BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Resource development and machine learning approaches for natural language processing tasks

PhD Thesis summary

PhD student: Anamaria Briciu
Scientific supervisor: Prof. dr. Czibula Gabriela

2022

Keywords: natural language processing, emotion detection, authorship attribution, machine learning, deep learning, autoencoders

Contents

List of publications

The ranking of publications was performed according to the CNATDCU (National Council for the Recognition of University Degrees, Diplomas and Certificates) standards applicable for doctoral students enrolled after October 1, 2018. All rankings are listed according to the classification of journals[1] and conferences[2] in Computer Science.

Publications in Web of Science - Science Citation Index Expanded

[CLB22] Gabriela Czibula, Mihaiela Lupea and **Anamaria Briciu**. *Enhancing the performance of software authorship attribution using an ensemble of deep autoencoders*. Mathematics, Special issue on Recent Advances in Artificial Intelligence and Machine Learning, 2022, under review (**2021 IF=2.592**, Journal IF Quartile Q1)

Rank A, 0 points.

[LBB21] Mihaiela Lupea, **Anamaria Briciu** and Elena Bostenaru. *Emotion-based Hierarchical Clustering of Romanian Poetry*. Studies in Informatics and Control, v. 30, n. 1, pp. 109–118, 2021 (**2021 IF=1.862**, Journal IF Quartile Q3)

Rank C, 2 points.

[LB19] Mihaiela Lupea and **Anamaria Briciu**. *Studying emotions in Romanian words using formal concept analysis*. Computer Speech & Language, v. 57, pp. 128–145, 2019 (**2019 AIS=0.117**, Journal AIS Quartile Q2)

Rank B, 4 points.

Publications in Web of Science, Conference Proceedings Citation Index

[LBCC22] Mihaiela Lupea, **Anamaria Briciu**, Istvan Gergely Czibula and Gabriela Czibula. *SoftId: An autoencoder-based one-class classification model for software authorship identification*. 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022), accepted.

Rank B - CORE2021, 2 points.

[1] https://uefiscdi.ro/premierea-rezultatelor-cercetarii-articole
[2] http://portal.core.edu.au/conf-ranks/

[BCL21] **Anamaria Briciu**, Gabriela Czibula and Mihaiela Lupea. *AutoAt: A deep autoencoder-based classification model for supervised authorship attribution*. 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2021), September 8-10, 2021, Procedia Computer Science 192, pp. 397-406.

Rank B - CORE2021, 4 points.

[LB17] Mihaiela Lupea and **Anamaria Briciu**. *Formal Concept Analysis of a Romanian emotion lexicon*. 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 111–118, 2017.

Rank C - CORE2017, 2 points.

[BL18] **Anamaria Briciu** and Mihaiela Lupea. *Studying the language of mental illness in Romanian social media*. 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 21–28, 2018 .

Rank C - CORE2018, 2 points.

Publications in journals and conference proceedings

[BL17] **Anamaria Briciu** and Mihaiela Lupea. *RoEmoLex - A Romanian Emotion Lexicon*. Studia Universitatis Babeș-Bolyai Informatica, v. 62, n. 2, pp. 45-56, 2017 (**indexed Mathematical Reviews**).

Rank D, 1 point.

[Bri19] **Anamaria Briciu**. *Quantitative Analysis of Style in Mihai Eminescu's Poetry*. Studia Universitatis Babeș-Bolyai Informatica, v. 64, n. 2, pp. 80–95, 2019 (**indexed Mathematical Reviews**).

Rank D, 1 point.

Publications score: 18 points.

Introduction

The main research domain of our doctoral thesis concerns the development of *natural language processing* resources and applications. Our PhD thesis is entitled "Resource development and machine learning approaches for natural language processing tasks" and aims at developing resources and machine learning and deep learning models for a series of language processing tasks.

Natural Language Processing (NLP) refers to the research area concerned with the exploration of how computers can be used to understand and manipulate natural language in order to complete communication tasks. The field of NLP has evolved from traditional linguistics along with advances in computing and the increasing availability of large volumes of electronically stored natural language text. Classical linguistics aimed at devising rules of language by way of mathematical formalization. This was also the approach of most early NLP systems, which employed hand-crafted rules to process natural language. However, such approaches have significant limitations in capturing meaning and dealing with the variety of expression. Thus, a fundamental shift characterized by simple, robust approximations instead of deep analyses, a focus on probabilistic language models and the creation of large annotated bodies of text to be used in conjunction with statistics-based machine learning models [Kle] resulted in the emergence of the *statistical natural language processing field*, which defines the current NLP field in good measure.

The importance of the field is owed to the many applications it has in a variety of contexts. Generally, NLP methods have been designed for indexing and searching large texts, Information Retrieval (IR) and Information Extraction (IE), classification of text into categories, machine translation, automatic summarization, question answering, knowledge acquisition and text generation [Cho03]. More concretely, the relevance of such techniques is in the automation of certain tasks that allow people more time to dedicate to creative work. Additionally, such approaches can provide a different type of learning, based on large amounts of data that would be impossible for a human to process within reasonable time limits. For instance, text classification and knowledge acquisition techniques are involved in a number of fields: *finance* (e.g. money laundering detection, risk management, customer relationship management, fraud detection, report analysis [GDKS20]), *political science* (e.g. analyzing voting behavior and policy making, tracking international conflict [CJ20]), *clinical research* (e.g. clinical decision support, clinical trial matching, computational phenotyping [FRC13, FH+99]) or *education* (e.g. automatic scoring, intelligent tutoring systems, evaluation of student writing and knowledge [Bur09]).

Of these natural language processing applications, we focus on two text classification tasks: *emotion detection* and *authorship attribution*. For the first of these, *emotion detection*, we developed a series of resources for Romanian language, namely a lexicon [BL17, LB17, LB19] and two corpora, and used them in the study of emotional expression in Romanian literary texts [Bri19, LBB21]. For the second task, *authorship attribution*, we proposed a series of *deep learning* models to identify authors of Romanian poetry [BCL21] and evaluated the ability of similar natural language processing pipelines to solve the problem of authorship attribution of source code [CLB22, LBCC22]. Next, we will detail the motivation in tackling each of these tasks.

Approached problems

Emotion detection

Firstly, emotion recognition from text is an important task due to its wide range of use cases, from opinion mining in commercial contexts to psychological analysis in clinical settings. Emotion detection can also play a significant role in educational software that integrates and adapts with regard to emotional components of the interaction and, practically, any system or process that involves emotional content. As emotions are a vital part of an individual's make-up, and consequently a thread interwoven into human existence, the range of applications of automatic emotion detection and analysis is quite wide.

However, the popularity of this field is proportional to the difficulty the task of emotion detection implies. There are some key research challenges, among which the complex nature of emotion expression, shortage of quality data and limitations of current computational models [STZ18]. The first of these challenges encompasses a number of characteristics of emotion expression that make it very difficult not only for computers to properly assess emotional content, but also for humans, such as: the fact that the same emotion can be conveyed in a number of different manners, with contrasting vocabulary and constructs, including irony and sarcasm, figurative language or context-dependent references; the number of possible emotional meanings and facets that a single sentence can incorporate; and the inherent intricacy of emotion taxonomies - while several models of emotion exist, none manage to definitively organize and classify all emotions and the relationships between them. With respect to the shortage of quality data, this is of particular importance in data-driven models, such as most state-of-the-art deep-learning ones currently employed in the field. Annotation of emotional content is a notoriously difficult task [Öhm20], for a series of reasons: annotator perspective and its influence in labeling the emotional content, as personal experience dictates understanding of various contexts and the emotional responses they elicit, the emotional dynamics present in any given conversation and their different facets - for example, *writer/speaker* emotion and *reader* emotion, and the questions that remain surrounding the adequacy of applying psychological theories of human interaction like emotion models without modifications in what is a limited modality of communication. Lastly, the relative inefficiency of current models can be explained based on the last observation. Modeling the task of emotion detection is in itself a difficult task, as one can define it as a binary classification problem, a multi-class classification problem with one or multiple labels for a piece of text, or a regression problem (i.e. emotion intensity prediction). Moreover, the natural language processing field still has a long way to go in managing to solve the natural language understanding problem, as existing approaches only manage to capture semantic aspects of language to a limited extent. The emotional and semantic aspects of language are strongly linked, which makes the development of emotion-enriched text representations such as emotional word embeddings a crucial aspect of research.

Early, rule-based approaches to the emotion detection task have failed to address the intricacies of the problem in a satisfying manner. However, recent advances based on deep learning models encourage continuous exploration of this field, with the latest studies addressing many of these difficulties. Our research focuses on *Romanian language-specific emotion recognition* with the objective of contributing to the development and alignment of the national field with international standards. The choice of topic is motivated by the engaging nature of the task, which can be viewed as the development of precise systems and models for the engine of human behavior, namely, emotions, and, consequently, its wide-ranging implications with regard to human-like artificial intelligence.

As a first step in achieving this goal, we curated a Romanian Emotion Lexicon, named **RoE-**

moLex, which records associations between a series of words and eight basic emotions (*Anger, Anticipation, Disgust, Fear, Joy, Sadness, Surprise, Trust*) and two sentiment orientations (*Positivity* and *Negativity*). Next, we collected data in the form of Romanian texts, creating the **RoWikiLit** corpus for which we could examine the emotional content with the help of RoEmoLex. We chose literary works for a number of reasons, among which the fact that some are freely available. Moreover, for popular works, there are vast amounts of literary analyses, critic and audience reviews. Therefore, performance of computational models can be assessed using these additional sources, not unimportant in complex tasks such as emotion studies. We then used *unsupervised learning* models to investigate the relationship between linguistic aspects of literary texts and emotional content. Lastly, we have developed an emotion detection data set containing Romanian sentences and paragraphs annotated with 14 emotions and 2 valences (**RoEmoData**), which will be the building block of subsequent emotion detection experiments in Romanian text.

Authorship attribution

Authorship analysis is a domain of NLP which includes text data mining and classification with the aim of extracting information from a text about the writing style, author's sociolinguistic characteristics and, finally, identifying the text's author. There are several related tasks in this domain: (1) *Authorship Attribution (AA)* - identifying the author(s) of a set of texts, (2) *Author Profiling* (AP) - finding author's demographics, such as age, gender, occupation and educational level, (3) *Authorship Verification* (AV) - checking whether a text was written or not written by an author, *Plagiarism Detection* (PD) - searching for paragraphs reproduced from the texts of other authors. For instance, *software plagiarism* is a major problem in educational and corporate environments which may be tackled through authorship attribution [BT07] by identifying textual similarities between pieces of source code.

Uncovering the hidden characteristics of authors from a corpus of textual data has many societal applications with specific aims in various domains [DFIC19], [SMS17]: *literature and history* (to determine the paternity of disputed or anonymous literary and historical documents; to detect pastiche; to compare the styles of different authors); *education* (to understand the personality of students; to detect plagiarism in academic work [HLLA14]); *social network analysis* (to extract users' profiles such as identity, sociolinguistic characteristics and their opinions [MGAPD+17]); *cybercrime investigation* (to identify malicious activities such as spamming, ransom messages, harassment, money laundering, illegal material distribution in e-mails, social blogs, SMS-text messages [STASH19]; to provide evidence in courts of law); *software engineering and cybersecurity* (to identify the author of a given piece of source code [KKG+19]; to detect software plagiarism [BT07] and malicious code; to prevent cyber attacks).

The challenges encountered in the task of authorship attribution can be summarized as follows. First, there is a question regarding the reliability of techniques in settings in which accuracy is paramount (e.g. legal cases), which raises the issue of whether there truly exists a "human stylome" [VHBT+05], namely specific patterns of language usage that each individual employs. Secondly, there is no universally accepted solution for the authorship attribution task. Consequently, the many techniques presented in the field are shown to be working in small-scale experiments, but there is rarely cross-validation with data sets from other domains or a rigorous comparison with similar works. To this point, genre of texts and sample representativeness are crucial issues [Juo06]. For instance, a model trained on poetic data of an author might not be able to recognize his works in prose. Therefore, for use in real world contexts, it may be useful to develop cross-genre models. Lastly, modifications in an author's manuscript by editors or type-setters might make it difficult to separate

the author's *true* style in published works. Largely, in authorship attribution studies, a convention is made that the final, published form of the text belongs in its entirety to the author.

Code authorship identification, or *Software Authorship Attribution (SAA)*, can be defined as the process of identifying programmers in given pieces of source code, based on a prior determination with regards to their distinguishing characteristics as pertaining to *programming style*. The programming style of a developer can be characterized by a number of preferences and choices made in the coding process, including, but not limited to the naming of variables, the libraries, the data structures and the control sequences used. Programming style also has a direct link to the programmer's proficiency and experience, and, of course, the particular mode of logical and creative thinking employed to solve a programming task [ARA+19].

While it is true that programming languages have much less flexible grammars than natural languages, it is widely accepted that *a programmer's coding style* can still be identified. Coding style encompasses the programmer's preferences in the expression of logic constructs, the definition of data structures and their subsequent use, and the naming of variables and constants, as well as the calls made to fixed and temporary data sets [SAM96].

It is only recently that the field of source code authorship attribution has gained attraction. The growing interest in SAA is due to the practical needs in the academic, economic and societal fields. Plagiarism detection, ghostwriting (detection of outsourced student programming assignments) are specific tasks solved with SAA in the academic field. In the cybersecurity domain, in which both individuals and organizations are targets, the cyber-attacks based on malicious software (adware, spyware, virus, worms, and more) are important issues that can be prevented with the help of SAA systems. The software engineering field benefits from SAA in solving different tasks such as software maintenance, software quality analysis, project management, plagiarism detection with important effects in the copyright and licensing issues [BKR+21].

Our work consists of *deep learning* approaches to solve the tasks of *literary authorship attribution*, and *software authorship attribution*, respectively. We developed three models: $AutoAt$, for literary AA, $AutoSoft$ with its extension $AuoSoft^{ext}$, for a multi-class SAA classification task, and $SoftId$, for a one-class classification SAA task. The proposed approaches successfully solve the given tasks, comparing favorably with existing machine learning classifiers in multi-class settings. In addition, the models provide features such as the computation of class membership probabilities and the possibility of re-framing the systems to solve one-class classification authorship attribution tasks by recognizing "known" and "unknown" authors ($AutoSoft$ and $AutoSoft^{ext}$).

Original Contributions

Our research was focused on two main directions: (1) the development of *emotion detection resources* for the Romanian language and the study of relationships between various linguistic features and emotional content with the help of *unsupervised learning* techniques and (2) the investigation of *deep learning* models for the task of authorship attribution, for both natural language texts and for segments of software code. Thus, results and our main contributions are separated based on these directions, which are presented in detail in Chapters 2 and 3:

1. **Emotion Detection**

   The many interesting studies on emotion detection for the English language prompted an interest in the development of the aligned Romanian field. To this end, we created a series of resources for the Romanian language, and proposed three types of analyses of semantic-emotional content: (1) using simple quantitative measures and statistics, (2) using a mathe-

matical theory of data analysis called *Formal Concept Analysis* and (3) using an unsupervised learning method, namely *hierarchical agglomerative clustering*. The main goal of these analyses was to uncover emotional patterns and their relationship with corresponding semantic content. Our results regarding this line of research were the following:

(a) Our first step was to develop **RoEmoLex**, a word-emotion association lexicon. RoEmoLex was developed in three stages, documented in three published papers: the first version of the lexicon ([LB17]) consisted of a small number of terms and included slight corrections over the original translated resource (EmoLex [MT10]). In RoEmoLex v.2. ([BL17]) a series of new terms were added. The third version of the lexicon [LB19] underwent rigorous verification and an in depth analysis of its content in order to present a ready-to-use resource. Details about the third version of RoEmoLex are included in Section 2.1.1. RoEmoLex is also publicly available at `https://www.cs.ubbcluj.ro/~ica/romanian-nlp-resources/index.html`, and was already used in a series of studies in the psychology [HO20] and political science fields [Bir20].

(b) Besides providing a ready-to-use resource in RoEmoLex, we also proposed an exploration of the lexicon's content through Formal Concept Analysis, a method that allows an intuitive representation of semantic and lexical relations between terms and their associated valences and emotions. Results were published in [LB19]. We showed that equivalences between emotion-based formal concept orderings and hierarchical semantic representations of terms (e.g. *hypernymy/hyponymy*), and emotional tag distribution in semantically meaningful sets of words could be exploited in further applications integrating the RoEmoLex resource. Details about this study can be found in Section 2.2.1.

(c) In our next study, we used RoEmoLex to investigate the poetic work of a Romanian author (Mihai Eminescu) with regards to several lexical, semantic and emotional aspects. Results were published in [Bri19]. Details about this analysis are included in Section 2.2.2. The original contribution of this paper to the field was an in-depth quantitative analysis of the works of Mihai Eminescu and a close examination of the relationship between the phases of the author's artistic expression and quantifiable aspects of language. In particular, the emotional and valence features we defined were unique, and while results of the simple statistical analysis in this regard were not definitive, they informed our future studies.

(d) As the previous study only encompassed a quantitative analysis of a limited set of documents, we compiled a larger data set of literary texts, **RoWikiLit**, from online sources, a process described in Section 2.1.2. This data set was used in two of our studies, of which one involved the study of semantic and emotional patterns in an *unsupervised learning* manner in a labeled subset of Mihai Eminescu's poetry. Results were published in [LBB21], and the original contribution of this paper represented the application of an unsupervised machine learning technique to a poetry corpus in order to explore associations between thematic content and emotional patterns, which showed that some of literary scholars' findings can be replicated using computational techniques. Details of this study can be found in Section 2.2.3.

(e) Our current work in progress is based on the creation of a Romanian Emotion Detection Data Set (**RoEmoData**), which includes Romanian sentences and paragraphs from 4 different semantic categories (conversational and diary-like entries using literal language, text involving figurative language and Romanian idioms, literary fragments and news and article excerpts) annotated with 14 emotions and 2 valences via crowdsourcing. The current state of the data set is described in Section 2.1.3.

2. **Authorship attribution**

   Our second direction of research started with an initial question of whether authors of literary works could be automatically identified based on certain text representations. Thus, our goal was to define a novel deep learning model to solve this task. After a fairly successful experiment in defining ***AutoAt***, a deep *autoencoder-based* classification model for *supervised authorship attribution* of Romanian poetic data, we investigated the ability of *autoencoder-based* models to distinguish between authors of software code. During our research, the following three deep learning based models were developed:

   (a) ***AutoAt***. The first model we created was based on an ensemble of deep autoencoders, which we evaluated on a subset of the RoWikiLit data set. The main idea of this model was to train one autoencoder for each author in the given data set with the goal of encoding the underlying characteristics (both structural and conceptual) of the documents belonging to the same author. The results of this study were published in [BCL21], and they are also described in Section 3.1. We obtained an F-score of 0.81 ± 0.02, which compared favorably to existing classifiers in 5 out of 6 cases.

   (b) ***AutoSoft*** and ***AutoSoft^{ext}***. The base model, $AutoSoft$, is similar to $AutoAt$, designed with the goal of investigating the performance of an autoencoder-based supervised classification model for software authorship attribution. The representation of source code was inspired from the NLP domain, with `doc2vec` program embeddings proving to be able to encode developer coding styles. Moreover, we proposed the extension of the $AutoSoft$ classifier, named $AutoSoft^{ext}$, to address the recognition of an "unknown" class in addition to the given set of developer classes. The results obtained in this study were detailed in [CLB22], and are also included in Section 3.2. The two models were introduced as proof-of-concept and evaluated on subsets of the Google Code Jam data set, on which we obtained $F\text{-}score$ values ranging from 0.902 to 0.986 for the $AutoSoft$ classifier. The $AutoSoft$ model surpasses other classifiers in the literature in **69%** of the cases, analysis verified by a two tailed paired Wilcoxon signed-rank test [kin19, Goob], where a p-value of 0.01369 was obtained, confirming that the improvement achieved by $AutoSoft$ is statistically significant, at a significance level of $alpha = 0.05$. For the $AutoSoft^{ext}$ model, the best results were obtained in a test configuration setting using N-grams with $N = 5$. In 7 different binary classification tasks with *original* and unknown authors, we obtained $F\text{-}score$ values between 0.983 and 0.996, which compared favorably against One Class Support Vector Machines classifiers, as the latter only outperformed $AutoSoft^{ext}$ in terms of $Specificity$.

   (c) ***SoftId***. This model was developed as a one-class classification model trained to encode and recognize the programming style of a given set of software developers, designed with the goal of detecting whether a certain source code is authored by a developer from the given set or by an "unknown" software developer. As opposed to the previous two models, $SoftId$ is based on a single autoencoder trained on textual representations of source codes. The results of this study were detailed in [LBCC22]. In 24 experiments on Google Code Jam subsets for which we varied the type of representation and the N-gram used, $SoftId$ outperforms the One-Class Support Vector Machines (OSVM) classifier used for comparison in 95% of cases. A detailed description of our experiments and the subsequent discussion based on the results obtained is available in Section 3.3.

Thesis Structure

The structure of the thesis is as follows.

The first chapter includes the theoretical background for the tasks considered in our thesis and the corresponding literature review. This chapter is separated into three sections.

In Chapter 2 we present our work in emotion detection in Romanian text. This chapter has two main subsections, Section 2.1, which presents the resources developed for this task, and Section 2.2, which presents our studies involving unsupervised analysis of semantic and emotional content using the developed resources.

In the third chapter, we present our experiments using deep learning techniques in the form of autoencoders on literary data (Section 3.1) and source code (Sections 3.2 and 3.3).

Chapter 1

Background

This chapter provides a brief overview of related work concerning the type of resources and tasks proposed in the present thesis. In addition, brief descriptions of the algorithms employed are given.

The chapter is separated into three sections: first, a section that details related work with respect to the type of resources and tasks presented; secondly, a section in which task-specific theoretical background is provided with respect to emotion analysis and literary perspectives on Mihai Eminescu's work and, finally, a section in which the methods and techniques employed are described.

The first of these sections represents an overview of the existing work in the two main research directions the thesis addresses: emotion detection from text and authorship attribution. The internal structure of this section follows that of the thesis as a whole, the first subsections focusing on the related work with regards to resources for emotion analysis, unsupervised analysis of semantic-emotional content in words and texts, quantitative analysis of literary style and emotion-based clustering of texts, respectively. This covers the first research direction, emotion detection and analysis.

As far as the existence of resources for emotion analysis in Romanian text, both in terms of lexicons and corpora, a survey of the literature indicates that there are few works that target the Romanian language, compared to the abundance of resources for English. It follows, then, that there is a similar trend in terms of emotion-based applications that consider the Romanian language. Nonetheless, there are several Romanian research groups which have developed and continue to develop reliable resources [TBM14, PT18, CD21] for a variety of tasks. Our work fills the research gap with respect to emotion analysis from Romanian text, where there are less studies than, for example, sentiment analysis, and only recently have supervised models been proposed [CD21].

The last two subsections include descriptions of the two authorship attribution problems that were tackled - *literary authorship attribution* and *software authorship attribution*, and subsequent reviews of the existing literature for both tasks, and therefore provide the necessary background for the second research direction presented in the thesis, *authorship attribution*. The main contributions of our research in the well-documented task of authorship attribution are (1) the development of a general autoencoder-based model for authorship attribution, which proves to work both in literary [BCL21] and software [CLB22] settings and (2) the flexibility and adaptability of the developed models, which can be re-framed to address different formulations of the authorship attribution problem.

The second section of the chapter covers the task-specific theoretical background needed for our emotion analysis studies, specifically the model of emotion employed in the creation of resources - namely, Robert Plutchik's model of emotion [Plu80] and aspects of Mihai Eminescu's work, who is the author we focus on in our quantitative analysis of poetic style and emotion-based clustering. Linguistic style, phases of creation and topics in Mihai Eminescu's work are all addressed in this section.

Finally, the methods and techniques employed in our studies are presented. First, as a theoretical basis for our unsupervised analysis of semantic-emotional content in words, Formal Concept Analysis (FCA) is described. FCA represents a mathematical rendition of the philosophical understanding of concepts, where *concepts* are defined through sets of *objects*, *attributes* and *relations* between them ([GW97], [Wil05]). We use this technique to order and organize concepts considering RoEmoLex terms as objects and emotional labels as attributes.

Secondly, the hierarchical agglomerative clustering algorithm is presented. Hierarchical Agglomerative Clustering (HAC) is an unsupervised technique that outputs a hierarchy of clusters, providing information about how clusters on each level of hierarchy are formed out of the clusters on the previous level of the hierarchy. This algorithm was used in the emotion-based study of Mihai Eminescu's poems.

Then, distributed word and text representations are discussed, with a focus on the `word2vec` and `doc2vec` models. In this context, the term *embeddings* refers to the group of techniques designed to obtained real-valued vectors as representations for individual words and documents. These distributed vector representations for a given set of words are learned based on usage, from a given corpus, most often by using a neural network, which leads to representations of units with similar meanings to be closer together. `doc2vec` representations were used in the majority of our authorship attribution studies.

Lastly, the deep learning models *autoencoders*, are described. An autoencoder is composed of two neural networks (named encoder and decoder) which are trained to approximate the identity function (i.e. to reconstruct the input), thus being known as *self-supervised learning* techniques. All the proposed authorship attribution models - $AutoAt$, $AutoSoft$ and $AutoSoft^{ext}$, and $SoftId$ are based on autoencoders, both a number of them used as an ensemble ($AutoAt$ and $AutoSoft$), or a single autoencoder used on its own ($SoftId$).

Chapter 2

Developing resources and applications for emotion detection in Romanian text

The importance of emotion detection from text is owed to the multitude of applications that it allows in a number of different fields, including psychology, sociology and political science. The challenges in emotion detection are mainly due to the fact that existing theories of emotion encompass a broader understanding of affect that include cognitive processes, behavior and psychophysiological changes, and textual communication in itself can not incorporate all of them. Specifically, the ambiguity of written text (e.g. use of sarcasm, irony, figurative language, etc.), the variety of manners in which a given emotion can be expressed, and the number of interpretations that can be assigned to a text by a reader represent a few of these challenges. In light of this, the development of quality resources for the computational analysis of emotions expressed in text is crucial, as is the study of different types of text with their help. Such studies could offer new perspectives with regards to existing analyses (e.g. in digital humanities studies) or provide starting points in the automatic recognition of emotion in human-generated text (e.g. in social media content, clinical self reports, etc.).

In this chapter, our efforts in contributing to the developing field of emotion detection from Romanian text are described. Our work follows two research directions: firstly, creating Romanian language resources, and secondly, exploring associations between semantic and emotional content of words and text in an *unsupervised* manner, with the help of the developed resources.

The studies included in this chapter were also published in five original papers [[BL17, BL18, LB17, LB19, LBB21]]. We summarize the main original contributions presented in this chapter:

1. Section 2.1 details the process of developing three Romanian language emotion detection resources: **RoEmoLex**, **RoWikiLit** and **RoEmoData**.

    - The three-stage development process of **RoEmoLex** is described in Section 2.1.1. RoEmoLex is a word-emotion association lexicon containing 9177 terms annotated with eight emotions (*Anger, Anticipation, Disgust, Fear, Joy, Sadness, Surprise* and *Trust*) and two valences (*Positivity, Negativity*). The first two stages of its development were detailed in two published works [LB17, BL17], while in a third paper we have presented the publicly available version of the lexicon (RoEmoLex v.3.) along with a *formal concept analysis* of the lexicon content [LB19]. The aim of this lexicon was to provide a starting point in the task of emotion analysis in Romanian text, a goal which was met, as researchers have already begun to use RoEmoLex in studies from different fields.
    - Section 2.1.2 presents **RoWikiLit**, a corpus of Romanian literary works mainly collected from the Romanian Wikisource website and annotated with various metadata to facilitate

analysis. Three subsets of the RoWikiLit corpus were used in our studies: a subset of poetic works from the author Mihai Eminescu annotated with publishing date and phase of creation information [Bri19], a subset of Mihai Eminescu's poetry labeled with literary topics [LBB21], and a subset of poetic works of 8 Romanian authors [BCL21]. The data in this corpus was collected with the aim of providing researchers ready-to-use Romanian textual data which evades issues related to privacy and copyright, as all items included in the RoWikiLit corpus are freely available for use.

- Section 2.1.3 presents **RoEmoData**, a Romanian Emotion Detection Data Set created for *machine learning*- and *deep learning*-based emotion recognition experiments for Romanian texts. This section provides the details about the content of the data set, the annotation process and a brief analysis of the portion of the data set that is fully annotated . This represents our current research.

2. Section 2.2 presents our work in terms of unsupervised analysis of semantic-emotional content.

   - Section 2.2.1 includes a detailed description of the work in [LB19], which aims at examining the structure and content of RoEmoLex through Formal Concept Analysis, a knowledge discovery technique. Dependencies between words at the emotional level are discovered, conceptual hierarchies are built and comparisons with information from another Romanian lexical resource (RoWordNet) are made. The results encourage further research and the possibility to integrate RoEmoLex as a subcomponent of emotion analysis systems for Romanian text.
   - Section 2.2.2 presents a computational analysis of a representative Romanian poetry corpus, the poetic work of Mihai Eminescu. We have examined a series of features addressing vocabulary richness, language complexity and emotional content in each phase of artistic creation. Results show that for a series of measures, theoretical observations from the literary criticism field find correspondence in quantitative analysis, indicating that this approach, and visualization of results, in particular, could be used as support tool for interpretation. This study is published in [Bri19].
   - Section 2.2.3 presents an emotion-based analysis of Mihai Eminescu's work, based on a corpus of 131 selected poems, using *hierarchical clustering* with lexicon-based emotion features obtained using RoEmoLex. Associations between thematic content and emotional patterns were explored, with results showing that some of literary scholars' findings can be replicated using a computational technique. This study is published in the original paper [LBB21].

2.1 Resources for Romanian language

2.1.1 RoEmoLex

RoEmoLex (Romanian Emotion Lexicon) [LB17, LB19, BL17] is a resource developed for text-based emotion detection in Romanian language by curating a translation of an English lexicon. RoEmoLex contains 9177 terms annotated with eight primary emotions (*Anger, Anticipation, Disgust, Fear, Joy, Sadness, Surprise, Trust*) and two polarity tags (*Positivity, Negativity*).

The lexicon went through three stages of development that included correcting translations, re-annotating data, integration of other resources and addition of new terms, and alignment with RoWordNet synsets, described in detail in the the following subsections from the thesis.

2.1.2 RoWikiLit - A Romanian literary corpus

In his book, "Macroanalysis - Digital Methods & Literary History" [Joc13], Matthew Jockers explains the benefits of computationally oriented approaches to literary research. In arguing for a blended approach of close and distant reading - that is, traditional literary analysis and macro-scale analysis on digital corpora, he considers that computational methods are a new, more appropriate method for evidence gathering in literary research in the era of big data. He states that while "close readings and digital searching will continue to reveal nuggets [..], the deeper veins lie buried beneath the mass of gravel layered above", and "what are required are methods for aggregating and making sense out of both the nuggets and the tailings".

Category	Number of works	Subgenres	Author examples
poetry	2157	ballads, meditations, odes pastels, elegies, epigrams, satires	M. Eminescu, V. Alecsandri G. Coșbuc
fables	307	-	G. Asachi, G. Alexandrescu
articles	2130	political articles, literary criticism opinion pieces	M. Eminescu, I.L. Caragiale, G. Ibrăileanu
children's stories	162	fairytales, moral stories	P Ispirescu, I. Creangă
short stories	417	-	I. Slavici, L. Rebreanu, A. Holban B. Ștefănescu-Delavrancea, N. Gane
plays	57	comedies, tragedies	I.L Caragiale
novels	62	-	I. Slavici, A. Odobescu, L. Rebreanu, G. Mihăescu
other prose	569	essays, historical fragments	N. Iorga

Table 2.1: Literary corpus statistics (July 2021)

The current corpus contains **5861** works of **57** Romanian authors, and includes *poetry*, *prose* with a wide range of types (children's stories, historical texts, novels, short stories), *plays* and *letters* gathered from an online source [1]. An additional **147** stand-alone books and **65** volumes with poems, esssays, children's stories or plays were downloaded from an online bookshop that provides a series of free e-books [2]. Table 2.1 presents an overview of this dataset.

Two subsets of this dataset were used in published or to be published studies, both containing poetic data. The first was a subset of 131 poems of a single author (Mihai Eminescu) for which additional information was manually acquired in the form of literary topics [LBB21]. Secondly, a subset of the 8 most prolific Romanian poets in the dataset was used for an authorship attribution study.

	Authors							
	Alexandru Macedonski	George Coșbuc	George Topîrceanu	Ion Minulescu	Mihai Eminescu	Octavian Goga	Vasile Alecsandri	Ștefan O. Iosif
No. poems	190	212	113	159	366	181	186	164
No. tokens	39 403	124 809	31 525	35 380	182 270	37 761	72 025	30 870

Table 2.2: Description of Authorship Attribution subset

The utility of the literary corpus has been proven in the two applications discussed above, both in the sense of contributions to the computational study of literary works, and as a starting point

[1] https://ro.wikisource.org/wiki/Pagina_principală
[2] https://www.bestseller.md/ebooks.html?limba=130

in experiments for more general natural language processing tasks such as authorship attribution. However, these tasks focused on small subsets. Larger scale studies and the employment of different perspectives, for example the use of trajectories of emotion instead of flat emotional content, are still unexplored research areas for Romanian language that could prove compelling.

2.1.3 RoEmoData

RoEmoData (Romanian Emotion Detection Data Set) is a data set developed for emotion detection in text for the Romanian language. This dataset contains approximately 6K sentences and paragraphs represented by diary-like entries, sentences uttered in conversations, short excerpts from literary works of Romanian authors and a series of news titles and article excerpts. Table 2.3 describes the current form of the dataset.

Set	Total number of instances	Number of annotated instances	% Annotated
CDIA	2187	2187	100%
EXPR	2193	2193	100%
NEWS&ARTICLES	815	165	20.25%
LIT_FRAGMENTS	1340	165	12.31%
Total	**6535**	**4710**	**72.07%**

Table 2.3: Number of instances in emotionally labeled dataset (April 2022)

2.2 Unsupervised analysis of semantic-emotional content

The presentation from this section is based on our original works, published in [LB19] (Section 2.2.1), [Bri19] (Section 2.2.2) and [LBB21] (Section 2.2.3).

2.2.1 Formal concept analysis of a Romanian Emotion Lexicon

This section details the exploration of the content of RoEmoLex through Formal Concept Analysis, a method that allows an intuitive representation of semantic and lexical relations between terms and their associated valences and emotions. Results were published in [LB19]. The proposed study showed that equivalences between emotion-based formal concept orderings and hierarchical semantic representations of terms (e.g. *hypernymy/hyponymy*), and emotional tag distribution in semantically meaningful sets of words could be exploited in further applications integrating the RoEmoLex resource.

The main research questions that the study aimed to answer are: (1) Does a RoWordNet noun hierarchy based on the hypernymy/hyponymy relation match the conceptual hierarchy generated by the terms' emotional content (as identified by RoEmoLex)?; (2) Can emotional content delineate meaningful groups of terms in RoEmoLex in specific conceptual categories? and (3) How to introduce a relative emotional scoring method for terms based on the obtained formal concept hierarchies?

2.2.2 Quantitative Analysis of Style in Mihai Eminescu's Poetry

This section details our original work [Bri19]. Quantitative stylistic methods aim to express certain aspects of a text in numeric form, thus allowing the introduction of fast, powerful and accurate computational approaches for analysis. While in the case of literature, the validity and usefulness of such studies is highly controversial, one cannot deny the opportunities brought forward by computational methods: first, the exploration of large sets of documents in search of patterns otherwise difficult to discover by human readers; second, the possibility of opening up new perspectives by uncovering latent features of texts. In this study, we investigate the poetic work of one of the most important Romanian poets, Mihai Eminescu, through a variety of quantitative methods addressing lexical, morphological, semantic and emotional aspects of text. In this section, we propose a comparison between the results of the computational approach and established interpretations of Eminescu's work in order to assess the viability of computational methods in poetic style studies.

For the data considered, we collected 339 poems from an available online source[3]. Of these, only works excluded were those that could not be definitively associated with a publication year.

2.2.3 Emotion-based hierarchical clustering of poems

This section presents details of the work done in [LBB21]. This study focuses on a subset of 131 of Mihai Eminescu's poems from RoWikiLit (Section 2.1.2). This poem set is comprised of poems for which thematic content was manually extracted from literary analysis books and anthologies. The considered topics are: *time* (45 poems), *cosmogony* (13 poems), *genius' condition* (9 poems), *nature* (68 poems), *death* (32 poems), *vision about creation* (17 poems).

The goal of this study was to investigate a clustering-based approach to unsupervisedly mine emotional patterns in Mihai Eminescu's poetry. Lexicon-based emotion features are used for the clustering algorithm and resulting clusters are assessed with regard to manually added characteristics of poems in the form of literary themes.

The approach employed in this study consists of creating vector representations of poems based on emotional content by computing a score for each valence and emotion as shown in Equation 2.1, using RoEmoLex (see Section 2.1.1).

$$score(e, poem) = \frac{1}{l_{poem}} \cdot \sum_{t_i \in poem} 1_{t_i \in RoEmoLex_e} \tag{2.1}$$

[3] https://ro.wikisource.org/wiki/Autor:Mihai_Eminescu

Chapter 3

Enhancing the performance of authorship attribution using deep autoencoders

Authorship attribution is an important task in a variety of domains, such as literature, education, and software engineering and cybersecurity. There have been few computational studies of authorship attribution of Romanian text, and, similarly, a relatively small number of works that exploited NLP-inspired features in the task of software authorship attribution. This chapter presents thre *autoencoder*-based approaches for the task of authorship attribution: first, on literary texts (Section 3.1) and secondly, on software programs (Section 3.2 and Section 3.3). The proposed models represent original contributions to the field, and can be summarized as follows:

1. Section 3.1 details the original work in [BCL21]. The study proposes $AutoAt$, a deep autoencoder-based classification model to solve the *authorship attribution* task on a subset of the RoWikiLit data set of 1571 poems authored by 8 Romanian poets using a distributed document representation (`doc2vec`). The results obtained compare favorably with respect to other machine learning classifiers. Additionally, the formulation of the $AutoAt$ model allows for the computation of the probability that a test instance belongs to a given author class, which may be a useful property in a variety of authorship attribution applications.

2. Section 3.2 presents a supervised classification model, $AutoSoft$, for software authorship attribution as a proof of concept. The model is composed of an ensemble of autoencoders that are trained to encode and recognize the programming style of software developers. The distributed vector representation of documents used is `doc2vec`, and experiments are performed on subsets of the Google Code Jam data set. The obtained results empirically prove the hypothesis that *autoencoders* are able to capture, from a computational perspective, relevant knowledge about how developers are writing their code. We also propose an extension of $AutoSoft$, $AutoSoft^{ext}$ to recognize not only the classes of the original authors (developers) on which $AutoSoft$ was trained, but an "unknown" class as well. This study has been summarized in an original paper that is currently under review [CLB22].

3. Section 3.3 presents $SoftId$, an autoencoder-based one-class classification model that is trained to recognize the programming style of a given set of developers. We use textual representations for the Python source code, and experiments are performed on subsets of 3, 5 and 12 developers from the Google Code Jam data set. The performance of the proposed classifier was evaluated using a series of measures which proved that $SoftId$ successfully solves the task of

authorship identification, outperforming a *One-class Support Vector Machine* classifier in an overwhelming majority of testing configurations, with respect to all the performance measures, and especially $Specificity$.This study has been summarized in an original paper that has been accepted for publication [LBCC22].

3.1 Authorship attribution of literary texts

The study proposes $AutoAt$, a deep autoencoder-based classification model which exploits the ability of autoencoders to encode meaningful data patterns, to solve an *authorship attribution* task. Experiments are conducted on a data set of 1571 poems authored by 8 Romanian poets using a distributed document representation. The proposed approach obtains comparable or better results with respect to other machine learning classifiers. Additionally, the formulation of the $AutoAt$ model allows for the computation of the probability that a test instance belongs to a given author class, which may be a useful property in a variety of authorship attribution applications. This aspect and the fact that $AutoAt$ performs well in the difficult task of authorship attribution on poetic data without the step of feature engineering being informed by domain knowledge show that the proposed classifier is a general one, with potential to be used successfully in other fields.

The three research questions that this study aims to answer are:

RQ1 How to introduce a multi-class classification model based on an ensemble of deep autoencoders to supervisedly identify the author of a given text, based on the encoded structural and conceptual relationships between the documents written by the same author?

RQ2 What is the performance of the approach introduced for answering RQ1 for identifying the authors of Romanian poetry and how does it compare to the performance of similar classification models?

RQ3 What is the relevance of the document embedding representation of the poetic texts in discriminating among different authors?

3.2 Authorship attribution of source code: the $AutoSoft$ model

This study introduces a supervised classification model, $AutoSoft$, for software authorship attribution as a proof of concept. The model is composed of an ensemble of autoencoders that are trained to encode and recognize the programming style of software developers. Subsequently, $AutoSoft$ will predict the author of a certain source code fragment, based on the similarity between the given code and the information encoded (through the autoencoders) about each software developer. We are exploiting the ability of autoencoders to encode, through their latent representations, patterns about the coding-style of specific software developers. In this proposal the representation of the software programs is inspired from the Natural Language Processing (NLP) domain [TCMM09]. A program, processed as a text (a sequence of specific tokens), is represented as a distributed vector provided by a `doc2vec` model [LM14]. Experiments are performed on software programs collected from an international programming competition organized by Google and previously used in the software authorship attribution literature. The obtained results empirically prove the hypothesis that *autoencoders* are able to capture, from a computational perspective, relevant knowledge about how developers are writing their code. An additional strength of $AutoSoft$ is the fact that it can be extended to recognize not only the classes of the original authors (developers) on which it was trained, but an "unknown"

class as well. As far as we are aware of, the approach proposed in this study is new in the literature regarding software authorship attribution.

To summarize, the study is focused towards answering the following research questions:

RQ1 How to design a supervised classifier based on an ensemble of autoencoders for predicting the software developer that is likely to author a certain source code, considering the encoded coding-style for the developers?

RQ2 Does the proposed classifier improve the software authorship performance compared to conventional classifiers from the machine learning literature?

RQ3 Could such a classification model be used not only to recognize the classes of developers it was trained on, but to detect an "unknown" class as well?

3.3 Authorship attribution of source code: the $SoftId$ model

This study introduces a one-class classification model, named $SoftId$, for software authorship identification. The proposed model is composed of an *autoencoder* (Autoencoder (AE)) that is trained to encode and recognize the programming style of a given set of software developers. The trained $SoftId$ model will be able to detect if a certain source code is authored by a developer from the original set or by an "unknown" software developer (other than the ones from the original set). At the decision stage, the probability that the query source code was written by an author from the original set is computed based on how similar is the testing instance to the information encoded by the AE. In our proposal, the source codes are represented using text embedding techniques applied in the NLP domain [TCMM09]. The novelty of the proposed classifier resides in solving the software authorship identification task in an open-set configuration using a deep autoencoder and based on textual representations of the source codes. To the best of our knowledge, the $SoftId$ classifier is new in the literature regarding software authorship identification.

The goal of the research presented in this study is to answer the following research questions.

RQ1 How to design an autoencoder-based one-class classifier for solving the software authorship identification as an open-set-recognition problem?

RQ2 What is the relevance of the textual representation of source codes in discriminating between original (known) and other (unknown) software developers?

RQ3 Which of the two corpus-based representations, *term frequency - inverse document frequency)* (TF-IDF) or *Latent Semantic Indexing* (LSI), is better suited for our approach?

Conclusions

The field of natural language processing is becoming more and more important as the quantity of textual data available electronically increases, both online, in the form of social media data and document archives for information retrieval services, and in private databases such as those consisting of electronic medical records or legal documents. In each task from this domain, the main challenge refers to finding an efficient model of language which the computer can manipulate to solve tasks usually completed by humans. Numerous approaches, from rule-based definitions of language to probabilistic language models, have been proposed and evaluated in a variety of tasks.

In this thesis we focused on two particular natural language processing tasks, namely *emotion detection* and *authorship attribution*. In the case of *emotion detection*, the aim was to develop a series of resources to aid the development of solution for this task for Romanian language, and evaluate the quality and usefulness of these resources in studies regarding associations between semantic and emotional content of words and texts. As for *authorship attribution*, we aimed to provide general and flexible deep learning based solutions for the task of identifying authors of both literary works and software programs.

The proposed original studies were separated in two categories corresponding to these two tasks: first, we presented the emotion detection resources we developed for the Romanian language, and then detailed the unsupervised analyses of semantic-emotional content in which these resources were used; in the second part of the thesis, we presented three models for authorship attribution, one evaluated on literary texts, and two on data sets of documents representing source code.

With respect to emotion detection, first, we presented the state of the three resources developed for emotion recognition and analysis for the Romanian language: RoEmoLex, RoWikiLit and RoEmoData.

RoEmoLex is a lexicon that contains 9177 terms annotated with Plutchik's eight primary emotions (*Anger, Anticipation, Disgust, Fear, Joy, Sadness, Surprise, Trust*) and two polarity tags (*Positivity, Negativity*). This lexicon went through three stages of development, starting from a translation of an English emotion lexicon, and our contributions consisted in translation corrections, alignment with other Romanian natural language processing resources such as RoWordNet, revision and re-annotation of emotional and valence labels, and enrichment of the lexicon with new terms. The final version of RoEmoLex, RoEmoLex v.3., provides part-of-speech information for each term, as well as the correspondent RoWordNet synset id for 8286 entries. The utility of the proposed lexicon was highlighted both in our studies of emotional content in literary works, and in the work of other researchers, who used RoEmoLex in studies in fields such as political sciences and psychology.

RoWikiLit is a literary corpus containing works of representative Romanian authors which includes various types of Romanian literary text, such as short stories, novels, poetry, essays and fairy tales. This data set was initially compiled to aid in the study of emotional content of Romanian text, as data was freely available, and had the advantage of having associated literary analyses which could help in the interpretation of the results obtained with regards to semantic and emotional content. However, the data set also proved to be of use in other natural language processing tasks, such as

authorship attribution.

Finally, as far as resources, **RoEmoData**, an emotionally annotated corpus of Romanian sentences and paragraphs was proposed. The entries of the data set were annotated with respect to 16 emotion and valence categories via crowdsourcing. Analysis of annotation reliability revealed that with some modifications such as the removal of certain emotion labels, the data set is suitable for use in emotion detection tasks. Through an investigation into the structure of the data set, interesting observations with regards to emotional presence and intensity emerged, as well as subsequent relationships between primary and secondary emotion labels. This analysis will inform future emotion detection experiments, which is our current focus.

Secondly, within the same field of emotion analysis, we detailed the unsupervised learning-based approaches used in our studies.

The first of these studies referred to the analysis of emotional terms in RoEmoLex using Formal Concept Analysis, which led to the discovery of meaningful groups of words as far as emotional content is concerned (`EmoSynsets`) and to the proposal of a formula to provide relative scores for RoEmoLex terms that denote the intensity of the expressed emotion.

The second study represented a computational analysis of a representative Romanian poetry corpus, the poetic work of Mihai Eminescu. We examined a series of features addressing vocabulary richness, language complexity and emotional content in each phase of artistic creation. Results showed that for a series of measures, theoretical observations from the literary criticism field find correspondence in quantitative analysis, indicating that this approach, and visualization of results, in particular, could be used as support tool for interpretation.

Finally, the third study focused on an emotion-based hierarchical clustering of Romanian poetry, which explored the extent of emotional-semantic associations in a subset of 131 works from poet Mihai Eminescu. Results showed that there is a partial overlap between affective and thematic content, consistent with literary evaluations of the same works. Moreover, the fact that computational approaches have the advantage of being objective and replicable, with unsupervised techniques such as clustering representing a valuable tool in the exploration of literary works was highlighted. Nonetheless, no specific emotional patterns, as determined by the proposed method, could be fully associated with particular literary themes.

With respect to authorship attribution, we proposed three models based on deep learning.

First, a self-supervised technique was proposed for authorship attribution in poetic texts. $\boldsymbol{AutoAt}$, a deep autoencoder-based classification model which exploits the ability of autoencoders to encode meaningful data patterns was proposed to solve this task. Experiments were conducted on a data set of 1571 poems authored by 8 Romanian poets using a distributed document representation. The proposed approach obtained comparable or better results with respect to other machine learning classifiers. Additionally, the formulation of the $AutoAt$ model allowed for the computation of the probability that a test instance belongs to a given author class, which may be a useful property in a variety of authorship attribution applications. This aspect and the fact that $AutoAt$ performed well in the difficult task of authorship attribution on poetic data without the step of feature engineering being informed by domain knowledge showed that the proposed classifier is a general one, with potential to be used successfully in other fields.

Secondly, we examined the performance of a similar approach on a data set of software programs to solve the software authorship attribution task. The $\boldsymbol{AutoSoft}$ classification model was proposed. The representation of the software programs was inspired from the natural language processing domain, and using this representation, the deep autoencoders applied to `doc2vec` program embeddings proved to uncover relevant hidden features of the software programs, that successfully distinguished the authors/developers. Moreover, an extension of the $AutoSoft$ classifier, $\boldsymbol{AutoSoft^{ext}}$ was pro-

posed to identify "unknown" author instances, and obtained good results compared to existing one-class classification approaches.

Lastly, we proposed the ***SoftId*** model with the aim of solving the software authorship identification problem in an open-set configuration, having one original (known) class. $SoftId$ represents an autoencoder-based one-class classifier, which was trained to learn and encode the programming style of an original set of developers. At testing time it will distinguish between the codes written by original authors and those written by other authors, based on the similarity between the tested instance and the information encoded by the autoencoder. Experiments conducted on a subset of Python programs [Gooa] proved that $SoftId$ successfully solves the task of authorship identification in this context, outperforming *One-class SVM* classifier in an overwhelming majority of testing configurations. The performed experiments also highlighted the relevance of features (i.e., textual representation of source codes) used by $SoftId$ for distinguishing the author of a piece of code ("known" or "unknown").

Future research directions

For both directions of research, the work can be extended in a number of ways.

As far as developing resources for emotion detection from Romanian text, our main goal is to complete the annotation of data in the remaining categories of the **RoEmoData** data set. For the annotated data sets of RoEmoData, additional work is in order to obtain a useful and reliable resource, specifically: (1) augumentation of the data sets with instances belonging to a *Neutral/No Emotion* class, either by determining a suitable threshold emotion score on the given sets, and considering some RoEmoData instances as *neutral*, or by adding such instances from an external source; (2) elimination of ambiguous emotional tags (e.g. *Anticipation*) which have very low inter-rater correlation, and (3) definition of a suitable annotator score aggregation method.

Once these steps are completed, we aim to design deep learning models using the existing annotated data to solve the task of emotion recognition in Romanian text. While labeled data sets are the basis of the supervised approaches we envisage, we also plan to use RoEmoLex to create *emotion-enriched word embeddings*. With regards to unsupervised analysis of semantic-emotional content, we consider it an essential step in understanding the resources and data sets with which we work. As future work, we propose the standalone exploration of emotion in various types of literary texts from the RoWikiLit corpus, and the definition of finer-grained measures that capture the nuances of expression throughout such texts, such as trajectories of emotion across text windows.

With regards to authorship attribution, for the identification of authors in literary texts, we propose a combination of the implicit (hidden) features extracted by `doc2vec` and stylistic features specific to literary texts (for instance, features that capture phonic phenomena in poems: euphony, assonance, alliteration, rhyme and words properties: frequency distribution, vocabulary richness [PLTA15]), for poems' representation.

As for $AutoSoft$, the generality of the proposed autoencoder-based classifier was tested in a task of source code authorship attribution, for which we envision the following extensions: solving the co-authorship problem by adapting the classification decision of $AutoSoft^{ext}$ model, and the use the representation of software programs based on another NLP technique, Latent Semantic Indexing (LSI) [MM00], which is language-independent, in $AutoSoft$.

With regards to $SoftId$, future work is envisioned as carrying out experiments on data sets collected from software development teams for further evaluation of the $SoftId$ classifier. As additional directions to improve $SoftId$, alternative functions will be considered for computing the distance D between vectorial representations of the source code, such as the Euclidian distance.

Bibliography

[ARA+19] Mohammed Abuhamad, Ji-su Rhim, Tamer AbuHmed, Sana Ullah, Sanggil Kang, and DaeHun Nyang. Code authorship identification using convolutional neural networks. *Future Generation Computer Systems*, 95:104–115, 2019.

[BCL21] Anamaria Briciu, Gabriela Czibula, and Mihaiela Lupea. $AutoAt$: A deep autoencoder-based classification model for supervised authorship attribution. *Procedia Computer Science*, 192:119–128, 2021.

[Bir20] Daniel Biro. Emotions in the political discourse in Romania. A corpus-driven analysis of multiword expressions. *Bulletin of the Transilvania University of Braşov, Series IV: Philology & Cultural Studies*, 13(1):17–38, 2020.

[BKR+21] Egor Bogomolov, Vladimir Kovalenko, Yurii Rebryk, Alberto Bacchelli, and Timofey Bryksin. Authorship attribution of source code: A language-agnostic approach and applicability in software engineering. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 932–944, 2021.

[BL17] Anamaria Briciu and Mihaiela Lupea. RoEmoLex - a Romanian Emotion Lexicon. *Studia Universitatis Babeș-Bolyai Informatica*, 62(2):45–56, 2017.

[BL18] Anamaria Briciu and Mihaiela Lupea. Studying the language of mental illness in romanian social media. In *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 21–28. IEEE, 2018.

[Bri19] Anamaria Briciu. Quantitative analysis of style in Mihai Eminescu's poetry. *Studia Universitatis Babeș-Bolyai Informatica*, 64(2):80–95, 2019.

[BT07] Steven Burrows and Seyed M. M. Tahaghoghi. Source code authorship attribution using n-grams. In *Proceedings of the twelth Australasian document computing symposium, Melbourne, Australia, RMIT University*, pages 32–39, 2007.

[Bur09] Jill Burstein. Opportunities for natural language processing research in education. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 6–27, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[CD21] Alexandra Ciobotaru and Liviu P. Dinu. RED: A novel dataset for Romanian emotion detection from tweets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 291–300, Held Online, September 2021. INCOMA Ltd.

[Cho03] Gobinda G Chowdhury. Natural language processing. *Annual Review of Information Science and Technology (ARIST)*, 37:51–89, 2003.

[CJ20] Kakia Chatsiou and Slava Jankin. Deep learning for political science. *The SAGE Handbook of Research Methods in Political Science and International Relations*, 2020.

[CLB22] Gabriela Czibula, Mihaiela Lupea, and Anamaria Briciu. Enhancing the performance of software authorship attribution using an ensemble of deep autoencoders. *Mathematics, Special issue on Recent Advances in Artificial Intelligence and Machine Learning*, submitted, 2022.

[DFIC19] Steven HH Ding, Benjamin Fung, Farkhund Iqbal, and William K Cheung. Learning Stylometric Representations for Authorship Analysis. *IEEE Transactions on Cybernetics*, 49(1):107 – 121, 2019.

[FH+99] Carol Friedman, George Hripcsak, et al. Natural language processing and its future in medicine. *Acad Med*, 74(8):890–5, 1999.

[FRC13] Carol Friedman, Thomas C Rindflesch, and Milton Corn. Natural language processing: State of the art and prospects for significant progress. *Journal of biomedical informatics*, 46(5):765–773, 2013.

[GDKS20] Aaryan Gupta, Vinya Dengre, Hamza Abubakar Kheruwala, and Manan Shah. Comprehensive review of text-mining applications in finance. *Financial Innovation*, 6(1):1–25, 2020.

[Gooa] Google. Google Code Jam Competition. https://codingcompetitions.withgoogle.com/codejam. Online; accessed 15 September 2021.

[Goob] Google. Online Web Statistical Calculators. https://astatsa.com/WilcoxonTest/. Online; accessed 01 February 2022.

[GW97] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 1997.

[HLLA14] N.D. Hansen, C. Lioma, B. Larsen, and S. Alstrup. Temporal Context for Authorship Attribution. A Study of Danish Secondary Schools. *Multidisciplinary Information Retrieval. IRFC 2014. Lecture Notes in Computer Science*, 8849:22 – 40, 2014.

[HO20] Andreea Horoiţą and Adrian Opre. False memories: Romanian Deese-Roediger-Mcdermott lists of words. *Cognition, Brain, Behavior*, 24(2):163–186, 2020.

[Joc13] Matthew L Jockers. *Macroanalysis: Digital methods and literary history*. University of Illinois Press, 2013.

[Juo06] Patrick Juola. Authorship attribution. *Information Retrieval*, 1(3):233–334, 2006.

[kin19] Chapter 6 - Inferential Statistics III: Nonparametric Hypothesis Testing. In Andrew P. King and Robert J. Eckersley, editors, *Statistics for Biomedical Engineers and Scientists*, pages 119–145. Academic Press, 2019.

[KKG+19] Vaibhavi Kalgutkar, Ratinder Kaur, Hugo Gonzalez, Natalia Stakhanova, and Alina Matyukhina. Code Authorship Attribution: Methods and challenges. *ACM Computing Surveys (CSUR)*, 52(1):1 – 36, 2019.

[Kle] Dan Klein. CS 294-5: Statistical Natural Language Processing. https://people.eecs.berkeley.edu/~klein/cs294-5/index.html. Online; accessed 29 April 2022.

[LB17] Mihaiela Lupea and Anamaria Briciu. Formal Concept Analysis of a Romanian Emotion Lexicon. In *13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP 2017)*, pages 111–118. IEEE, 2017.

[LB19] Mihaiela Lupea and Anamaria Briciu. Studying emotions in Romanian words using Formal Concept Analysis. *Computer Speech & Language*, 57:128–145, 2019.

[LBB21] Mihaiela Lupea, Anamaria Briciu, and Elena Bostenaru. Emotion-based Hierarchical Clustering of Romanian Poetry. *Studies in Informatics and Control*, 30(1):109–118, 2021.

[LBCC22] Mihaiela Lupea, Anamaria Briciu, Istvan Gergely Czibula, and Gabriela Czibula. $SoftId$: An autoencoder-based one-class classification model for software authorship identification. In *Proceedings of the 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)*, page submitted, 2022.

[LM14] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014*, volume 32, pages 1188–1196, 2014.

[MGAPD+17] Ilia Markov, Helena Gómez-Adorno, Juan-Pablo Posadas-Durán, Grigori Sidorov, and Alexander Gelbukh. Author profiling with doc2vec neural network-based document embeddings. In Obdulia Pichardo-Lagunas and Sabino Miranda-Jiménez, editors, *Advances in Soft Computing*, pages 117–131, Cham, 2017. Springer International Publishing.

[MM00] Jonathan I. Maletic and Andrian Marcus. Using Latent Semantic Analysis to identify similarities in source code to support program understanding. In *Proceedings 12th IEEE internationals conference on tools with artificial intelligence. ICTAI 2000*, pages 46–53. IEEE, 2000.

[MT10] Saif Mohammad and Peter Turney. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 26–34, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[Öhm20] Emily Öhman. Emotion annotation: Rethinking emotion categorization. *CEUR Workshop Proceedings*, 2865:134–144, 2020.

[PLTA15] Ioan-Iovitz Popescu, Mihaiela Lupea, Doina Tătar, and Gabriel Altmann. *Quantitative Analysis of Poetic Texts*. De Gruyter Mouton, 2015.

[Plu80] Robert Plutchik. A general psychoevolutionary theory of emotion. *Emotion: Theory, Research, and Experience*, 1:3–33, 1980.

[PT18] Vasile Păiș and Dan Tufiș. Computing distributed representations of words using the CoRoLa corpus. *Proceedings of the Romanian Academy*, 19(2):403–409, 2018.

[SAM96] Philip Sallis, Asbjorn Aakjaer, and Stephen MacDonell. Software forensics: Old methods for a new science. In *Proceedings 1996 International Conference Software Engineering: Education and Practice*, pages 481–485. IEEE, 1996.

[SMS17] S. Swain, G. Mishra, and C. Sindhu. Recent approaches on Authorship Attribution techniques — An overview. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, pages 557–566, 2017.

[STASH19] Sicong Shao, Cihan Tunc, Amany Al-Shawi, and Salim Hariri. One-class Classification with Deep Autoencoder Neural Networks for Author Verification in Internet Relay Chat. In *Proceedings of 16th IEEE/ACS International Conference on Computer Systems and Applications*, pages 1–8, 2019.

[STZ18] Armin Seyeditabari, Narges Tabari, and Wlodek Zadrozny. Emotion detection in text: a review. *CoRR*, abs/1806.00674, 2018.

[TBM14] Dan Tufiș and Verginica Barbu Mititelu. *The Lexical Ontology for Romanian*, volume 48 of *Text, Speech and Language Technology*, pages 491–504. Springer, 2014.

[TCMM09] Doina Tătar, Gabriela Serban Czibula, Andreea Diana Mihis, and Rada Mihalcea. Textual entailment as a directional relation. *Journal of Research and Practice in Information Technology*, 41(1):53–64, 2009.

[VHBT+05] Hans Van Halteren, Harald Baayen, Fiona Tweedie, Marco Haverkort, and Anneke Neijt. New machine learning methods demonstrate the existence of a human stylome. *Journal of Quantitative Linguistics*, 12(1):65–77, 2005.

[Wil05] Rudolf Wille. Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. In Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors, *Formal Concept Analysis*, Lecture Notes in Computer Science, vol 3626, pages 1–33. Springer-Verlag, Berlin, Heidelberg, 2005.