of precision and recall. It often requires a careful balance, aiming to achieve a satisfactory trade-off between correctly identifying positive instances and minimizing false positives. While a threshold value can be set for each class separately, it was decided against it, since it would be highly specific to the dataset at hand and would require an order of magnitude higher amount of testing and adjusting hyperparameters.

In Figure 6.5 the experiments with different multi-label thresholds can be seen. As the multi-label threshold goes to the extremes, the resulting performance decreases, but while maintaining it between 0.1 and 0.3, it stays around the same value with the best performance at 0.2, thereby achieving the best trade-off between recall and precision.

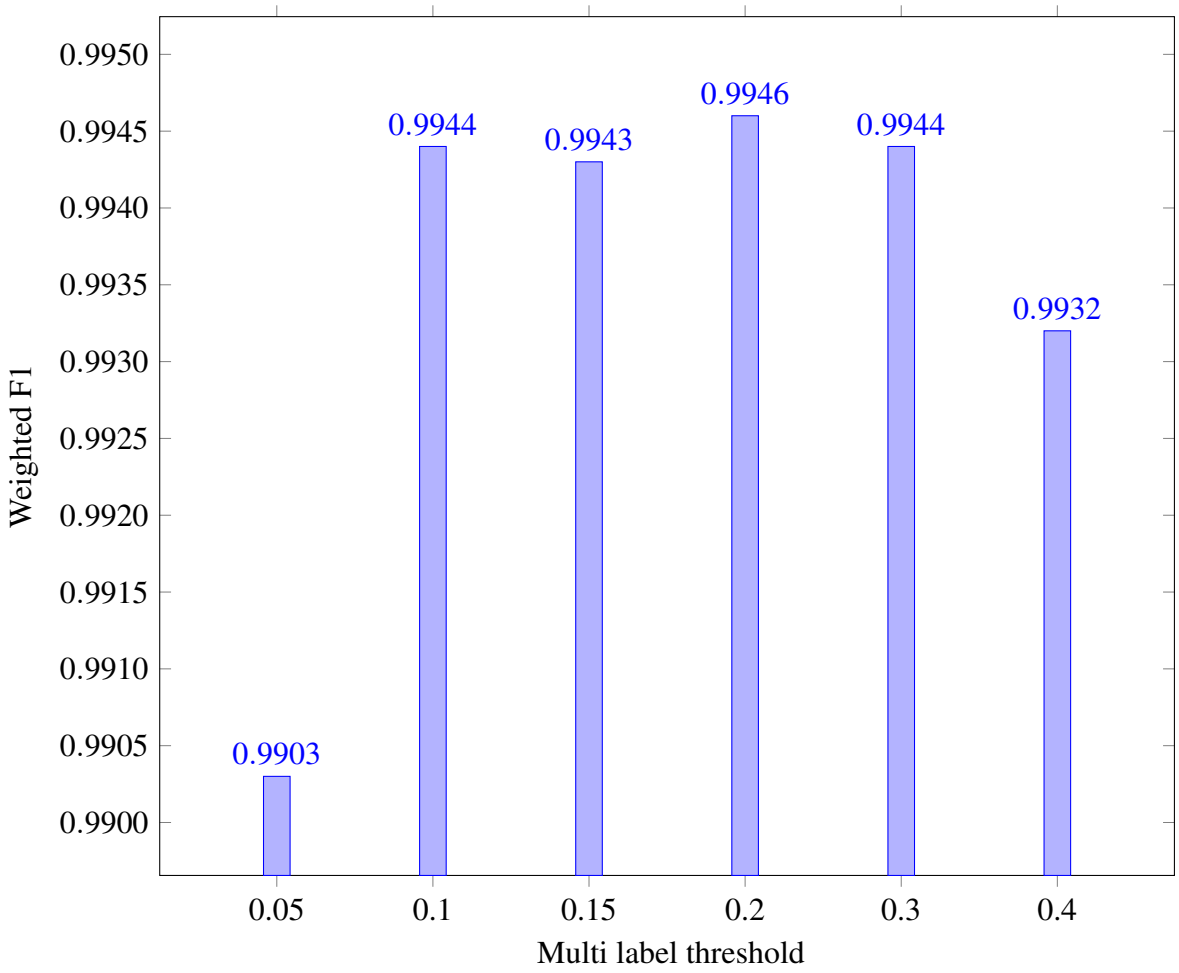


Figure 6.5: Multi label thresholds of the proposed method on the proposed dataset with other hyperparameters standardized

6.3.2 Comparison with the alternative approaches

In this subsection, the alternative approaches are compared to the proposed method and the baseline model.

In Table 6.1 the approaches are trained and tested on the proposed dataset. The approach without column concatenation shows significantly worse results than the other three approaches. The proposed method and the approaches of static and dynamic representations of adjacent columns achieve similar results, indicating that not much useful information was gained from the additional context. While the additional context does not equate to better features in the case of the proposed dataset, it still might result in better results on datasets that have often coinciding columns present in both training and test data. In the proposed dataset, all columns are randomly put into a large table, thereby making the alternative approaches useless. On the other hand, the Viznet dataset, although it is not used for sensitive data detection, but rather for

semantic column labeling in general, has only a few columns in each table, and they might be closer related to each other or often appear together in the data. The results on the Viznet dataset are shown in Table 6.1 and while the dynamic approach slightly outperforms our proposed approach (in the third decimal), it does not do it by a significant margin.

Since the alternative approaches did not yield better results for the datasets used, it was not reasonable to propose them as the main method of the thesis, because of their 3 to 5 times increased complexity and computational resources. While they did not prove better in these use cases, one should bear in mind that they might bring more fruitful results in databases with closely co-occurring columns.

Comparison of alternative approaches with the proposed method					
Datasets	Results				
Proposed dataset	Metric	Proposed method	No column concatenation	Static adjacent columns	Active adjacent columns
	Weighted F1	0.9952	0.9930	0.9952	0.9953
Viznet	Metric	Proposed method	No column concatenation	Static adjacent columns	Active adjacent columns
	Weighted F1	0.896	0.876	0.898	0.900

Table 6.1: The comparison of results of the proposed model and the models of related work on their datasets on metrics they provided in their research papers.

6.3.3 Comparisons on datasets from related work

In order to compare the proposed method with related work and get a basic idea of how well the machine learning part performs, the performance on datasets from similar tasks of models discussed in the related work section is compared. Only the machine learning part of the proposed model is used, as the rule-based methods are specifically created with sensitive data types in mind, which mostly do not occur in these datasets. Hence, in further experiments, only the part of the model up to the post-processing step is evaluated.

While column headers often carry important information that can help in the detection of column labels in real-world applications, datasets from related work do not take them into account. This is either because they use the available column headers to generate labels for their dataset or because they consider that headers could not be created in such a way that they adequately represent real-world data. To compare the proposed method with alternative approaches on these datasets, the part of the proposed model which takes column headers into account is removed and only the input containing field values was left.

In Sherlock [66], the dataset was created by extracting specific columns from the VizNet [81] corpus and labeling them by their column headers, thereby creating 78 possible labels. The dataset was split into a training/validation/test set using the 60/20/20 percentages division. As seen in Table 6.2, their method achieved a weighted F1 score of 0.890 on the test set, while the base part of the proposed model scored slightly better, achieving a weighted F1 score of 0.896, without using column headers or additional rule-based methods.

Comparison of the proposed model with related work models on their datasets			
Datasets	Results		
VizNet	Metric	Sherlock	Proposed model
	Weighted F1	0.890	0.896
Faker data	Metric	SIMON	Proposed model
	Weighted F1	0.84	0.996
WikiTables	Metric	SeLab	Proposed model
	Micro F1	0.51	0.72
	Macro F1	0.66	0.80

Table 6.2: The comparison of results of the proposed model and the models of related work on their datasets on metrics they provided in their research papers.

SIMON [65] semantically classifies columns in tabular data. The main data that they used for training, as well as their initial tests, were generated from Faker [80] and contained nine different label types. This thesis followed their data generation procedure and generated 10,000 columns of data, which were split according to their proposal with a 60/30/10 training/validation/test percentage division. Taking into account that the created data was automatically generated, the proposed model still achieved significantly better results than SIMON. As seen in Table 6.2, their model produced a weighted F1 score of 0.84, while the proposed model classified the dataset almost perfectly, with a weighted F1 score of 0.996. The performance of the proposed model clearly shows that only using character-level features and omitting broader context negatively impacts the model's performance.

SeLaB [14] uses a curated version of the WikiTables dataset from the WikiTables corpus [84], which contains data from over 1.6 million Wikipedia pages. As seen in Table 6.2, their model achieved a macro F1 score of 0.51 and a micro F1 score of 0.72, the proposed model managed to achieve a macro F1 score of 0.66, and a micro F1 score of 0.80, thereby showing the advantages of capturing the context of the whole column.

The SATO [67] method took a subset of the Viznet [81] dataset used in [66] dataset and split it into five parts, instead of training and testing the model on the exact dataset of Sherlock.

After the cross-validation was performed on those five parts, the authors very unconventionally reported their results as the average of those five test runs of the cross-validation. They did not create a separate test set which the model has not seen, thus it was unknown how to properly compare the proposed model with theirs since the use of methods that clearly overfit is not encouraged. The dataset used in SATO was split with a 60/20/20 split between train, validation, and test corpora, thereby taking the datasets of their cross-validation indexed as 0, 1, and 2 as training; 3 as validation; and 4 as test sets. We achieved a weighted F1 score of 0.90, which is comparable to the result of the proposed model on the full Viznet dataset and is expected since SATO worked on a segment of it.

Other methods mentioned in related work, such as TaBERT [68] and TABBIE [69], deal with the same problem of embedding a database table but are used for very different downstream tasks. They focus on the tasks of question-answering or outlier detection and are not meant to classify columns of database tables, so comparing the proposed model with theirs would probably lead to misleading information.

Lastly, SemTab [99] is a yearly challenge for the ontology task of knowledge base construction, which also incorporates a task where column labels need to be predicted. However, as the task was created as an ontology challenge, the authors did not provide training and test datasets for comparison, but rather tested the submitted models which were trained on arbitrary datasets, or used approaches that are not based on machine learning and did not require training data. As the experiments were conducted without a predefined training and test set, there was no feasible way to properly compare the proposed model with the results from SemTab.

6.3.4 Training and testing on the proposed dataset

The results on the proposed DeSSI dataset (test part) for the proposed method that uses only the machine learning part are presented in Table 6.3. They are presented for each label, giving its precision (P), recall (R), and F1 score. The results show a very high and sometimes perfect F1 score for some classes. The test dataset results of the baseline model, related work, and both the proposed model with and without the rule-based methods are reported in Table 6.4. The results show that the proposed model significantly outperforms the baseline model, as well as Sherlock and SeLaB, in all aspects, while the difference between the proposed models with and without rule-based methods can mostly be seen in the macro-averaged metric types. While the machine learning part on its own has very good results in general, the rule-based part of the model provides classification when a label has been found under strict rules, and in doing so only bolsters the recall of labels that can be detected in such a way. Rule-based methods are not used on the most represented labels, such as Other data, Phone number, Address, and Person, and thus the improvement will mostly be visible in the macro-averaged recall metric, and less so in other metrics.

Results of the proposed model on the proposed DeSSI dataset, by label				
Class	P	R	F1	S
Other data	0.9955	0.9940	0.9947	1334
Phone number	0.9966	0.9954	0.9960	876
Address	0.9973	0.9987	0.9980	742
Person	1.0000	0.9852	0.9925	741
Email	1.0000	0.9956	0.9978	678
NIN	0.9793	0.9822	0.9807	673
Date	0.9982	0.9982	0.9982	570
Organization	1.0000	1.0000	1.0000	434
GPE	1.0000	1.0000	1.0000	424
Geolocation	1.0000	1.0000	1.0000	400
SWIFT/BIC	0.8667	0.9811	0.9204	53
IBAN	1.0000	1.0000	1.0000	28
Passport	0.9130	0.8400	0.8750	25
Religion	1.0000	1.0000	1.0000	22
CCN	1.0000	1.0000	1.0000	18
ID Card	0.8000	1.0000	0.8889	16
Sexuality	1.0000	1.0000	1.0000	14
Gender	0.9286	1.0000	0.9630	13
Nationality	0.9231	1.0000	0.9600	12
Race	1.0000	1.0000	1.0000	9

Table 6.3: The results showing precision (P), recall (R), F1 score (F1), and support (S) of only the machine learning part of the proposed model on the proposed DeSSI dataset, test part. Where support depicts the number of columns with that sensitive data inside of them.

6.3.5 Training and testing on real-world data

The principal goal of any sensitive data detection approach is to work efficiently with real-world data [100]. Therefore, for internal testing and analyzing how well DeSSI represents real-world data, another dataset was created from real-world data that cannot be openly published due to the strict NDA with a collaborating company. That dataset is around half the size of the proposed dataset and the results on it are slightly worse than the results on the proposed DeSSI dataset, by around 0.02 for weighted F1, achieving a weighted F1 score of 0.976 (compare to

Comparisons of models on DeSSI					
Type	Baseline	Sherlock	SeLaB	*Proposed method*	Proposed method
Macro F1	0.8672	0.895	0.7790	0.9802	0.9837
Weighted F1	0.9170	0.931	0.7429	0.9957	0.9959

Table 6.4: The results of the baseline model, Sherlock, SeLab, the proposed method with only the machine-learning part (*Proposed method*), and the whole proposed method on macro-averaged and weighted metrics of and F1 score (F1) on the proposed DeSSI dataset, test part.

Table 6.4). The slight drop in performance is expected and can be attributed to the variations and noisy data that are found in the real-world dataset. In creating DeSSI, the aim was to introduce as much nuance and noise as possible, however, due to the diverse nature of these variations in real-world datasets, it is impossible to represent all of them without making the dataset lose its generalizability. Thus, it is expected that results will always be slightly better on the proposed DeSSI dataset than on real-world datasets.

Chapter 7

Conclusions

7.1 Main findings

This thesis aimed to develop novel models and explore the field of sensitive data detection in structured datasets, as well as to the field of semantic column labeling. By implementing an attention-based model pre-trained on natural language on the combination of natural language and structured language present in structured datasets.

The main objectives of this research were to address the challenge of structured sensitive data detection and make significant contributions to the field. The thesis successfully achieved these objectives by:

1. Proposing a novel method – The thesis introduced a novel approach that combines active context-based classification with traditional rule-based heuristics. This method aims to accurately classify columns into one or more labels, improving the detection of sensitive data in structured datasets.
2. Creating a benchmark dataset – To provide a standard benchmark for evaluating structured sensitive data detection methods, a new dataset was developed. This dataset fills a crucial gap in the field, enabling researchers to assess the performance of their approaches and establish a baseline for future studies.
3. Exploring alternative approaches – The thesis explored alternative approaches by making slight modifications to the model architecture and the concept of context. These explorations allowed for the investigation of different hypotheses and provided insights into the benefits and limitations of various design choices.
4. Conducting comprehensive experiments – Rigorous experimentation was conducted to fine-tune the proposed method and evaluate its performance. The experiments involved identifying optimal hyperparameter values, comparing alternative approaches, and assessing the superiority of the proposed method over the baseline model and the related work.

5. Demonstrating superior performance – The proposed method's classification model demonstrated superior performance compared to the baseline model, as well as the models from related work, on both the novel dataset and published datasets for semantic column labeling. This highlights the effectiveness of the active context-based approach and its potential for broader applications.

By achieving these objectives, this thesis significantly contributes to the field of structured sensitive data detection. The proposed method and benchmark dataset provide valuable tools for researchers and practitioners to enhance the accuracy and efficiency of sensitive data detection in structured datasets. The thesis opens avenues for further research and development in the domain, ultimately improving data privacy and security in various applications.

7.2 Limitations

While the proposed model demonstrates a significant increase in performance compared to other models from related work, it is important to acknowledge its limitations. Despite efforts to address these limitations through various workarounds, they still impose constraints on the model's capabilities.

Firstly, the model operates with only a subset of the possible input due to computational complexity. Considering all values within a table becomes infeasible in terms of computation, leading to the model only considering a limited portion of the data. This limitation can potentially result in the model missing important patterns or information present in the excluded data.

Secondly, the model has a fixed input size, which poses challenges when dealing with larger inputs. To overcome this, the model splits the input into smaller parts, processes them separately, and averages the results. While this approach helps manage computational constraints, it may not capture the full context of the input and could potentially lead to a loss of important information.

Thirdly, the proposed dataset used in the experiments is synthetic. Despite efforts to make synthetic datasets resemble real-world scenarios, they inherently lack the complexity and variability of real datasets. Real-world datasets contain nuances, inconsistencies, and contextual dependencies that are difficult to fully replicate in synthetic datasets. This limitation raises concerns about the generalizability of the proposed model to real-world scenarios although the results on real-world datasets presented in the thesis suggest that the method generalizes well to a degree.

Addressing these limitations remains a challenge, as they are inherent to the nature of the problem and the constraints of the computational resources available.

7.3 Future work

Throughout this thesis, significant effort has been devoted to conducting extensive work, training, curation, and experimentation. However, it is important to acknowledge that computational complexity and time constraints have imposed limitations on certain aspects of the research. As a result, there is room for further work and expansion, particularly in terms of dataset development and conducting more comprehensive tests with increased computational resources.

One potential avenue for future work is the expansion of the dataset itself. This could involve increasing the size and diversity of the dataset by incorporating additional real-world data sources. By expanding the dataset with more data, edge cases, and combinations of labels in a single column, researchers would have access to a wider range of scenarios and data patterns, allowing for more robust evaluation and validation of the proposed approaches.

Furthermore, future research could explore alternative model architectures that capture input and context in different ways. The current approach has demonstrated promising results, but there may be other architectures or techniques that can further improve the model's ability to extract relevant information from the input. Exploring different model architectures could potentially lead to enhanced performance and a better understanding of the underlying patterns in structured data.

To fully explore these possibilities, it would be beneficial to have access to greater computational resources. With more computational power, researchers would be able to conduct more extensive experiments, fine-tune hyperparameters, and explore larger-scale models [101] [102]. This would allow for a deeper investigation into the effectiveness and scalability of the proposed methods.

In conclusion, while this thesis has made significant contributions to the field of structured sensitive data detection, there are still avenues for further exploration and improvement. Expanding the dataset, exploring different model architectures, and leveraging increased computational resources are potential directions for future work to enhance the capabilities and performance of the proposed approaches.

Bibliography

[1]U.S. National Institute of Standards and Technology (NIST), “Guide to protecting the confidentiality of personally identifiable information (pii)”, Tech. Rep. Special Publication 800-122, 2018.

[2]European Parliament and of the Council, “Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)”, https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016.

[3]California Office of the Attorney General, “California consumer privacy act (ccpa)”, https://oag.ca.gov/privacy/ccpa, 2021.

[4]Presidency of the Republic of Brazil, “General personal data protection law (lei geral de proteção de dados pessoais - lgpd)”, http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm, 2018.

[5]Spector, A. Z., Norvig, P., Wiggins, C., Wing, J. M., Data Science in Context: Foundations, Challenges, Opportunities. Cambridge University Press, 2022.

[6]Huang, T. S., Wu, S. D., “Multimedia processing: Past, present and future”, Signal Processing: Image Communication, Vol. 29, No. 8, 2014, str. 877–890.

[7]Grishman, R., “Message understanding conference-6: A brief history”, in COLING-96, 1996, str. 466–471.

[8]Manning, C. D., Schütze, H., Foundations of Statistical Natural Language Processing. MIT press, 1999.

[9]Jurafsky, D., Martin, J. H., “Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition”, 2000.

[10]Likforman-Sulem, L., Zahour, A., “Ocr systems: A review”, Journal of Imaging, Vol. 4, No. 3, 2018, str. 38.

[11]Jurafsky, D., Martin, J. H., Speech and Language Processing, 3rd ed. Pearson, 2019.

[12]McAfee, A., Brynjolfsson, E., "Big data: The management revolution", Harvard Business Review, Vol. 90, No. 10, 2011, str. 60–68.

[13]Quinn, P., Malgieri, G., "The Difficulty of Defining Sensitive Data—The Concept of Sensitive Data in the EU Data Protection Framework", German Law Journal, Vol. 22, No. 8, 2021, str. 1583–1612.

[14]Trabelsi, M., Cao, J., Heflin, J., "SeLaB: Semantic labeling with bert", in 2021 International Joint Conference on Neural Networks (IJCNN), 2021, str. 1-8.

[15]Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., Gómez-Berbís, J. M., "Named entity recognition: Fallacies, challenges and opportunities", Computer Standards & Interfaces, Vol. 35, No. 5, 2013, str. 482-489, dostupno na: https://www.sciencedirect.com/science/article/pii/S0920548912001080

[16]Kužina, V., Vušak, E., Jovi ć, A., "Methods for automatic sensitive data detection in large datasets: a review", in 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO). IEEE, 2021, str. 187–192.

[17]Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J., "Distributed representations of words and phrases and their compositionality", Advances in neural information processing systems, Vol. 26, 2013, str. 3111–3119.

[18]Sait, S. M., Jelani, D. I., Ali, M. A. N., "Dimensionality reduction techniques: A review", Journal of Artificial Intelligence, 2014, dostupno na: https://www.researchgate.net/publication/283263187_Dimensionality_Reduction_Techniques_A_Review

[19]Hamilton, W. L., Ying, R., Leskovec, J., Graph Representation Learning. Morgan & Claypool Publishers, 2020.

[20]Vušak, E., Kužina, V., Jovi ć, A., "A survey of word embedding algorithms for textual data information extraction", in 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO). IEEE, 2021, str. 181–186.

[21]Zhang, X., Zhao, J., LeCun, Y., "Character-level convolutional networks for text classification", in Advances in neural information processing systems, 2015, str. 649–657.

[22]Lin, Y., Liu, Z., Sun, M., "Chinese word embeddings with enhanced features", in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015), 2015.

[23]LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., "Gradient-based learning applied to document recognition", Proceedings of the IEEE, Vol. 86, No. 11, 1998, str. 2278–2324.

[24]Goodfellow, I., Bengio, Y., Courville, A., "Deep learning", in Deep learning. MIT Press, 2016, str. 265–313.

[25]Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., "Enriching word vectors with subword information", in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, str. 3–14.

[26]Gage, P., "A new algorithm for data compression", C Users Journal, Vol. 12, No. 2, 1994, str. 23–38.

[27]Joulin, A., Grave, E., Bojanowski, P., Mikolov, T., "Bag of tricks for efficient text classification", arXiv preprint arXiv:1607.01759, 2016.

[28]Firth, J., A Synopsis of Linguistic Theory 1930-1955. Studies in Linguistic Analysis, 1957.

[29]Mikolov, T., Chen, K., Corrado, G., Dean, J., "Efficient estimation of word representations in vector space", arXiv preprint arXiv:1301.3781, 2013.

[30]Pennington, J., Socher, R., Manning, C. D., "GloVe: Global vectors for word representation", in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, str. 1532–1543.

[31]Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., "Deep contextualized word representations", arXiv preprint arXiv:1802.05365, 2018.

[32]Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., "BERT: Pre-training of deep bidirectional transformers for language understanding", arXiv preprint arXiv:1810.04805, 2018.

[33]Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. *et al.*, "Improving language understanding by generative pre-training", Preprint., 2018, dostupno na: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[34]Alpaydin, E., Introduction to Machine Learning, 2nd ed. MIT Press, 2010.

[35]Hastie, T., Tibshirani, R., Friedman, J., The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, 2009.

[36]Jordan, M. I., Mitchell, T. M., “Machine learning: Trends, perspectives, and prospects”, Science, Vol. 349, No. 6245, 2015, str. 255–260.

[37]Goyvaerts, J., Levithan, S., Regular Expressions Cookbook. O’Reilly Media, 2009.

[38]Zheng, A., Casari, A., Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O’Reilly Media, 2018.

[39]Jackson, P., Introduction to Expert Systems. Addison-Wesley, 1998.

[40]Staab, S., Studer, R., Handbook on Ontologies. Springer Science & Business Media, 2011.

[41]Russell, S., Norvig, P., Davis, E., Artificial Intelligence: A Modern Approach. Pearson, 2010.

[42]“CLIPS - c language integrated production system”, http://www.clipsrules.net/.

[43]“Drools - business rules management system”, https://www.drools.org/.

[44]Baum, L. E., Petrie, T., “Statistical inference for probabilistic functions of finite state markov chains”, The annals of mathematical statistics, Vol. 37, No. 6, 1966, str. 1554–1563.

[45]Chen, T., Cullen, R. M., Godwin, M., “Hidden markov model using dirichlet process for de-identification”, Journal of biomedical informatics, Vol. 58, 2015, str. S60–S66.

[46]Stubbs, A., Kotfila, C., Uzuner, Ö., “Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1”, Journal of biomedical informatics, Vol. 58, 2015, str. S11–S19.

[47]Blei, D. M., Ng, A. Y., Jordan, M. I., “Latent dirichlet allocation”, Journal of machine Learning research, Vol. 3, No. Jan, 2003, str. 993–1022.

[48]Lafferty, J. D., McCallum, A., Pereira, F. C. N., “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”, in Proceedings of the Eighteenth International Conference on Machine Learning, ser. ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, str. 282–289.

[49]Berg, H., Dalianis, H., “Augmenting a de-identification system for swedish clinical text using open resources and deep learning”, in 22nd Nordic Conference on Computational Linguistics (NoDaLiDa), Turku, Finland, September 30, 2019. Linköping University Electronic Press, 2019, str. 8–15.

[50]Liu, Z., Chen, Y., Tang, B., Wang, X., Chen, Q., Li, H., Wang, J., Deng, Q., Zhu, S., "Automatic de-identification of electronic medical records using token-level and character-level conditional random fields", Journal of biomedical informatics, Vol. 58, 2015, str. S47–S52.

[51]Rumelhart, D. E., Hinton, G. E., Williams, R. J. *et al.*, "Learning internal representations by error propagation", 1985.

[52]Jordan, M. I., "Serial order: A parallel distributed processing approach", in Advances in psychology. Elsevier, 1997, Vol. 121, str. 471–495.

[53]Srivastava, A., Ekbal, A., Saha, S., Bhattacharyya, P. *et al.*, "A recurrent neural network architecture for de-identifying clinical records", in Proceedings of the 13th international conference on natural language processing, 2016, str. 188–197.

[54]Hochreiter, S., Schmidhuber, J., "Long short-term memory", Neural computation, Vol. 9, No. 8, 1997, str. 1735–1780.

[55]Richter-Pechanski, P., Amr, A., Katus, H. A., Dieterich, C., "Deep learning approaches outperform conventional strategies in de-identification of german medical reports.", in GMDS, 2019, str. 101–109.

[56]Madan, A., George, A. M., Singh, A., Bhatia, M., "Redaction of protected health information in ehrs using crfs and bi-directional lstms", in 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO). IEEE, 2018, str. 513–517.

[57]Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., "Attention is all you need", Advances in neural information processing systems, Vol. 30, 2017.

[58]García-Pablos, A., Perez, N., Cuadros, M., "Sensitive data detection and classification in spanish clinical text: Experiments with bert", arXiv preprint arXiv:2003.03106, 2020.

[59]Johnson, A. E., Bulgarelli, L., Pollard, T. J., "Deidentification of free-text medical records using pre-trained bidirectional transformers", in Proceedings of the ACM Conference on Health, Inference, and Learning, 2020, str. 214–221.

[60]Wu, C., Wu, F., Qi, T., Huang, Y., "Named entity recognition with context-aware dictionary knowledge", in China National Conference on Chinese Computational Linguistics. Springer, 2020, str. 129–143.

[61]Trifacta, “Data wrangling tools & software”, https://www.trifacta.com, accessed: 2022-01-10. 2014.

[62]Microsoft, “Microsoft Power BI”, https://powerbi.microsoft.com/en-us/, accessed: 2022-01-10. 2016.

[63]Google, “Cloud data loss prevention”, https://cloud.google.com/dlp, accessed: 2022-01-10. 2018.

[64]PII, “PII Catcher for files and Databases”, https://pypi.org/project/piicatcher/, accessed: 2022-01-10. 2018.

[65]Azunre, P., Corcoran, C., Dhamani, N., Gleason, J., Honke, G., Sullivan, D., Ruppel, R., Verma, S., Morgan, J., “Semantic classification of tabular datasets via character-level convolutional neural networks”, arXiv preprint arXiv:1901.08456, 2019.

[66]Hulsebos, M., Hu, K., Bakker, M., Zgraggen, E., Satyanarayan, A., Kraska, T., Demiralp, Ç., Hidalgo, C., “Sherlock: A deep learning approach to semantic data type detection”, in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, str. 1500–1508.

[67]Zhang, D., Suhara, Y., Li, J., Hulsebos, M., Demiralp, Ç., Tan, W.-C., “Sato: Contextual semantic type detection in tables”, arXiv preprint arXiv:1911.06311, 2019.

[68]Yin, P., Neubig, G., Yih, W.-t., Riedel, S., “TaBERT: Pretraining for joint understanding of textual and tabular data”, arXiv preprint arXiv:2005.08314, 2020.

[69]Iida, H., Thai, D., Manjunatha, V., Iyyer, M., “TABBIE: Pretrained representations of tabular data”, arXiv preprint arXiv:2105.02584, 2021.

[70]Kužina, V., Petric, A.-M., Bariši ć, M., Jović, A., “Cassed: Context-based approach for structured sensitive data detection”, Expert Systems With Applications, Vol. 223, 2023, str. 119924.

[71]“IBM Security Guardium Data Protection”, https://www.ibm.com/products/ibm-guardium-data-protection, accessed: 2021-05-05.

[72]“Nightfall AI”, https://nightfall.ai/, accessed: 2021-05-05.

[73]“Presidio - Data Protection and Anonymization API”, https://github.com/microsoft/presidio, accessed: 2021-05-05.

[74]“Gretel AI”, https://gretel.ai/, accessed: 2021-05-05.

[75]Le, Q., Mikolov, T., "Distributed representations of sentences and documents", in International conference on machine learning. PMLR, 2014, str. 1188–1196.

[76]Ateniese, G., Felici, G., Mancini, L. V., Spognardi, A., Villani, A., Vitali, D., "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers", CoRR, Vol. abs/1306.4447, 2013, dostupno na: http://arxiv.org/abs/1306.4447

[77]Dwork, C., Smith, A., Steinke, T., Ullman, J., "Exposed! A Survey of Attacks on Private Data", Annual Review of Statistics and Its Application (2017), 2017, accessed: 2021-04-08.

[78]Chen, R., Lu, M., Chen, T., Williamson, D. F. K., Mahmood, F., "Synthetic data in machine learning for medicine and healthcare.", Nature Biomedical Engineering, Vol. 5, 2021, str. 493–497, accessed: 2022-01-03, dostupno na: https://doi.org/10.1038/s41551-021-00751-8

[79]Nikolenko, S. I., "Synthetic data for deep learning", CoRR, Vol. abs/1909.11512, 2019, dostupno na: http://arxiv.org/abs/1909.11512

[80]Faker, "Faker", https://faker.readthedocs.io/en/master/, accessed: 2021-02-23. 2021.

[81]Hu, K., Gaikwad, S., Hulsebos, M., Bakker, M. A., Zgraggen, E., Hidalgo, C., Kraska, T., Li, G., Satyanarayan, A., Demiralp, Ç., "VizNet: Towards a large-scale visualization learning and benchmarking repository", in Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, str. 1–12.

[82]Pyle, D., Data Preparation for Data Mining, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

[83]DeSSI, "DeSSI - dataset for structured sensitive information", https://www.kaggle.com/sensitivedetection/dessi-dataset-for-structured-sensitive-information, accessed: 2022-01-14. 2022.

[84]Bhagavatula, C. S., Noraset, T., Downey, D., "Tabel: Entity linking in web tables", in The Semantic Web - ISWC 2015, Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Thirunarayan, K., Staab, S., (ur.). Cham: Springer International Publishing, 2015, str. 425–441.

[85]Kužina, V., "CASSED software", https://github.com/VKuzina/CASSED, 7 2022.

[86]Ye, X., Chen, Q., Wang, X., Dillig, I., Durrett, G., "Sketch-Driven Regular Expression Generation from Natural Language and Examples", Transactions of the Association for Computational Linguistics, Vol. 8, 11 2020, str. 679-694.

[87]Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., Vollgraf, R., "FLAIR: An easy-to-use framework for state-of-the-art NLP", in NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), 2019, str. 54–59.

[88]Sanh, V., Debut, L., Chaumond, J., Wolf, T., "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter", arXiv preprint arXiv:1910.01108, 2019.

[89]Loshchilov, I., Hutter, F., "Decoupled weight decay regularization", arXiv preprint arXiv:1711.05101, 2017.

[90]Sasaki, Y., "The truth of the f-measure", Journal of Machine Learning Research, Vol. 8, No. Nov, 2007, str. 667–685.

[91]Sokolova, M., Lapalme, G., "A systematic analysis of performance measures for classification tasks", in Proceedings of the 2009 ACM Symposium on Applied Computing. ACM, 2009, str. 102–107.

[92]Hutter, F., Kotthoff, L., Vanschoren, J., Automated Machine Learning: Methods, Systems, Challenges. Springer, 2019.

[93]Kingma, D. P., Ba, J., "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980, 2014.

[94]Prechelt, L., "Early stopping-but when?", Neural Networks, Vol. 11, No. 4, 1998, str. 761–769.

[95]Smith, L. N., "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay", arXiv preprint arXiv:1803.09820, 2018.

[96]Lillicrap, T., Hunt, J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., "Don't decay the learning rate, increase the batch size", in International Conference on Learning Representations (ICLR), 2017, dostupno na: https://arxiv.org/abs/1711.00489

[97]Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P. T. P., "On large-batch training for deep learning: Generalization gap and sharp minima", in International Conference on Learning Representations (ICLR), 2017, dostupno na: https://arxiv.org/abs/1609.04836

[98]Tsoumakas, G., Katakis, I., Vlahavas, I., “Multi-label classification: An overview”, in International Conference on Data Mining and Knowledge Discovery. Springer, 2007, str. 935–939.

[99]Cutrona, V., Chen, J., Efthymiou, V., Hassanzadeh, O., Jiménez-Ruiz, E., Sequeda, J., Srinivas, K., Abdelmageed, N., Hulsebos, M., Oliveira, D. *et al.*, “Results of SemTab 2021”, Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, Vol. 3103, 2022, str. 1–12.

[100]Menzies, T., Greenwald, J., Frank, A., “Data mining for very busy people”, in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004, str. 770–775.

[101]Beltagy, I., Peters, M. E., Cohan, A., “Longformer: The long-document transformer”, arXiv preprint arXiv:2004.05150, 2020.

[102]Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., Le, Q. V., “Xlnet: Generalized autoregressive pretraining for language understanding”, Advances in neural information processing systems, Vol. 32, 2019.

Biography

Vjeko Kužina was born in Zagreb 1997. He received his Master of Engineering in Computer Science from the University of Zagreb Faculty of Electrical Engineering and Computing(FER), Zagreb, Croatia, in 2020. From September 2019 until July 2020. he has worked as a Backend Engineer in Infinum. In September 2020. he began his Ph.D. in Computer science and worked as an assistant and researcher on FER. He successfully finished the project he was a researcher on in August 2023. and started working in the Biotech startup, Ani Biome, as a Machine Learning developer. He is the author of 2 conference papers and 1 international journal paper.

List of published articles

Publications in Journals

1. Kužina, V., Petric, A.-M., Bariši ć, M., Jović, A., "CASSED: Context-based Approach for Structured Sensitive Data Detection", Expert Systems With Applications, Vol. 223, 2023, pp. 119924.

Conference Papers

1. Kužina, V., Vušak, E., Jovi ć, A., "Methods for Automatic Sensitive Data Detection in Large Datasets: a Review", 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), 2021, pp. 187-192.
2. Vušak, E., Kužina, V., Jovi ć, A., "A Survey of Word Embedding Algorithms for Textual Data Information Extraction", 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), 2021, pp. 181-186.

Životopis

Vjeko Kužina rođen je u Zagrebu 1997. godine. Diplomirao je računarstvo na Fakultetu elektrotehnike i računarstva (FER), Sveučilište u Zagrebu, Hrvatska, 2020. godine. Od rujna 2019. do srpnja 2020. radio je kao Backend inženjer u Infinumu. U rujnu 2020. započeo je doktorski studij iz računalnih znanosti te je radio kao asistent i istraživač na FER-u. Uspješno je završio projekt na kojem je bio istraživač u kolovozu 2023. i počeo raditi u biotehnološkom startupu, Ani Biome, kao inženjer strojnog učenja. Autor je 2 konferencijska rada i 1 međunarodnog časopisnog rada.