mathematics

MDPI

Article

Machine-Learning-Based Approaches for Multi-Level Sentiment Analysis of Romanian Reviews

Anamaria Briciu [1,*], Alina-Delia Călin [1,*], Diana-Lucia Miholca [1,*], Cristiana Moroz-Dubenco [1], Vladiela Petrașcu [1] and George Dascălu [2]

[1] Department of Computer Science, Babeș-Bolyai University, 1 M. Kogalniceanu Street, 400084 Cluj-Napoca, Romania; cristiana.moroz@ubbcluj.ro (C.M.-D.); vladiela.petrascu@ubbcluj.ro (V.P.)
[2] T2 S.R.L., 35 Ceauș Firică Street, 145100 Roșiori de Vede, Romania; george.dascalu@t-2.srl
* Correspondence: anamaria.briciu@ubbcluj.ro (A.B.); alina.calin@ubbcluj.ro (A.-D.C.); diana.miholca@ubbcluj.ro (D.-L.M.)

Abstract: Sentiment analysis has increasingly gained significance in commercial settings, driven by the rising impact of reviews on purchase decision-making in recent years. This research conducts a thorough examination of the suitability of machine learning and deep learning approaches for sentiment analysis, using Romanian reviews as a case study, with the aim of gaining insights into their practical utility. A comprehensive, multi-level analysis is performed, covering the document, sentence, and aspect levels. The main contributions of the paper refer to the in-depth exploration of multiple sentiment analysis models at three different textual levels and the subsequent improvements brought with respect to these standard models. Additionally, a balanced dataset of Romanian reviews from twelve product categories is introduced. The results indicate that, at the document level, supervised deep learning techniques yield the best outcomes (specifically, a convolutional neural network model that obtains an AUC value of 0.93 for binary classification and a weighted average F1-score of 0.77 in a multi-class setting with 5 target classes), albeit with increased resource consumption. Favorable results are achieved at the sentence level, as well, despite the heightened complexity of sentiment identification. In this case, the best-performing model is logistic regression, for which a weighted average F1-score of 0.77 is obtained in a multi-class polarity classification task with three classes. Finally, at the aspect level, promising outcomes are observed in both aspect term extraction and aspect category detection tasks, in the form of coherent and easily interpretable word clusters, encouraging further exploration in the context of aspect-based sentiment analysis for the Romanian language.

Keywords: sentiment analysis; latent semantic indexing; machine learning; deep learning; CNN; dense embedding layer; aspect term extraction; aspect category detection; Romanian language

MSC: 68T50

Citation: Briciu, A.; Călin A.-D.; Miholca, D.-L.; Moroz-Dubenco, C.; Petrașcu, V.; Dascălu, G. Machine-Learning-Based Approaches for Multi-Level Sentiment Analysis of Romanian Reviews. *Mathematics* **2024**, *12*, 456. https://doi.org/10.3390/math12030456

Academic Editor: Ioannis G. Tsoulos

Received: 24 December 2023
Revised: 22 January 2024
Accepted: 23 January 2024
Published: 31 January 2024

1. Introduction

The increased prevalence of digital communication in recent years has amplified the importance of automatically extracting and assessing sentiment in textual data, with organizations and researchers engaged in exploration of models with this capability, that allow them to gain insights into customer preferences and pinpoint emerging trends. An especially relevant application domain for sentiment analysis (SA) research revolves around the examination of consumer product reviews, which have evolved into an integral component of the purchasing process. Given that reviews inherently consist of opinions and evaluations of products and often employ subjective language, there is significant potential for sentiment identification at multiple textual levels. This includes the assessment of overall product evaluations (document-level SA), finer-grained analysis that aims to capture shifts in sentiment within a document (sentence-level SA), and the exploration of targeted

sentiment, which involves identifying pairs of product features and the specific sentiments expressed in relation to these features (aspect-level SA).

This work presents an extensive examination of SA approaches for texts in Romanian, proposing an in-depth analysis at the document, sentence, and aspect levels, with the objective of filling a gap in the existing literature, which lacks multi-level investigation of datasets that hold commercial value for the Romanian language. Thus, the primary goal of this study is to assess the appropriateness of current machine learning and deep learning models for sentiment analysis in the context of the Romanian language, in order to acquire a comprehensive understanding of their viability for practical implementation in various business scenarios.

The original contributions of our study are as follows: (1) an in-depth exploration of SA models' performance at multiple textual levels for Romanian-language documents; (2) the introduction of a balanced dataset of Romanian reviews (structured in twelve different product categories), with five automatically assigned labels; and (3) improvements that we bring with respect to the standard models.

Below, we summarize the research questions we aim to answer within this paper.

RQ1 Is latent semantic indexing (LSI) in conjunction with conventional machine learning classifiers suitable for sentiment analysis of documents written in Romanian?

RQ2 Can deep-learned embedding-based approaches improve the performance of document- and/or sentence-level sentiment analysis, as opposed to classical natural language processing (NLP) embedding-based deep learning approaches?

RQ3 What is the relevance of different textual representations in the task of sentence polarity classification, and what impact do additional preprocessing steps have in this task?

RQ4 In terms of aspect extraction, is it feasible for a clustering methodology relying on learned word embeddings to delineate groups of words capable of serving as aspect categories identified within a given corpus of documents?

RQ5 How can the aspect categories discussed within a document be identified, if an aspect category is given through a set of words?

The rest of this paper is structured as follows. Section 2 includes a succinct description of the tasks addressed in this paper. A literature review on sentiment analysis models for the Romanian language and other related aspects is provided in Section 3. Section 4 is dedicated to the description of the methodology employed, while Section 5 presents the results we obtained. Additionally, we include a comparison of our approach with existing works in the literature and an analysis of the obtained results in Section 6. The last section, Section 7, contains conclusions and directions for future work.

2. Sentiment Analysis

Sentiment analysis is the area of research concerned with the computational study of people's opinions, sentiments, emotions, moods, and attitudes [1], and it involves a number of different tasks and perspectives. In this section, we include descriptions of the specific tasks from the sentiment analysis domain we have addressed in the present study.

2.1. Document-Level Sentiment Analysis (DLSA)

At the document level, sentiment analysis systems are concerned with identifying the overall sentiment from a given text. The assumption this task is based on is that a single opinion is expressed in the entire document. The advantage of simplicity in the definition of the problem has encouraged a substantial amount of work, especially in the early stages of exploration within the field.

In a machine learning and deep learning context, the DLSA task can be viewed as a classic text classification problem, in which the classes are represented by the sentiments/polarities [1]. The task can be formalized as a binary classification problem, in which the two classes are represented by the positive and negative polarities. There are various multi-class formulations of the sentiment analysis task in the literature. Most commonly, a third neutral class is considered besides the positive and negative ones to define a

three-class classification problem [2,3]. In cases in which finer-grained sentiment labels are available, the targeted classes are, usually, strongly negative, negative, neutral, positive, and strongly positive [4–6]. In this context, any features or models used in the traditional text classification tasks may be applied, or new, explicit sentiment-oriented features, such as the occurrence of words from a sentiment lexicon, may be introduced.

However, a main disadvantage of DLSA refers to the assumption that a document, regardless of its length, contains a single opinion, and, consequently, a single overarching sentiment is expressed. Evidently, this does not always hold. Thus, researchers have progressively shifted their focus towards more fine-grained types of analysis.

2.2. Sentence-Level Sentiment Analysis (SLSA)

The objective of sentence-level sentiment analysis is to ascertain the sentiment conveyed in a specific sentence [7].

The motivation behind SLSA stems from the recognition that a single document can contain diverse opinions with varying polarities. This is particularly evident in texts like reviews, where users may make positive evaluations and negative evaluations in the same review. For example, a review with an average number of stars in a defined rating system is almost guaranteed to comprise both. Additionally, it is not uncommon for reviews to include neutral and objective statements of fact. This task thus serves as a connection between DLSA and aspect-level sentiment analysis. It aims to offer a more comprehensive view of the sentiment expressed in a document, without the intention of identifying the exact entities and aspects that the sentiment is directed towards. When considering the level of complexity, it can be observed that, although sentences may be regarded as short documents (and, thus, the problem can be formalized in an identical manner as for DLSA), they possess significantly less content compared to full-length documents. Consequently, the process of categorization becomes more challenging [8].

2.3. Aspect-Based Sentiment Analysis (ABSA)

While it is crucial to obtain an understanding of user opinion through analysis at the document level, and decompose it further into a study at the sentence level, in reviews, users often make evaluations with respect to different aspects of a given product, where an aspect refers to a characteristic, behavior, or trait of a product or entity [9]. For instance, for mobile phones, aspect categories of interest to users, generally, are battery life, photo and video quality, sound, and performance. Thus, creating a system that provides a summary of opinion polarity with regard to each of these aspects would be of great use for both users, who could benefit from customized recommendations aligned with their preferences and priorities, and for businesses, who could pinpoint areas of improvement in their products or services and make targeted changes to enhance product quality and customer satisfaction.

Aspect-based sentiment analysis is defined as the problem of identifying aspect and sentiment elements from a given text (usually a sentence) and the dependencies between them, either separately or simultaneously [10]. There are four fundamental elements of ABSA: aspect terms (words or expressions that are explicitly included in the given text, and that refer to an aspect that is the target of an opinion), aspect categories (a unique aspect of the given entity that usually belongs to a small list of predefined characteristics that are of interest), opinion terms (expressions through which a sentiment is conveyed towards the targeted aspect), and sentiment polarity (generally, positive, negative, or neutral).

Separate tasks can be defined to identify each of these elements and their dependencies: aspect term extraction (ATE), aspect category detection (ACD), opinion term extraction (OTE), and aspect sentiment classification (ASC). The ATE task aims to identify the explicit expressions used to refer to aspects that are evaluated in a text [10]. If formulated as a supervised classification task, then the goal is to label the tokens of a sentence as referring to an aspect or not. Since this implies the existence of annotated data, which is scarce for most languages besides English, a significant number of works employ unsupervised approaches. In recent years, this type of approach has involved the use of word embeddings

in various self-supervised techniques enhanced with attention mechanisms to learn vector representations of aspects [11,12].

As for ACD, which aims to identify the discussed aspect categories for a given sentence, most state-of-the-art approaches formalize the task as a supervised text classification problem where a generally small set of predefined, domain-specific aspect categories represent the classes [13]. Unsupervised formulations often involve two steps: first extracting candidate aspect terms (the ATE task), and then grouping or mapping these terms to corresponding aspect categories. Manual assignment of labels to the obtained groups is a common practice in such approaches [11,14], but recent works [12,15] have proposed various methods to automate the process.

3. Related Work

This section presents an overview of recent SA approaches found in the literature, structured according to the distinct levels of sentiment analysis addressed by our study (document, sentence, and aspect) and focusing on those targeting the Romanian language.

With respect to sentiment analysis (and NLP tasks, in general), Romanian is known as an under-resourced language, with few comprehensive, publicly available datasets or corpora, as well as dedicated tools. As indicated by the LiRo benchmark and leaderboard platform [16], LaRoSeDa [17] is, to date, the only publicly available large corpus for sentiment analysis in Romanian. It consists of 15,000 positive and negative product reviews, extracted from an electronic commerce platform, that have been automatically labeled based on the number of associated stars. Although perfectly balanced (out of the total number of reviews, half being positive and the other half negative), the dataset is highly polarized, the great majority of positive reviews being rated five stars, while most of the negative ones, one star. Moreover, the authors admit that the labeling process is sub-optimal (as stars' numbers do not always faithfully reflect the associated polarity of a review), mentioning manual labeling or noise removal as future improvement tasks.

Regarding models, in recent years, transformer-based ones (both multi- and monolingual) have become the de facto standard within the NLP domain. BERT (bidirectional encoder representations from transformers) has been adopted as the baseline for transformer models, providing state-of-the-art results for various NLP tasks. For Romanian sentiment analysis, there are multi-lingual (mBERT [18], XML-RoBERTa [19]), and dedicated BERT-models available (Romanian BERT [20], RoBERT [21]), with the ones in the latter category performing better, due to their training on comprehensive language-specific datasets. In addition, approaches aimed at achieving higher performance on domain-specific analysis (such as JurBERT [22]) or at adapting the large-scale pretrained Romanian BERTs to computationally constrained environments (such as DistilBERT [23] or ALR-BERT [24]) have also been reported. When it comes to speed and efficiency, the multi-lingual, lightweight fastText [25] (also covering Romanian) is a popular alternative to multi-lingual BERTs, with the latter being more suited though for complex, data-intensive tasks.

In addition to the previously mentioned approaches, several research papers (detailed in the following) have reported the usage, improvement, or comparison of various classical and deep learning models, with the purpose of achieving similar or better results for SA in Romanian. As resulted from our investigation, most of the existing work has targeted the document level, with only a few studies explicitly covering the sentence- and aspect-based ones.

3.1. DLSA for Romanian

The papers mentioned in this subsection report experimenting with either only classical machine learning (ML) approaches, only deep learning (DL) ones, or both.

Within the first category, the work of Burlăcioiu et al. [26] aims to capture users' perceptions with respect to telecommunications and energy services, by analyzing 50,000 scraped reviews of mobile applications, offered by Romanian providers in these fields. They compare the results of five well-known SA models (logistic regression (LR), decision trees (DT),

k-nearest neighbors (kNN), support vector machines (SVM), and naïve Bayes (NB)) on a balanced, automatically labeled version of the dataset, using term frequency–inverse document frequency (TF-IDF) encoding [27]. The best accuracy is obtained by employing DT and SVM (79.5% on average for the two models), with the former achieving better time performance. Russu et al. [28] provide a solution for sentiment analysis at the document and aspect levels, considering unstructured documents written in Romanian. They employ two different methods for sentiment polarity classification: one using SentiWordnet [29] as a lexical resource, and one based on the use of the Bing search engine. The experiments are conducted on a perfectly balanced corpus, consisting of 1000 movie reviews written in Romanian (500 positive and 500 negative), manually extracted from several blogs and websites. The documents have been manually labeled, based on the individual scores assigned by the user (in the range [1–10]). To identify document-level polarity, the authors experiment with random forest (RF), kNN, NB, and SVM, the maximum precision values obtained being 81.8% (using SentiWordnet) and 79.2% (using Bing queries).

Regarding DL approaches, the authors of LaRoSeDa, Tache et al. [17], propose using self-organizing maps (SOM), instead of the classical k-means algorithm, for clustering word embeddings generated by either word2vec [30] or Romanian BERT. The top accuracy rate reported on test data is 90.90%, by employing BERT-bag of word embedding (BERT-BOWE). Echim et al. [31] aim to optimize well-known NLP models (convolutional neural network (CNN), long short-term memory (LSTM), bi-LSTM, gated recurrent unit (GRU), Bi-GRU) with the aid of capsule networks and adversarial training, the new approaches being used for satire detection and sentiment analysis in Romanian. For the latter task, they use the LaRoSeDa dataset, the best accuracy (99.08%) being obtained using the Bi-GRU model with RoBERT encoding and dataset augmentation.

Belonging to the category of combined ML approaches, there is the work of Neagu et al. [32], whose general purpose is building a multinomial classifier (negative/positive/neutral) to be used for inferring the polarity of Romanian tweets in a video-surveillance context. By using both classical (Bernoulli NB, SVM, RF, LR) and deep learning approaches (deep neural network (DNN), CNN, LSTM), together with different types of encodings (TF-IDF/doc2vec for classical ML and DNN, word2vec for CNN and LSTM), they argue that, by adapting the NLP pipeline to the specificity of the data, good results can be achieved even in the absence of a comprehensive Romanian dataset (their dataset consists of 15,000 tweets, translated from English). The best obtained accuracy (78%) has resulted from using Bernoulli NB with TF-IDF encoding, while the state-of-the-art value (81%) is provided by the multi-lingual BERT, with a training time penalty though. Istrati and Ciobotaru [33] report on creating a framework aimed at brands' monitoring and evaluation, based on the analysis of Romanian tweets, that includes an SA binomial classifier trained and tested on a corpus labeled by the authors. The data are preprocessed using four proposed pipelines, the resulting sets being used to train and test various ML models, both classical and modern. The best accuracy and F1-scores are achieved by using a neural network with fastText [25], that being the model chosen for the framework classifier. Coita et al. [34] use SA in order to assess the attitude of Romanian taxpayers towards the fiscal system. In this respect, they try to predict the polarity of each of the answers provided by around 700 respondents to a 3-item questionnaire, using a BERT model pretrained and tested on a corpus of around 38,000 movie and product reviews in Romanian. BERT is chosen, as it provides maximum accuracy (98%) among several compared models, namely itself, recurrent neural network (RNN), and three classical ML approaches: LR, DT, and SVM.

3.2. SLSA for Romanian

Buzea et al. [35] introduce a novel sentence-level SA approach for Romanian, using a semi-supervised ML system based on a taxonomy of words that express emotions. Three classes of emotions are taken into account (positive, negative, and neutral). The obtained results are compared to those provided by classical ML algorithms, such as DT, SVM, and NB. Experiments are conducted using a corpus of around 26,000 manually annotated

news items from Romanian online publishers and more than 42,000 labeled words from the Romanian dictionary. In terms of F1-score, the proposed system outperforms the three classical algorithms for the neutral and negative classes, while for the positive class, the highest metric value is achieved by DT.

Using a custom-made application, Roşca and Ariciu [36] aim to evaluate the performance of the Azure Sentiment Analysis service at sentence level for five languages, including Romanian. With this purpose, they generate 100 sentences per language, half positive and the other half negative. Although the service performs SA using three sentiment classes (positive, negative, and neutral), their evaluation only considers the first two, assuming any neutral label as incorrect. Classification accuracy is computed for three types of sentences: shorter than 100 characters, in the range of 100–250 characters, and longer. The reported accuracies are 83% for the first and last categories and 90% for the middle one.

3.3. Aspect Term Extraction (ATE) and Aspect Category Detection (ACD)

The only work that proposes a complete ABSA system for the Romanian language is that of Russu et al. [28], who also aim to identify sentiment at the document level, as described in Section 3.1. In this paper, the authors use seven syntactic rules to identify aspect terms and opinion words in a set of movie reviews. The polarity associated with the discovered entity is computed either using SentiWordnet or a search engine, using a set of seed words.

In this context, we provide a succinct description of unsupervised approaches for the ATE and ACD tasks, which are the two ABSA tasks we address in this paper.

For the task of aspect term extraction, early unsupervised approaches were generally based on rules [37–39]. For instance, Hu and Liu [37] use an association mining approach to identify product features and a WordNet-based approach to predict the orientation of opinion sentences. Other works propose analyzing the syntactic structure of a sentence at the word or phrase level to identify aspects and aspect-word/sentiment-word relations [39]. Such rule-based approaches are also frequently employed for aspect category detection. Hai et al. [40] attempt to find features (aspects) expressed implicitly in text through a two-step co-occurrence association rule mining approach. In the first phase, the co-occurrence is computed for opinion words and explicit features, extracted from a set of cell phone reviews in Chinese, and they refer to verbs and adjectives, and nouns and noun phrases, respectively. Additional constraints based on syntactic dependencies are applied for the extraction. In the second step, a k-means clustering algorithm is applied to the identified rule consequences, which are the explicit aspects, to generate more robust rules that can be then used for implicit aspect identification. Schouten et al. [41] propose a similar co-occurrence-based approach, but their unsupervised model uses a set of seed words for the considered aspect categories.

Another type of unsupervised approach to these tasks is represented by variants of classic topic modeling techniques. Titov and McDonald [42], for example, propose a multi-grain topic model (MG-LDA), which aims to capture two types of topics, global and local, and pinpoint rateable aspects to be modeled by the latter, the local topics. Brody and Elhadad propose the use of a standard LDA algorithm, but treat each sentence as a separate document to guide the model towards aspects of interest to the user, rather than global topics present in the corpus [43]. A topic modeling approach is also proposed by García-Pablos et al. [44], but it is a hybrid one, also making use of word embeddings and a Maximum Entropy classifier to tackle ABSA tasks.

In terms of neural models, He et al. [11] rely on word embeddings in the context of an attention-based approach, through which aspect embeddings are learned by a neural network similar to an auto-encoder. Tukens and van Cranenburgh [15] propose a simple two-step technique for aspect extraction, which first selects candidate aspects in the form of nouns with the help of a part-of-speech (PoS) tagger, and then employs contrastive attention to select aspects.

While there are approaches that rely mainly on clustering techniques, they are less frequent. An example of a clustering-based approach is that of Ghadery et al. [45], who use k-means clustering on representations of sentences obtained by averaging word2vec embeddings and a soft cosine similarity measure, to determine the similarity between a sentence and an aspect category, represented by a set of seed words.

As far as word clustering is concerned, the identification of semantically meaningful groups in a vocabulary has been a topic of interest for decades. Recent approaches either focus on using word clustering to detect topics in a document [46–48], or use it as a technique to enhance the performance of classifiers by means of improved document representations [17]. Sia et al. [46] explore the ability of embedding-based word clusters to summarize relevant topics from a corpus of documents. Different types of word embeddings are examined, both contextualized and non-contextualized, along with a number of hard (k-means, spherical k-means, k-medoids) and soft (Gaussian mixture models and von Mises–Fisher Models) clustering techniques to identify topics in documents. CluWords, the model proposed in [47], is shown to advance the state-of-the-art in topic modeling by exploiting neighborhoods in the embedding space to obtain sets of similar terms (i.e., meta-words/CluWords), which, in turn, are used in document representations with a novel TF-IDF strategy designed specifically for weighting the meta-words.

4. Methodology

4.1. Case Study

This section describes the dataset used in our study, a new dataset comprising reviews written in Romanian. We start by providing a brief summary of the data collection process and our motivation in creating the RoProductReviews dataset, and then we present a detailed description of its content, highlighting its suitability for the proposed tasks.

4.1.1. Data Collection

The reviews that make up the RoProductReviews dataset were manually collected from a highly popular Romanian e-commerce website. Specifically, the gathered information consists of the text of the review, the title, and the associated number of stars, which ranges between 1 and 5, and can be viewed as a numerical representation of customers' satisfaction with the reviewed product. In this context, assigning 1 star to a review represents complete dissatisfaction, while a 5-star evaluation indicates complete satisfaction with the product. Reviews were collected for a total of 12 product categories of electronics and appliances. The only criteria used in selecting reviews were the number of associated stars and the length of the text: the first, in terms of having a balanced dataset on the whole with respect to positive and negative sentiment, as we planned to use supervised learning techniques for the task of sentiment analysis, and the second, with the ABSA task in mind, reviews with longer texts were sought out to be included along with short, one-sentence reviews, since, generally, in the longer reviews, discussions about specific aspects of the product are included.

Through this data collection process, we built a balanced dataset with reviews written between 2014 and 2023 that is representative of the various modes of expression encountered in e-commerce product evaluations. To prevent the introduction of bias, ten individuals with diverse backgrounds collected the data. Clear guidelines outlining the purpose and intended structure of the dataset were provided to ensure consistency.

4.1.2. Dataset Description

Table 1 presents the number of reviews in the dataset associated with each number of stars, as well as the number of sentences they consist of. Additionally, the total number of tokens, the number of unique tokens, and unique lemmas are included for each category, as well as the average sentence length, computed as the average number of words in a sentence.

Table 1. RoProductReviews statistics.

	Number of Reviews	Number of Sentences	Average Sentence Length	Number of Tokens	Number of Unique Tokens	Number of Unique Lemmas
1 star	1357	3574	16.51	67,188	7669	5547
2 stars	1152	3873	18.39	81,120	9152	6840
3 stars	1280	4014	18.54	84,997	9691	6984
4 stars	1309	3621	17.43	72,305	8671	6203
5 stars	1336	2869	14.03	46,282	6123	4436

The RoProductReviews dataset is utilized in its entirety, as presented in Table 1, for the document-level sentiment analysis tasks. The classification labels for this dataset consist of either the assigned number of stars (for multi-class classification) or a positive/negative label derived from aggregating the higher- and lower-rated reviews, respectively (i.e., reviews with ratings of 1 and 2 stars are considered negative, while reviews associated with 4 and 5 stars are deemed positive; reviews with a 3-star rating are discarded in this setting). Although there is a possibility that the labels as obtained do not always faithfully reflect the sentiment expressed in the review [17], we consider them sufficient in terms of the intended experiments at the document level.

Nevertheless, when it comes to classifying sentiment at the sentence level, the rating assigned to the review that contains the sentences is an inaccurate predictor of the sentiment being communicated. Hence, a manual annotation procedure was utilized for a specific subset of RoProductReviews. A total of 2067 short reviews, consisting of single sentences, were annotated by 5 annotators who were only presented with the text of the review, but not the number of stars associated with it. A sentiment label was assigned if it was agreed upon by the majority; otherwise, the instance was discarded. Limitations exist in the annotation process, primarily inherent to sentiment annotation. Specifically, we emphasize the challenge of accurately identifying sentiment in extremely short sentences lacking explicit sentiment words or featuring ambiguous language. Additionally, annotators may delineate between neutral, positive, and negative sentiment differently, resulting in conflicting label assignments for the same sentence. To address these limitations, the annotation process incorporates majority voting, mitigating the impact of these challenges.

The reviews were chosen due to the fact that they did not require any additional processing in terms of sentence segmentation. The annotators utilized a labeling system that consisted of three categories: negative, neutral, and positive. As a consequence, a subset consisting of 804 reviews (sentences) annotated with the label negative was obtained. Additionally, there were 171 reviews annotated with the label neutral and 1092 reviews annotated with the label positive. A series of examples from this subset of RoProductReviews is included in Table 2.

Generally, RoProductReviews is characterized by a relatively equitable distribution among the various rating categories, with the exception of the 2-star category, which shows a lower level of representation. This under-representation of reviews in the 2-star category can be attributed to data availability constraints. During the data collection process, there was a noticeable scarcity of 2-star ratings, with a significant portion of unfavorable reviews predominantly attributed to a 1-star rating. It is plausible that customers articulating adverse sentiments may encounter challenges in acknowledging positive aspects of the reviewed product, which, in turn, might result in a milder form of negative evaluation, namely, a 2-star rating.

Table 2. Examples of manually annotated one-sentence reviews.

Review Text	Product Category	Number of Stars	Label
Asa cum m-am asteptat…face treaba pt birou *As expected…it does the job for the office.*	Monitor	5	Positive
Funcționează bine, mulțumit deocamdată de el *It works well, satisfied with it for now.*	Smartwatch	5	Positive
E un router ok *It's an ok router*	Router	4	Positive
NU E ULTRA SUPER CALITATE DAR E BUN *It's not ultra-super quality, but it's good*	Speakers	4	Positive
Este doar bună pentru jocuri şi desene, pozele ies ca pe telefoanele mai vechi *It's only good for games and drawings; the photos come out like on older phones*	Tablet	3	Neutral
Sunt acceptabile la redarea sunetului, dar la convorbiri nu prea se aude microfonul *They are acceptable for sound playback, but the microphone is not very audible during calls*	Headphones	3	Neutral
Mi s-a blocat de nenumărate ori și pierdea des semnalul *It has frozen numerous times, and it often lost the signal*	Smartphone	2	Negative
Nu ține deloc bateria, după nici 12 ore de la încărcarea completă (100%) s-a descărcat complet *The battery doesn't hold at all; after not even 12 h from a full (100%) charge, it completely discharged*	Fitness bracelet	1	Negative
Cel mai silențios mouse, dar conexiune prin infraroșu mediocră, se întrerupe non-stop *The quietest mouse, but with a mediocre infrared connection, it keeps disconnecting non-stop*	Mouse	1	Negative
Procesor slab rău *Terribly weak processor*	Laptop	1	Negative

Regarding sentence length, we can observe that sentences in the rating categories that do not indicate complete satisfaction or dissatisfaction with the reviewed product (i.e., 2-, 3-, and 4-star categories) tend to be longer. This is intuitive, as in these cases, customers are more likely to provide detailed accounts of both the strengths and weaknesses of the product to justify their assigned rating. This is especially evident in reviews associated with 3 stars, an evaluation customers generally make after careful analysis of a series of positive and negative aspects of the reviewed product. Alternatively, 1-star and 5-star reviews may only consist of short sentences such as "*Nu recomand/Don't recommend*", "*Calitate proasta/Bad quality*", "*Slab/Weak*" and "*Super/Super*", "*Tableta excelentă/Excellent tablet*", "*Multumit de achizitie/Content with my purchase*", respectively.

Table 3 presents analogous statistics, this time segmented by product category. The dataset exhibits diversity in terms of the number of reviews gathered for each category. For instance, there is a nearly threefold difference in the number of reviews collected for *smartphones* compared to *routers*.This diversity is essential for creating a realistic evaluation scenario for various sentiment analysis models directed toward specific product categories (e.g., aspect-based sentiment analysis), reflecting the real-world scenario where certain product types enjoy more popularity and consequently accumulate more reviews than others.

Table 3. RoProductReviews dataset description per category.

Product Category	Number of Reviews	Number of Sentences	Average Sentence Length	Total Number of Tokens	Number of Unique Tokens	Number of Unique Lemmas
Headphones	409	984	16.54	18,480	2990	2205
Fitness bracelets	599	1578	16.45	29,500	4117	2967
Keyboard	899	2522	17.71	51,160	5856	4249
Laptop	404	1313	16.70	25,005	4070	3038
Monitor	419	971	15.98	17,852	3061	2328
Mouse	395	1062	16.69	20,383	3157	2298
Router	300	853	17.25	16,785	2919	2248
Smartphone	897	2348	17.16	45,906	6432	4882
Smartwatch	577	1469	17.38	29,213	4285	3109
Speakers	455	1429	18.59	30,240	4325	3163
Tablet	680	1753	15.59	31,445	4440	3291
Vacuum cleaner	400	1669	18.88	35,923	4842	3418
TOTAL	6434	17,951	17.08	351,892	21,430	15,311

We note that, despite this imbalance across product categories, the distribution of reviews in each star rating category is preserved. With a few exceptions (*monitor, tablet, smartphone*), the sets are almost perfectly balanced in this respect.

Additionally, we present a series of statistics that further support the use of the RoProductReviews dataset for the sentiment analysis tasks addressed in this study. Specifically, to provide context for the aspect identification task, which relies on the identification and grouping of nouns, we computed the part-of-speech distribution within each product category with the help of the NLP-Cube Part of Speech Tagging Tool [49]. We found that nouns represent approximately 20% of all tokens for every category. The percentage of adjectives ranges from 0.04 to 0.06, with *vacuum cleaner* reviews having the smallest proportion and *monitors*, the highest. Alternatively, *vacuum cleaner* reviews are the richest in terms of verbs (0.14), while reviews about peripherals, like *monitors* and *keyboards*, have the smallest proportion of verbs, along with *routers* (0.11). Similarly, small differences are observed with respect to adverbs: the highest percentage of adverbs can be found in *headphones* reviews (0.12), with *router* at the other end (0.09). The notable presence of nouns in reviews provides a favorable foundation for our proposed approach to aspect identification, which relies on noun clustering, but underscores the necessity of devising an effective method for discerning the most relevant nouns. As for adjectives, a part-of-speech traditionally linked with sentiment, we note that their relatively low presence may be due to users often expressing sentiment with regard to products by stating what works and what does not (e.g., *I can't run multiple applications simultaneously*), or by providing domain-specific clues (e.g., *the refresh rate is 144 Hz, and it shows*). Nonetheless, out of all adjectives, between 38% and 50% are valenced across categories (as identified by the lexicon RoEmoLex [50]), which lends credit to the possibility of exploring dependency-based approaches to associating sentiment with the aspect terms discovered through nouns. Interestingly, around 20–25% of verbs in each category are also found in the sentiment lexicon, while only about 13–17% of nouns and 5–6% of adverbs are used to express sentiment directly.

In view of this analysis, we consider that the proposed dataset is suitable for a case study that aims to examine the appropriateness of different machine learning and deep learning models for sentiment analysis for the Romanian language.

4.2. Theoretical Models

This section includes the formalization of the sentiment analysis tasks at each level, which target $\mathcal{D}$, a collection of documents that, in our case study, refers to the RoProductReviews dataset. Each $doc \in \mathcal{D}$, where $doc = \{w_1, w_2, \ldots, w_N\}$ represents a document from the collection comprising N words, and w_i with $1 \leq i \leq N$ is a word in the document. Let $\mathcal{V}$ be the vocabulary used in this collection, defined as:

$$\mathcal{V} = \bigcup_{doc \in \mathcal{D}} doc \tag{1}$$

Additionally, we denote by $\mathcal{D}_c$ the collection of documents in a given product category c.

4.2.1. Document-Level Sentiment Analysis

The task of sentiment analysis at the document level assumes that the document doc (for example, a movie review or, as in our case, a product review) expresses an opinion regarding a specific (single) entity e. In this context, document sentiment classification aims to determine the overall sentiment s expressed related to the entity e, which can be positive or negative (in binary classification). The sentiment options, however, can be extended to a range, in our case, the five stars ranging from 1 (strongly negative) to 5 (strongly positive), leading to a multi-class classification problem [1].

4.2.2. Sentence-Level Sentiment Analysis

Sentence-level sentiment analysis assumes that each sentence st expresses a single opinion, oriented towards a single known entity e. Therefore, the goal of classification at the sentence level is to identify the sentiment s expressed in sentence st regarding the entity e. Since reviews, by definition, express opinions about a product or service, it is expected that at least one of the multiple sentences in a document expresses a positive or negative opinion. This is why document-level analysis can ignore the neutral class, but sentence-level analysis cannot: a sentence within a review can be objective, which means that it does not express any sentiment or opinion and is therefore neutral [1].

4.2.3. Aspect Term Extraction and Aspect Category Detection

We address the aspect term extraction task through an examination of word embeddings and their subsequent properties in the learned vector space. We build on previous research that indicates that aspects are explicitly referred to in texts through nouns [15,37,40], and employ a clustering algorithm to obtain groups of similar words, particularly nouns, that are interpreted as aspect categories.This analysis serves as an initial step for addressing the ABSA task, which currently lacks extensive exploration in the context of the Romanian language. We also provide a method to estimate the presence of an aspect in a document (sentence/review), thus addressing the aspect category detection task, highlighting its potential for application at both the document and sentence levels.

Let $\mathcal{N}_c$ be the set of nouns used in a category c. A clustering algorithm is applied on the set $E_c = \{embedding_w | embedding_w = f_{model}(w), w \in \mathcal{N}_c\}$, which contains the embeddings obtained through embedding model f_{model} for the nouns used throughout documents in category c. A partition $\mathcal{P}$ of set E_c is thus generated, with $\mathcal{A} \in \mathcal{P}$ a set of similar embeddings, where for similarity, a suitable metric is chosen.

The sets $\mathcal{A}_w, \mathcal{A}_w = \{w | embedding_w \in \mathcal{A}\}$ represent candidate aspect categories and their members, candidate aspect terms. To obtain the most relevant aspects from each product category, we apply the following heuristic: we eliminate from consideration sets $\mathcal{A}$ for which $|\mathcal{A}| < 3$ and $|\mathcal{A}| > 10$, where $|\mathcal{A}|$ represents the number of elements in set $\mathcal{A}$. We based this decision on the potential interpretability of such word groups: less than three words might not provide sufficient information for identifying an overarching aspect category, while a group of more than ten words will most likely contain miscellaneous terms with respect to semantic information, especially when considering the restricted vocabulary of only nouns. Then, we rank the remaining sets $\mathcal{A}$ to obtain the most representative groups with respect to the considered product category. Each set $\mathcal{A}_w$ is associated with a value defined as $score_{freq} = \sum_{w \in \mathcal{A}_w} \sum_{doc \in \mathcal{D}_c} freq(w, doc)$, which considers the overall frequency of the nouns in set $\mathcal{A}$ in the considered reviews $doc \in \mathcal{D}_c$ from a given product category c. We also experimented with a ranking based on the number of documents covered by the words in the obtained sets, with $score_{coverage} = |\bigcup_{w \in \mathcal{A}_w} \{doc | w \in doc, doc \in \mathcal{D}_c\}|$, and obtained similar results. Then, according to the ranking given by one of these scores, the top t percent groups are considered the most relevant aspect categories, as, according to the ranking, these are the most frequently discussed in the given category. A short, descriptive label is assigned manually to each of these clusters based on its content.

4.3. Data Representation

4.3.1. Preliminaries: Data Preparation and Preprocessing

This section describes the preprocessing steps taken for each of the proposed analyses.

In all cases, a preprocessing step was performed, which involved the transformation of the text to lowercase and the removal of URL links. For stop word removal, the list provided with the advertools library version 0.13.5 (https://github.com/eliasdabbas/advertools (accessed on 20 January 2024)) was used, from which the words that may express opinions or sentiments, such as *bine* (well), *bună* (good), or *frumos* (beautiful), were removed.

In the approach at the document level, the title of the review was concatenated at the beginning of the review text to be classified. Moreover, the stop words were not removed, to avoid loss of information relevant to the model and to be able to perform a baseline comparison. Also, punctuation was not removed because there were several emoticons which were punctuation based (and not Unicode characters).

For the sentence-level approach, the title of the review was not taken into consideration, as it usually contains two or three words, summarizing the review without forming a sentence. Similar to the document-level approach, the punctuation was not removed, due to the possible existence of text emoticons. As for the stop words, experiments were run both with and without removing them, to assess their impact on the model performance.

In terms of analysis at the aspect level, a number of preprocessing steps were followed. Punctuation, stop words and URLs were also removed for this task, as they represented elements that either could not represent aspect terms or could not contribute to the definition of aspect categories. Additionally, lemmatization of the tokens was performed. Part-of-speech tagging was the last step in our preprocessing process, the result of which was only used at the clustering stage to identify the nouns in a given set of reviews.

4.3.2. TF-IDF Representation

Term frequency–inverse document frequency is a commonly used algorithm that transforms text into numeric representations (embeddings) to be used with machine learning algorithms. As its name suggests, this method combines two concepts: term frequency (TF)—the number of times a term w (word) appears in a document doc—and document frequency (DF)—the number of documents in which a term appears. For the SLSA case, we consider each sentence to be a document and, thus, compute the frequency with which a specific term appears in a sentence and the number of sentences that contain that specific term.

Term frequency can be simply defined as the number of times the term appears in a document, while inverse document frequency (IDF) works by computing the commonness of the term among the documents contained in the corpus.

By using the inverse document frequency, infrequent terms have a higher impact, leading to the conclusion that the importance of a term is inversely proportional to its corpus frequency. While the TF part of the TF-IDF algorithm contains information about a term's frequency, the IDF results in information about the rarity of a specific term.

4.3.3. LSI Representation

In addition to the TF-IDF representation described in the previous subsection, we also propose the examination of the relevance of features extracted by latent semantic indexing (LSI) [51] in a sentiment classification task for the Romanian language.

LSI is a count-based model for representing variable-length texts (in our case, documents and sentences containing reviews written in Romanian) as fixed-length numeric vectors. It builds a matrix of occurrences of words in documents and then uses singular-value decomposition to reduce the number of words while keeping the similarity structure between documents.

Therefore, each document doc is represented as a vector composed of numerical values corresponding to a set $\mathcal{F} = \{ft_1, ft_2, \ldots, ft_{size}\}$ of $size$ features extracted from the review text directly using LSI.

- $doc^{LSI} = (doc_1^{LSI}, \cdots, doc_{size}^{LSI})$, where doc_i^{LSI} ($\forall 1 \leq i \leq size$) denotes the value of the i-th feature computed for the document doc in the documents dataset by using the LSI-based embedding.

As far as the experimental setup is concerned, for extracting the LSI-based embeddings for the documents, we used the implementation offered by Gensim [52]. We opted for $size = 30$ as the length of the embedding and for $num\_topics = 30$ as the number of latent dimensions that represents the number of topics in the given corpus. For the SLSA task, the *size* was reduced to 10, since most of the sentences contain less than 30 terms even before reduction.

4.3.4. Deep-Learned Representation

An alternative to count-based feature extraction for machine learning approaches is represented by using neural models that can automatically generate features for the considered tasks.

In deep learning approaches, specific word-embedding techniques have been developed, which are actually based on neural network layers and dense vectors [30]. In our experiments, we used dense embedding in conjunction with four deep learning networks: CNN, global average pooling (GAP), GRU, and LSTM.

As far as the experimental setup is concerned, after following the general preprocessing step described in Section 4.3.1, we used word number encoding, considering a vocabulary of 15,000 words, and a padding for each review to 500 words These encoding parameters were chosen after performing a search of best parameters based on the characteristics of our dataset and literature findings. The embedding is performed in the first dense embedding layer of each machine learning model. The text document is encoded using a word-embedding dense layer, which is then processed by the network layers. Formally, given doc^{EM}, a text document embedded with a model of token sequences (in which a token could be a word or a letter), with N terms in the document, we have $doc^{EM} = x_1 x_2 \dots x_N$, where $x_i = (x_i^1, x_i^2, \dots, x_i^M) \in \mathbb{R}^M$ is a token embedding of size M. Next, the embedding is submitted to linear transformations (for the CNN model), average region functions (in the GAP model), or memory units and gates (in recurrent neural networks, such as LSTM and GRU).

4.3.5. Word Representations

As far as word representations are concerned, word2vec [30] embeddings are used, a type of representation learned through a neural network from a text corpus. The word2vec model, $f_{w2v} : \mathcal{V} \longrightarrow \mathbb{R}^{mw}$, is an embedding model that maps each word $w \in \mathcal{V}$ to a vector representation (embedding) that has size mw: $embedding_w = (em_1, em_2, \dots, em_{mw})$, where em_i denotes the value of the i-th feature computed for the word w by the model f_{w2v}.

For the proposed tasks, the word2vec model was trained on the corpus of all reviews, with a number of preprocessing steps employed, as described in Section 4.3.1. Next, word embeddings for all lemmas in the vocabulary were learned. We experimentally determined the size of 150 for the word vectors to be the best performing.

4.4. Models

4.4.1. Supervised Classification

To assess the relevance of the TF-IDF and LSI-based embeddings when it comes to the automatic polarity classification for reviews written in Romanian, we trained and evaluated multiple standard machine learning classification models, such as SVM, RF, LR, NB, voted perceptron (VP), and multilayer perceptron (MLP).

The models used in deep learning approaches were configured using a dense embedding base layer, which assumes 500 as the embedding dimension, on top of which the particular model layers are added. The CNN model has a convolution 1D layer, a global max pooling 1D layer, and a hidden dense layer with output 24. For GRU, the hidden layers consist of a bidirectional GRU layer and a dense layer with 24 output units, while LSTM

contains a bidirectional LSTM layer and a dense layer with 24 output units. The GAP model contains an average pooling layer and a dense one with 24 output units. The output dense layer (which is the same for all models) has one unit in the case of binary classification and five output units for multi-class classification. The activation function [53] for the hidden dense layer is the rectified linear unit (ReLU). For binary classification, the output dense layer is the *sigmoid* function, and the models are compiled using binary cross-entropy as the loss function and the adaptive movement estimation optimizer Adam [54]. In the case of multi-class classification with 3 or 5 classes, the models are compiled using the sparse categorical cross-entropy function, and for the output dense layer, we use the $softmax$ function.

Each training session of a model was performed for at most 30 epochs, with early stopping after five epochs without any improvement on the loss function. The implementation was performed using the scikit-learn version 1.3.1 (https://scikit-learn.org/stable/ (accessed on 20 January 2024)) and keras version 2.14.0 (https://keras.io/ (accessed on 20 January 2024)) Python packages.

4.4.2. Unsupervised Analysis

As a clustering technique, we employ k-means and SOM. For similar tasks, k-means is the most frequently encountered [46,55], but SOM has shown better performance in recent studies [56]. Therefore, we aimed to examine the suitability of the two techniques in terms of a Romanian-language dataset. For both, the initial number of nodes/clusters was set at 200, value which was experimentally determined to generate the best results for our dataset. For k-means, the implementation from the scikit-learn library version 1.3.1 was used, with no additional parameters. For SOM, we used the implementation from the NeuPy version 0.6.5 (http://neupy.com/pages/home.html (accessed on 20 January 2024)) Python package, with the learning radius set at 1 and the step at 0.25. The distance used was cosine. The top 5% percent of the obtained aspect clusters are considered representative ($t = 0.05$).

4.5. Evaluation

4.5.1. Methodology

In order to reliably evaluate the performance of the proposed approaches, we performed 10 repetitions of 5-fold cross-validation in all the experiments carried out on our dataset, *RoProductReviews*.

During the cross-validation process, the confusion matrix for the classification task was computed for each testing subset. Based on the values from the confusion matrix, multiple performance metrics, as described in Section 4.5.2, were computed. For each metric, the values were averaged during the cross-validation process, and the 95% confidence interval (CI) of the mean values was calculated.

4.5.2. Performance Indicators

Supervised classification. Based on state-of-the-art views, the most used performance metrics in sentiment analysis are accuracy (*Accuracy*), F1-score (*F*1), precision (*Precision*), recall (*Recall*), specificity ($Specificity$), and area under the ROC curve (AUC). These can be calculated individually for every class in the dataset or as an arithmetic or weighted average for the entire model. To compute each metric, we require the resulting confusion matrix, a matrix that, in supervised learning, evaluates the performance of a model comparing the actual class of an entry versus the predicted class. In this sense, for a class k, we denote with TP_k the true positives of class k and with TN_k the true negatives of class k. TP_k is defined as the number of instances from class k correctly classified in class k, and TN_k is defined as the number of instances that are not in class k and have been correctly classified as a different class from k. FP_k denotes the false positives, meaning the number of instances that are not in class k but have been classified as being class k, and FN_k denotes the false negatives, meaning the number of instances that are in fact in class k but have

been incorrectly classified to be a different class from k. In Equation (2), we define the accuracy of a class k, denoted by $Accuracy_k$. We present the definition for precision for a class k, denoted as $Precision_k$, in Equation (3). In Equation (4), the formula for computing the recall for a class k, denoted by $Recall_k$, is presented. The specificity for a class k, denoted as $Specificity_k$, is computed as in Equation (5).

$$Accuracy_k = \frac{TP_k + TN_k}{TP_K + FP_k + TN_K + FN_k} \tag{2}$$

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \tag{3}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \tag{4}$$

$$Specificity_k = \frac{TN_k}{TN_k + FP_k} \tag{5}$$

The area under the ROC curve is generally employed for classification approaches that yield a single value, which is then converted into a class label using a threshold. For each threshold value, the point $(1 - Specificity, Recall)$ is represented on a plotm and the AUC value is computed as the area under this curve. For the approaches where the direct output of the classifier is the class label, there is only one such point, which is linked to the (0, 0) and (1, 1) points. The AUC measure represents the area under the trapezoid and is computed as in Equation (6).

$$AUC_k = \frac{Recall_k + Specificity_k}{2} \tag{6}$$

The last measure used, the F1-score for a class k, is defined in Equation (7).

$$F1_k = \frac{2 \times Precision_k \times Recall_k}{Precision_k + Recall_k} \tag{7}$$

All the previously mentioned performance evaluation measures range from 0 to 1. For better classifiers, larger values are expected.

For a binary classification in sentiment analysis, we have two classes (the positive class and the negative class); thus, we denote the metrics referring to positive predicted values (PPVs) for the precision of the positive class and negative predicted values (NPVs) for the precision of the negative class. In the general case of multi-class classification with NC classes, having calculated the performance indicators per each class with the above formulas, we define the overall weighted average for each performance metric $PI \in \{Accuracy, Precision, Recall, F1\}$ as in Equation (8).

$$PI = \sum_{k=1}^{NC} weight_k * PI_k, \tag{8}$$

where PI_k is the performance indicator for class k, and $weight_k$ is the weight of class k. The weight of a class k is computed as $weight_k = I_k / I_{NC}$, with I_k equal to the number of instances from class k in the dataset and I_{NC} the total number of instances for all classes in the dataset.

Unsupervised analysis. For the proposed unsupervised analysis, we used two evaluation measures, namely normalized pointwise mutual information (NPMI) [57] and a WordNet-based similarity measure. NPMI is the normalized variant of *pointwise mutual information*, a measure commonly used to evaluate association. This normalized variant has the advantage of a range of values with fixed interpretation.

$$NPMI(w_1; w_2) = \frac{\log\left(\frac{p(w_1,w_2)}{p(w_1)\cdot p(w_2)}\right)}{-\log p(w_1, w_2)} \tag{9}$$

In Equation (9), the formula for computing the NPMI for two words is shown, where $p(w_1)$ and $p(w_2)$ represent the probabilities of occurrence of words w_1 and w_2, respectively, and $p(w_1, w_2)$ is the probability of the co-occurrence of the two. For an aspect cluster $\mathcal{A}_w = \{w_1, w_2, \ldots w_{N_a}\}$, containing N_a words denoted by w_i, $1 \leq i \leq N_a$, the NPMI value is computed as an average over the NPMI values obtained for every pair (w_i, w_j), $i < j$.

While it was defined in the context of collocation extraction, the NPMI measure has also been used in topic modeling literature to evaluate topic coherence [46,47], as it was found to reflect human judgment [58].

The NPMI bases the assessment on the co-occurrence of terms, while the proposed WordNet-based measure takes advantage of the hierarchy of noun and noun phrases in WordNet, in which *is-a* (hyponymy/hypernymy) relations, as well as *part-of* associations, are recorded. We are especially interested in the hierarchy determined by the *is-a* relationships between nouns, as we need to evaluate the ability of determined groups of nouns (aspect *terms*) to describe a more general concept (aspect *category*). Thus, we used a measure that describes how closely related two words are in this hierarchical structure of the WordNet lexical database: the Leacock and Chodorow (LCH) similarity [59]. We compute this metric as in Equation (10), using the Romanian WordNet (RoWordNet [60]).

$$LCH(synset_{w_1}, synset_{w_2}) = -\log_2 \frac{sp(synset_{w_1}, synset_{w_2}) + 1}{2 \cdot maxWNDepth} \tag{10}$$

In Equation (10), the Leacock–Chodorow similarity is computed between the first senses of the two terms w_1 and w_2, which are encapsulated in RoWordNet *synsets*. Thus, we denote by $sp(synset_{w_1}, synset_{w_2})$ the shortest path length between the concepts represented by w_1 and w_2 in the WordNet hierarchy, while $maxWNDepth$ represents the maximum taxonomy depth.

The NPMI measure has the advantage of evaluating performance on an unseen test set, providing a realistic measure of the proposed approach. However, we argue that, while NPMI may be an informative measure with respect to the coherence of topics, which are defined as sets of words that co-occur, it is less suitable for measuring the coherence of groups of words meant to be interpreted as aspect terms which define an aspect category. Usually, when discussing an aspect of a product, the number of aspect terms from a given category used in the same sentence, and even review, is limited—in fact, these aspect terms are often used interchangeably. For NPMI, the range of values is $[-1, 1]$, with values of -1 characterizing words that occur separately, but not together, and values of 1 describing words that only occur together. As for LCH, the range of values is $(0, log(2 * maxWNDepth)]$, where the maximum RoWordNet depth in the hypernymy tree is 16. Considering that $sp(synset_{w_1}, synset_{w_2}) = 0$ when w_1 and w_2 have the same sense, a higher value for the LCH measure signifies increased relatedness between the concepts represented by w_1 and w_2.

5. Results

In this section, we present the results of our study, which aims to investigate the efficacy of machine learning techniques in sentiment analysis, specifically applied to a dataset of Romanian reviews. Results are provided for the three textual levels we addressed: document (as detailed in Section 5.1), sentence (outlined in Section 5.2), and aspect level (discussed in Section 5.3).

5.1. Document Level

The first embedding we evaluated in the context of sentiment analysis when using the RoProductReviews dataset is the one based on LSI.

The classifiers employed in evaluating the relevance of the LSI-based embedding for sentiment analysis were SVM, RF, LR, and a neural-network-based model (VP [61] for binary classification and MLP for multi-class classification).

The results obtained when classifying the RoProductReviews reviews on two classes of polarity, positive and negative, when representing the reviews as LSI-based embeddings, are given in Table 4. For each of the four models, we present the mean value and confidence interval calculated for each performance metric used in evaluation, methodology that was described in Section 4.5. We have obtained AUC values up to 0.894 and F1-score values up to 0.893. The best-performing classification model is LR, which is immediately followed by VP, for which AUC and F1-score values of 0.891 were obtained.

The performances obtained in the case of multi-class classification are given in Table 5. The conclusion that has been drawn for binary classification, regarding the relative performance of the classifiers, holds, the best-performing classifier remaining logistic regression. LR obtained a weighted average F1-score value of 0.690, while the second-best classifier is still the artificial neural network model, in particular the MLP that replaced the VP used for binary classification. A weighted average F1-score value of 0.676 was obtained by the MLP classifier.

Table 4. Results obtained for LSI-based binary classification with the RoProductReviews dataset. The highest value for each performance indicator is marked in bold.

Performance Indicator	SVM Mean	95% CI	RF Mean	95% CI	LR Mean	95% CI	VP Mean	95% CI
Accuracy	0.878	0.001	0.880	0.001	**0.893**	0.001	0.891	0.001
Precision PPV	**0.924**	0.001	0.903	0.002	0.911	0.001	0.897	0.002
Precision NPV	0.838	0.001	0.858	0.001	0.876	0.001	**0.885**	0.003
Average precision	0.881	0.001	0.881	0.001	**0.893**	0.001	0.891	0.002
Sensitivity/Recall—TPR	0.830	0.001	0.859	0.001	0.878	0.001	**0.890**	0.003
Specificity—TNR	**0.928**	0.001	0.902	0.002	0.909	0.001	0.892	0.003
AUC	0.879	0.001	0.881	0.001	**0.894**	0.001	0.891	0.001
F1-score Positive Class	0.875	0.001	0.881	0.001	**0.894**	0.001	0.893	0.001
F1-score Negative Class	0.881	0.001	0.880	0.001	**0.892**	0.001	0.889	0.001
Weighted F1-score	0.878	0.001	0.880	0.001	**0.893**	0.001	0.891	0.001

Table 5. Results obtained for LSI-based multi-class classification with the RoProductReviews dataset. The highest value for each performance indicator is marked in bold.

Performance Indicator		SVM Mean	95% CI	RF Mean	95% CI	LR Mean	95% CI	MLP Mean	95% CI
Accuracy	Avg	0.660	0.001	0.660	0.005	**0.689**	0.001	0.675	0.002
Precision	Class 1 Star	0.599	0.001	0.644	0.002	0.656	0.002	**0.660**	0.014
	Class 2 Stars	0.568	0.002	0.579	0.004	0.602	0.003	**0.604**	0.016
	Class 3 Stars	**0.733**	0.003	0.676	0.005	0.727	0.003	0.709	0.023
	Class 4 Stars	0.617	0.004	0.622	0.004	**0.650**	0.002	0.628	0.011
	Class 5 Stars	**0.858**	0.001	0.816	0.004	0.825	0.002	0.788	0.008
Recall	Class 1 Star	0.712	0.002	0.698	0.003	**0.714**	0.002	0.708	0.016
	Class 2 Stars	**0.607**	0.003	0.588	0.005	0.606	0.004	0.593	0.019
	Class 3 Stars	0.669	0.002	0.667	0.003	**0.698**	0.002	0.679	0.017
	Class 4 Stars	0.680	0.003	0.671	0.003	**0.696**	0.003	0.657	0.015
	Class 5 Stars	0.626	0.002	0.681	0.003	0.720	0.002	**0.728**	0.007
F1 Score	Class 1 Star	0.650	0.001	0.670	0.001	**0.684**	0.001	0.682	0.005
	Class 2 Stars	0.587	0.002	0.583	0.004	**0.604**	0.003	0.597	0.005
	Class 3 Stars	0.700	0.001	0.671	0.004	**0.712**	0.002	0.693	0.004
	Class 4 Stars	0.647	0.003	0.646	0.003	**0.672**	0.003	0.642	0.004
	Class 5 Stars	0.724	0.001	0.743	0.003	**0.769**	0.002	0.757	0.004
Precision	Weighted Avg	0.677	0.001	0.670	0.001	**0.694**	0.002	0.680	0.003
Recall	Weighted Avg	0.660	0.001	0.663	0.001	**0.689**	0.002	0.675	0.002
F1-Score	Weighted Avg	0.663	0.001	0.665	0.001	**0.690**	0.002	0.676	0.002

Table 6 shows the results obtained for binary classification on the RoProductReviews dataset using the deep learning models, while in Table 7, results for multi-classification with five classes are presented.

Table 6. Binary classification using deep learning models with the RoProductReviews dataset. The highest value for each performance indicator is marked in bold.

Performance Indicator	LSTM Mean	95% CI	GRU Mean	95% CI	CNN Mean	95% CI	GAP Mean	95% CI
Accuracy	0.918	0.007	0.920	0.006	**0.930**	0.005	0.918	0.005
Precision PPV	0.924	0.014	0.925	0.012	**0.930**	0.007	0.929	0.010
Precision NPV	0.912	0.010	0.915	0.010	**0.931**	0.006	0.908	0.011
Average precision	0.918	0.012	0.920	0.011	**0.931**	0.006	0.918	0.011
Sensitivity/Recall—TPR	0.915	0.011	0.919	0.011	**0.934**	0.006	0.910	0.013
Specificity—TNR	0.920	0.017	0.921	0.014	**0.926**	0.008	**0.926**	0.012
AUC	0.918	0.007	0.920	0.006	**0.930**	0.005	0.918	0.005
AUPRC	0.920	0.006	0.922	0.006	**0.932**	0.004	0.920	0.005
F1-score Positive Class	0.919	0.006	0.921	0.006	**0.932**	0.004	0.919	0.005
F1-score Negative Class	0.916	0.007	0.918	0.007	**0.928**	0.005	0.917	0.005
Average F1-score	0.918	0.007	0.920	0.006	**0.930**	0.005	0.918	0.005
Weighted F1-score	0.918	0.007	0.920	0.006	**0.930**	0.005	0.918	0.005

Table 7. Multi-class classification using deep learning models with the RoProductReviews dataset. The highest value for each performance indicator is marked in bold.

Performance Indicator		GAP Mean	95% CI	LSTM Mean	95% CI	GRU Mean	95% CI	CNN Mean	95% CI
Accuracy	Avg	0.652	0.013	0.722	0.011	0.739	0.009	**0.767**	0.005
Precision	Class 1 Star	0.699	0.036	0.738	0.030	0.732	0.025	**0.800**	0.026
	Class 2 Stars	0.527	0.039	0.626	0.027	0.645	0.026	**0.692**	0.023
	Class 3 Stars	0.624	0.039	0.698	0.033	0.726	0.022	**0.738**	0.021
	Class 4 Stars	0.637	0.026	0.727	0.024	**0.751**	0.022	0.750	0.019
	Class 5 Stars	0.805	0.040	0.833	0.022	0.846	0.022	**0.854**	0.014
Recall	Class 1 Star	0.714	0.040	0.713	0.042	0.722	0.031	**0.796**	0.023
	Class 2 Stars	0.557	0.064	0.646	0.039	0.665	0.030	**0.691**	0.029
	Class 3 Stars	0.607	0.041	0.712	0.028	**0.726**	0.020	0.718	0.023
	Class 4 Stars	0.605	0.047	0.713	0.031	0.750	0.023	**0.767**	0.019
	Class 5 Stars	0.758	0.043	0.815	0.026	0.825	0.023	**0.848**	0.017
F1 Score	Class 1 Star	0.702	0.018	0.721	0.021	0.725	0.017	**0.796**	0.009
	Class 2 Stars	0.533	0.032	0.633	0.022	0.653	0.021	**0.690**	0.012
	Class 3 Stars	0.610	0.021	0.702	0.012	0.725	0.010	**0.726**	0.008
	Class 4 Stars	0.617	0.025	0.718	0.015	0.750	0.014	**0.757**	0.009
	Class 5 Stars	0.776	0.016	0.822	0.012	0.834	0.014	**0.851**	0.009
Precision	Weighted Avg	0.663	0.014	0.727	0.010	0.743	0.009	**0.769**	0.006
Recall	Weighted Avg	0.652	0.013	0.722	0.011	0.739	0.009	**0.767**	0.005
F1-Score	Weighted Avg	0.652	0.014	0.722	0.011	0.740	0.009	**0.767**	0.005

The best results in the case of binary classification are obtained by the CNN model, with accuracy 0.930, average precision 0.931, recall 0.934, and F1-score 0.930. The other three models have a similar performance of accuracy 0.918 for LSTM, 0.920 for GRU, and 0.918 for GAP. We generally notice a slightly higher precision and F1-score for the positive class than the negative class (for example, GAP precision PPV is 0.929, and LSTM precision NPV is 0.908), which may be due to the slight imbalance of the dataset (2509 negative reviews and 2615 positive reviews), but not very significant, meaning it could also be the result of the random cross-validation experimental setup.

For multi-class classification, the best overall results are also obtained by the CNN model (accuracy 0.767, precision 0.769), followed by GRU (accuracy 0.739, precision 0.743), then LSTM (accuracy 0.722, precision 0.727), and the worst performance is obtained by GAP (accuracy 0.652, precision 0.669).

In terms of performance metric indicators per class, the best result is obtained by all models for Class 5, corresponding to five stars, with the highest value of 0.854 for precision using CNN. The next best value yielded by all models is for Class 1, corresponding to one-star evaluations, with values up to 0.800 for precision using CNN. The worst performance is obtained for class 2-star, for which the highest value is 0.692 for precision with CNN, and the lowest is 0.527 for precision with GAP. This result could be somewhat influenced by the slight imbalance of dataset classes (only 1152 instances of two-star reviews, while there are 1336 reviews with five stars). Moreover, the higher results for the classes with five stars and one star could be explained by the fact that they are the extremes of the rating scale. This means that the sentiment conveyed in the class 5-star and class 1-star reviews is more intense and clearly expressed as *positive* (when the customer is clearly satisfied) or *negative* (expressing customer dissatisfaction).

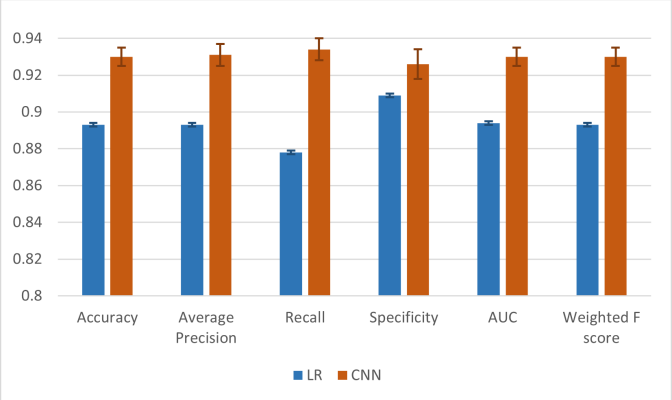
Consequently, the classifiers may also find it easier to identify sentiment patterns in these two rating categories, while for the classes with two, three, and four stars, the reviews may present reasons both in favor of and against the reviewed product, thus a mix of sentiment.

In terms of computation time, the CNN model required the least time for training and repeated cross-fold validation (approximately 8 h), as opposed to the other models, which required between 31 and 48 h on the same hardware device. However, while in this case, CNN proves to be the best choice among deep learning models, an important limitation remained for the execution time, which was much higher than that of classical approaches, for example, those based on LSI embedding and machine learning classifiers such as NB, RF, or SVM.

In the following, we have compared the results obtained by the LR model, which proved to be the best-performing classifier on the RoProductReviews dataset, with those obtained using CNN, which proved to be the best-performing of the deep learning models.

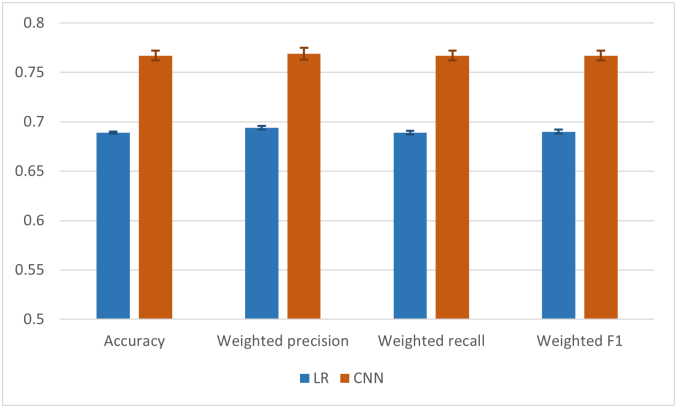
The comparison for binary classification is visually presented in Figure 1a, while Figure 1b depicts the comparison for multi-class classification, when the 95% confidence intervals of the weighted average performance indicators values for the five classes was considered. As Figure 1a,b show, CNN leads to consistent better performance.

We have also comparatively analyzed the results at the class level for both binary and multi-class classification. The comparison at the class level is shown in Figure 2a,b and, as it can be observed, it reinforces the conclusion that CNN behaves consistently better for classifying product reviews written in Romanian in classes of polarity.



(**a**) Binary classification

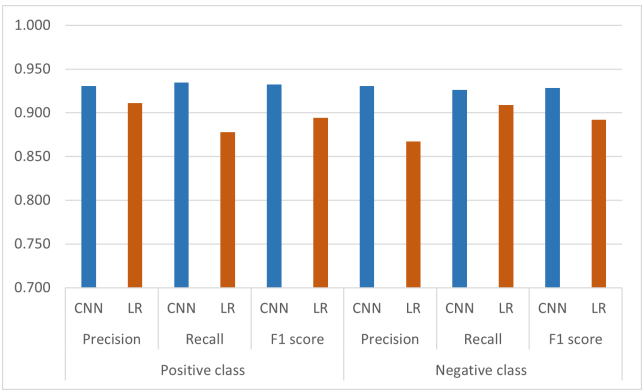
Figure 1. *Cont.*



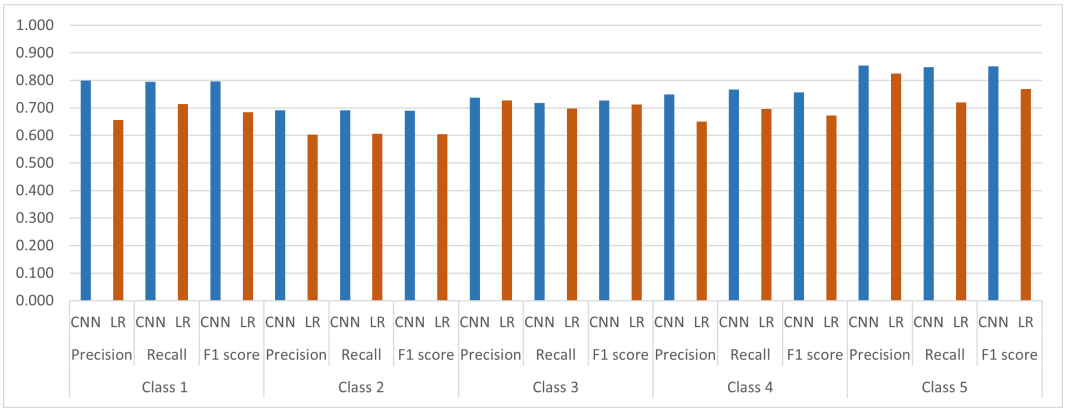
(**b**) Multi-class classification

Figure 1. Comparison between CNN and LR for binary and multi-class classification on the RoProductReviews dataset.

While for the binary classification CNN performs similarly for both positive and negative classes, LR presents small differences in performance for each class and measure. However, for the multi-class classification, there is a consistent behavior of the two models in which class 5 stars presents the best results, while class 2 stars presents the worst result. This shows that specific characteristics of the dataset are most probably responsible for the confusion in classification, namely the smaller number of reviews for two stars (1152) in comparison with the other classes.



(**a**) Binary classification



(**b**) Multi-class classification

Figure 2. Comparison between CNN and LR for binary and multi-class classification on the RoProductReviews dataset at class level.

5.2. Sentence Level

Tables 8 and 9 show the classification results obtained at sentence level for the RoProductReviews dataset, using the TF-IDF and LSI representations presented in Section 4.3, both by removing and not removing stop words, configurations which are denoted as "without", and "with", respectively. Specifically, Table 8 contains the results obtained for the TF-IDF representation and Table 9, the results obtained for the LSI representation. These experiments were performed with three goals in mind: (1) establishing whether the removal of stop words influences the classification results, (2) deciding which representation works best for the sentences from the RoProductReviews dataset, and (3) choosing the algorithm that is best suited for sentence-level sentiment classification.

In order to answer the first question, we have analyzed the results from each table individually, thus leading to a conclusion for each representation. For TF-IDF, almost all the averaged performance indicators (accuracy, precision, recall, and F1-score) are higher for the case when stop words are not removed, with the exceptions of precision for NB. However, if we are to look at the percentage difference, which is 0.4%, we can state that this exception does not impact the overall conclusion; that is, for the TF-IDF representation, the removal of stop words negatively influences the classification results. This means that, although stop words are, by definition, insignificant for determining the sentiment expressed in a sentence, given the fact that sentences, as opposed to documents, contain only brief opinions, the removal of stop words shortens the sentence even more, leading to a decreased classification performance.

The same conclusion holds for the LSI representation as well: all the averaged performance metrics are higher when not removing stop words, for all the classifiers used in the experiments. Thus, we can answer the first question: the removal of stop words does influence the classification results, in a negative manner.

Table 8. Results obtained for TF-IDF-based sentence-level classification with the RoProductReviews dataset. The highest value for each performance indicator is marked in bold.

Performance Indicator		Stopwords	SVM Mean	SVM 95% CI	LR Mean	LR 95% CI	RF Mean	RF 95% CI	NB Mean	NB 95% CI
Accuracy	Avg	Without	0.774	0.012	0.778	0.012	0.751	0.017	0.663	0.015
		With	0.804	0.012	**0.805**	**0.012**	0.771	0.012	0.668	0.013
Precision	Positive	Without	**0.817**	**0.018**	0.806	0.016	0.776	0.025	0.643	0.017
		With	0.815	0.016	0.814	0.017	0.800	0.018	0.662	0.017
	Neutral	Without	0.300	0.287	0.000	0.000	0.265	0.172	0.190	0.082
		With	**0.400**	**0.307**	0.100	0.177	0.174	0.134	0.192	0.076
	Negative	Without	0.722	0.022	0.741	0.025	0.753	0.030	0.782	0.030
		With	0.787	0.021	**0.794**	**0.019**	0.763	0.021	0.745	0.023
Recall	Positive	Without	0.837	0.021	0.853	0.014	0.854	0.026	**0.938**	**0.010**
		With	0.899	0.014	0.907	0.013	0.868	0.016	0.905	0.013
	Neutral	Without	0.008	0.008	0.000	0.000	0.056	0.035	0.045	0.022
		With	0.012	0.009	0.004	0.006	0.039	0.021	**0.064**	**0.026**
	Negative	Without	**0.853**	**0.021**	0.842	0.020	0.759	0.040	0.421	0.026
		With	0.842	0.018	0.836	0.019	0.796	0.019	0.475	0.022
F1-Score	Positive	Without	0.827	0.014	0.828	0.014	0.812	0.014	0.763	0.013
		With	0.855	0.011	**0.858**	**0.012**	0.832	0.011	0.764	0.011
	Neutral	Without	0.017	0.016	0.000	0.000	0.080	0.045	0.070	0.033
		With	0.022	0.017	0.006	0.011	0.061	0.031	**0.093**	**0.036**
	Negative	Without	0.781	0.013	0.788	0.016	0.753	0.017	0.547	0.026
		With	0.813	0.015	**0.814**	**0.014**	0.778	0.013	0.579	0.019
Precision	Weighted Avg	Without	0.741	0.027	0.715	0.016	0.725	0.022	0.661	0.017
		With	0.772	0.026	**0.747**	**0.022**	0.734	0.019	0.657	0.015
Recall	Weighted Avg	Without	0.774	0.012	0.778	0.012	0.751	0.017	0.663	0.015
		With	0.804	0.012	**0.805**	**0.012**	0.771	0.012	0.668	0.013
F1-Score	Weighted Avg	Without	0.743	0.013	0.745	0.014	0.728	0.018	0.621	0.017
		With	0.770	0.014	**0.771**	**0.013**	0.748	0.012	0.637	0.016

Table 9. Results obtained for LSI-based sentence-level classification with the RoProductReviews dataset. The highest value for each performance indicator is marked in bold.

Performance Indicator		Stopwords	SVM Mean	SVM 95% CI	LR Mean	LR 95% CI	RF Mean	RF 95% CI	NB Mean	NB 95% CI
Accuracy	Avg	Without	0.663	0.014	0.660	0.010	0.699	0.012	0.609	0.013
		With	0.724	0.013	0.713	0.012	**0.728**	**0.011**	0.669	0.012
Precision	Positive	Without	**0.781**	**0.018**	0.763	0.020	0.738	0.020	0.683	0.017
		With	0.745	0.018	0.725	0.013	0.749	0.013	0.698	0.014
	Neutral	Without	0.000	0.000	0.000	0.000	**0.229**	**0.116**	0.000	0.000
		With	0.000	0.000	0.000	0.000	0.271	0.171	0.017	0.052
	Negative	Without	0.574	0.019	0.577	0.016	0.662	0.019	0.537	0.020
		With	0.698	0.024	0.694	0.024	**0.708**	**0.022**	0.628	0.025
Recall	Positive	Without	0.639	0.017	0.641	0.019	0.781	0.017	0.632	0.018
		With	0.821	0.019	**0.833**	**0.017**	0.829	0.017	0.780	0.019
	Neutral	Without	0.000	0.000	0.000	0.000	**0.034**	**0.018**	0.000	0.000
		With	0.000	0.000	0.000	0.000	0.024	0.013	0.001	0.004
	Negative	Without	**0.837**	**0.021**	0.827	0.018	0.730	0.025	0.706	0.020
		With	0.749	0.024	0.701	0.020	0.743	0.020	0.659	0.020
F1-Score	Positive	Without	0.702	0.014	0.697	0.014	0.759	0.013	0.656	0.014
		With	0.781	0.012	0.775	0.010	**0.786**	**0.010**	0.736	0.010
	Neutral	Without	0.000	0.000	0.000	0.000	**0.057**	**0.029**	0.000	0.000
		With	0.000	0.000	0.000	0.000	0.045	0.025	0.002	0.006
	Negative	Without	0.681	0.018	0.680	0.014	0.694	0.014	0.610	0.017
		With	0.722	0.016	0.697	0.016	**0.724**	**0.014**	0.642	0.016
Precision	Weighted Avg	Without	0.636	0.016	0.629	0.013	0.667	0.016	0.570	0.014
		With	0.666	0.016	0.653	0.014	**0.695**	**0.018**	0.616	0.015
Recall	Weighted Avg	Without	0.663	0.014	0.660	0.010	0.699	0.012	0.609	0.013
		With	0.724	0.013	0.713	0.012	**0.728**	**0.011**	0.669	0.012
F1-Score	Weighted Avg	Without	0.637	0.015	0.633	0.011	0.675	0.013	0.584	0.013
		With	0.693	0.015	0.681	0.013	**0.701**	**0.013**	0.639	0.012

Once we have established that better results are obtained without removing the stop words, in order to answer the second question, we only compare the results obtained for TF-IDF and LSI representations when keeping the stop words, presented in the same tables (Tables 8 and 9, respectively). For all the averaged performance indicators, all algorithms yield higher values for the TF-IDF representation, with the exception of NB. Yet, the difference in accuracy and weighted recall is 0.1% between the two representations, while the difference in F1-score is 0.2%. Taking all of these into consideration, we can state that the TF-IDF representation is better suited for all the algorithms employed in the experiments. This conclusion can be motivated by the nature of the representations themselves since LSI attempts to reduce the dimensionality of the TF-IDF representation, and sentences can be viewed as very short documents, reducing the dimensionality leads to a loss of relevant information.

Finally, so as to choose the algorithm that is best suited for SLSA on the RoProductReviews dataset, we compare the performance indicators obtained with the TF-IDF representation for SVM, LR, RF, and NB. Figure 3 presents these values, gathered from Tables 8 and 9. The results for each category are not included in Figure 3, because we consider the averaged performance indicators to suffice for the intended comparison; however, the values for these metrics can be found in the respective tables. Therefore, considering these performance indicators, LR obtains the highest values for accuracy, weighted recall, and weighted F1-score, while SVM leads to the highest weighted precision value. Yet, since the difference in weighted precision between the two algorithms is 0.025, we can conclude that LR is the best-suited algorithm for the task of sentiment analysis at the sentence level, which coincides with the conclusion drawn for the document level. Therefore, we can state that sentiment analysis for Romanian can be performed at both the document and sentence levels using the LR algorithm.

If we are to look at the results obtained for each class, as presented in Table 8, one can notice that the results obtained for the neutral class are very low, which is explainable given the unbalanced dataset. In order to solve this problem, a higher number of neutral

sentences, comparable to that of the positive and negative sentences, should be used for training the algorithms.

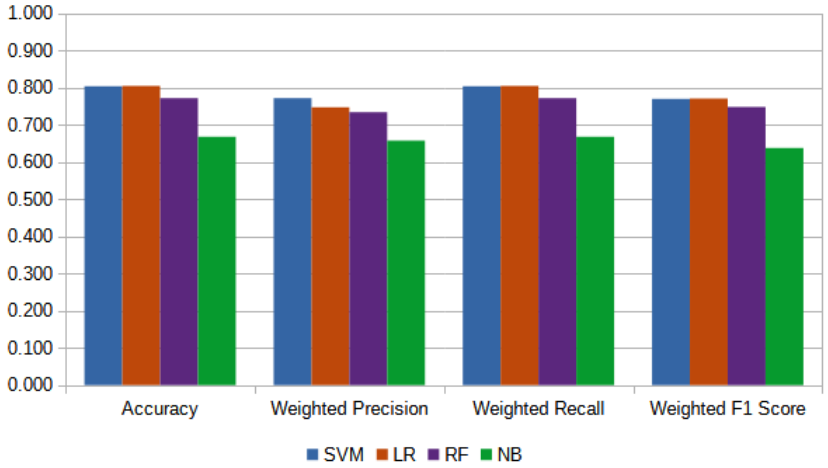


Figure 3. Sentence-level classification with the RoProductReviews dataset.

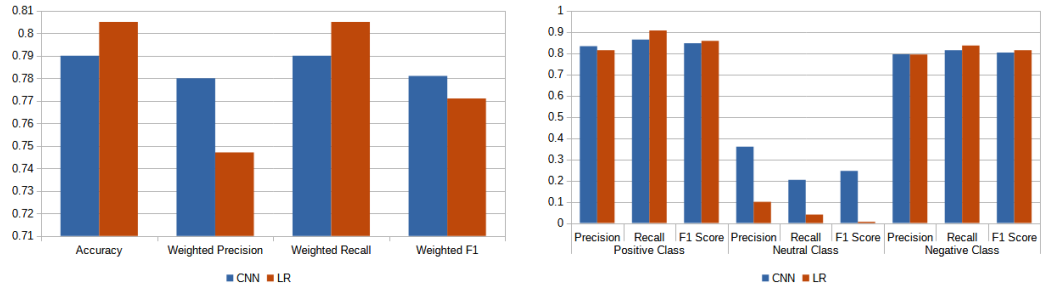
Concluding, from the results presented in Tables 8 and 9, in order to perform the task of SLSA on the RoProductReviews dataset, the LR algorithm should be applied on the TF-IDF representation of the sentences, without removing the stop words, leading to an accuracy score of 0.805, a weighted precision score of 0.747, a weighted recall score of 0.805, and a weighted F1-score of 0.771.

Given that CNN is the most effective model at the document level, embedding encoding and CNN are used in the deep learning technique at the sentence level on the RoProductReviews subset for multi-classification with three classes. The results are presented in Table 10, and are comparable to the other approaches presented previously for SLSA. The accuracy obtained is 0.790, while weighted precision is 0.780, weighted recall 0.790 and weighted F1-score 0.781. This means that, in comparison with the results obtained for LR, the deep learning approach leads to better precision and F1-score, while the classical ML algorithm obtains higher accuracy and recall values. Figure 4a presents this comparison, for an easier analysis. Since the differences are very small—0.015 in accuracy and 0.015 in weighted recall, in favor of LR, and 0.033 in weighted precision and 0.01 in weighted F1 in favor of CNN—a clear conclusion cannot be drawn: each of these two algorithms can be used to perform the task of SLSA.

Table 10. Multi-class classification using deep learning models at sentence level with three classes: negative, neutral, and positive. The highest value for each performance indicator is marked in bold.

Performance Indicators		**CNN Mean**	**95% CI**
Accuracy	Avg	**0.790**	0.011
Precision	Negative	0.795	0.025
	Neutral	0.360	0.078
	Positive	0.833	0.017
Recall	Negative	0.814	0.020
	Neutral	0.203	0.050
	Positive	**0.864**	0.021
F1-Score	Negative	0.803	0.010
	Neutral	0.246	0.045
	Positive	0.847	0.007
Precision	Weighted Avg	0.780	0.012
Recall	Weighted Avg	**0.790**	0.012
F1-Score	Weighted Avg	0.781	0.010

However, there are very big differences in the performance indicators per class. Given that the dataset is very unbalanced, this limits the deep learning model's learning (1092 instances for the positive class, 171 instances for the neutral class, 804 instances for the negative class). As such, the neutral class performs very poorly (the lowest value is 0.203 for recall), while the positive class performs the best (the highest value is 0.847 for the F1-score). In comparison to LR, as presented in Figure 4b, CNN performs better for the neutral class and obtains better precision for the positive and negative classes, while LR outperforms CNN in terms of recall and F1-score for both the positive and negative classes.



(**a**) (**b**)

Figure 4. Comparison between CNN and LR for sentence-level classification with the RoProductReviews dataset: (**a**) overall and (**b**) with respect to class.

5.3. Aspect Level

5.3.1. Aspect Term Extraction

In general, in aspect-based sentiment analysis, *aspects* are specific to a product type. Users may be interested in the *battery life*, *photo/video quality*, and *performance* of a phone, but paying more attention to *memory* in case of an external hard drive and *coverage* for a wireless router. Naturally, there are common aspects that can be evaluated for multiple product types, which is owed to the overlap in the category taxonomy itself. For instance, *processing speed* can be evaluated on all electronics with a processing unit, as can *sound quality* on devices that support audio input and output. In this paper, we attempt to discover the most important aspects of a product category from our dataset using two clustering approaches.

Table 11 shows the results obtained for each of the two clustering algorithms employed, SOM and k-means, in terms of a mean over the random states and the 95% CI, for each of the product categories in the dataset. In terms of NPMI, in 8 out of 12 cases, the SOM algorithm provides better results, with k-means clusters achieving a higher score for *fitness bracelets*, *headphones*, and *monitor* product categories, though by small margins. For *smartwatches*, the generated clusters obtain the same score for both algorithms. If the clusters are evaluated using the LCH metric, for 7 out of 12 product categories, the SOM algorithm partitions the considered words better.

Overall, the low NPMI scores could be explained by the nature of the word groups. For some aspect categories, some aspect terms might be used interchangeably rather than co-occur in the same review. For instance, in terms of evaluating the *price* of a product, users most often limit themselves to either saying *A lot of money, but it's worth it* or *A good price for what it offers*, but not both in the same review, as the sentences have very similar meaning. Therefore, nouns *money* and *price* may occur less frequently together than in other types of text that discuss a finance topic. Alternatively, when evaluating the *functionalities* of a smartwatch, one could talk, in the same text span, about health monitoring using a number of different words: for instance, *somn/sleep* monitoring, *puls/heart rate* measuring, *tensiune/blood pressure*.

Table 11. Results for aspect term extraction and grouping in terms of NPMI and LCH. An average over the random states is provided along with the value for the 95% CI.

	SOM				K-Means			
	NPMI		LCH		NPMI		LCH	
Product Category	**Mean**	**95% CI**	**Mean**	**95% CI**	**Mean**	**95% CI**	**Mean**	**95% CI**
Fitness bracelets	−0.698	0.017	1.412	0.031	−0.697	0.028	1.390	0.023
Headphones	−0.627	0.025	1.484	0.036	−0.603	0.023	1.427	0.029
Keyboard	−0.660	0.024	1.421	0.022	−0.666	0.023	1.432	0.024
Laptop	−0.648	0.013	1.384	0.030	−0.685	0.017	1.385	0.025
Monitor	−0.596	0.025	1.364	0.016	−0.587	0.022	1.386	0.022
Mouse	−0.636	0.011	1.626	0.043	−0.654	0.019	1.526	0.026
Router	−0.616	0.027	1.609	0.027	−0.736	0.019	1.497	0.032
Smartphone	−0.700	0.016	1.406	0.034	−0.732	0.025	1.351	0.021
Smartwatch	−0.687	0.016	1.352	0.024	−0.687	0.012	1.395	0.024
Speakers	−0.632	0.019	1.487	0.016	−0.649	0.013	1.452	0.025
Tablet	−0.735	0.013	1.352	0.024	−0.753	0.017	1.347	0.031
Vacuum cleaner	−0.604	0.011	1.421	0.022	−0.642	0.016	1.448	0.019

The LCH score, on the other hand, might deal well with the first case, identifying *price* and *money* as similar concepts, but lacks the ability to contextually assess the relatedness of groups like the second example (e.g., *sleep*, *heart rate*, *blood pressure*).

In the following, we present more detailed results for a selected product category, *laptop*. Appendix A includes results for another product category, *monitors*, to showcase the ability of the approach to identify relevant aspect categories for different product types.

The noun groups $\mathcal{A}_{w_l}$ obtained in one example run for the category *laptop* are presented in Table 12, which shows the eight aspect clusters that were obtained. As it can be seen, these noun clusters are relatively easy to interpret. For the first cluster, $\mathcal{A}_{w_{l_1}}$, the label of *durability/reliability* was assigned, as the words within represent either words that refer to time (*perioadă/period*, *timp/time*, *an/year*, *lună/month*) or to the use of the product (*utilizare, folosire/usage*), with potential issues (*pană/breakdown*, *problemă/problem*). Temporal words are also used to form $\mathcal{A}_{w_{l_2}}$, but, in this case, it is more likely that the *battery life* of the laptop is discussed, since the referenced periods of time are shorter: *saptaman*, *saptamană/week*, *oră/hour*. This differentiation between temporal words used in *battery life* and *durability* aspect clusters indicates that using word2vec embeddings trained on the review corpus allows the clustering process to capture associations that go beyond classic semantic categories (e.g., grouping together words that refer to time). The ability of the learned representations to encode information from the specific usage patterns from the corpus they are trained on aids the formation of meaningful groups in terms of their ease of interpretability as aspect categories.

Table 12. Example clusters obtained using SOM for product type *laptop*.

	Terms	**Assigned Label**	**NPMI**	**LCH**
$\mathcal{A}_{w_{l_1}}$	**perioadă, timp, pană, problemă, utilizare, inceput, an, lună, folosire** *period, time, breakdown, problem, usage, start, year, month*	Durability	−0.481	1.805
$\mathcal{A}_{w_{l_2}}$	**baterie, saptamană, saptaman, figură, oră** *battery, week, issue, hour*	Battery life	−0.445	1.595
$\mathcal{A}_{w_{l_3}}$	**așteptare, stea, ron, pret, ban, raport, leu** *expectation, star, Romanian leu (RON), price, cent/money, ratio*	Price	−0.516	0.890
$\mathcal{A}_{w_{l_4}}$	**mufă, wireless, pachet, adaptor, laptop, receiver, cutie, usb** *socket, wireless, package, adapter, laptop, receiver, box, USB*	Connectivity	−0.543	1.708
$\mathcal{A}_{w_{l_5}}$	**medie, design, slab, calitate, pro, ok, rest, aspect, dorit, material** *average, design, poor, quality, pro, ok, otherwise, wanted, material*	Build quality / Design	−0.564	1.608
$\mathcal{A}_{w_{l_6}}$	**foto, imagine, rezoluție, hd, display, ecran, caracteristică** *photo, image, resolution, HD (High Definition), display, screen, characteristic*	Display	−0.635	1.385
$\mathcal{A}_{w_{l_7}}$	**calculator, win, sită, desktop, stick, windows, ubunt** *computer, win, site, desktop, stick, Windows, Ubuntu*	Operating system	−0.407	1.779
$\mathcal{A}_{w_8}$	**modul, proces, driver, instalar, bios, drive, boot, parolă** *module, process, driver, installation, BIOS, drive, boot, password*	Software components	−0.632	1.618

Cluster $\mathcal{A}_{w_{l_3}}$ can be interpreted as referring to *price* or the value for money, while $\mathcal{A}_{w_{l_4}}$ can be assigned a label of *connectivity* based on words such as *mufă/socket*, *adaptor/adapter*, *receiver*, *wireless*, *USB*. $\mathcal{A}_{w_{l_6}}$ is equally easy to interpret, as it contains nouns that almost exclusively refer to the *display* aspect category. As far as aspect clusters $\mathcal{A}_{w_{l_7}}$ and $\mathcal{A}_{w_{l_8}}$ are concerned, we highlight the distinction between the *operating system* and *software components* aspects, both of which can provide insights into the laptop's hardware, software and performance. However, the first terms (i.e., terms comprising $\mathcal{A}_{w_{l_7}}$) are relevant when discussing the laptop's compatibility with various software, operating systems, and its ability to access websites and web content effectively, while terms in $\mathcal{A}_{w_{l_8}}$ lean towards descriptions of internal components and the system configuration, often discussed in laptop reviews to evaluate its performance, ability to upgrade, and security features.

A somewhat less obvious cluster is $\mathcal{A}_{w_{l_5}}$. The terms included in this group are frequently used to either express an evaluation with regard to the *quality* of a product (terms *calitate/quality*, *pro*), or address some general aspects (e.g., *"În rest, n-au fost probleme"/"Otherwise, there were no issues"*, *"În rest, e ok"/"Otherwise, it's fine"*).

This, in turn, highlights one of the limitations of the proposed approach, namely the use of automatic PoS tagging, which may, at times, erroneously identify words as nouns, either because of homonymy (for instance, "bun" can be both an adjective, meaning *good*, or a noun, meaning *asset*) or the use of more informal constructions such as *super ok*, *super tare* (*super nice*), *super mulțumit* (*super content*), which the tagger may have difficulties in correctly processing.

5.3.2. Aspect Category Detection

In this subsection, we present results for the *aspect category detection* task, using the aspect clusters presented in Section 5.3.1 for the product type *laptop* to identify their presence in a review.

Table 13 provides a series of example reviews from the product category *laptop*, chosen to reflect the diversity of expression in the corpus, both in terms of the length of reviews, and in terms of the explicit and implicit discussion of aspects.

As it can be seen, our approach manages to identify both implicitly and explicitly referred aspects. This is owed to the use of word embeddings that capture subtle semantic similarities. For instance, if assessing the results obtained for reviews R_{l_6} or R_{l_7}, we observe that in R_{l_6}, only *operating systems* are referred to explicitly, while the updating issues point somewhat indirectly to the aspect *software components*. For R_{l_7}, it is interesting to see the distribution of the aspects, with *battery life*, *durability/reliability*, and *build quality/design* identified to cover, in large part, the target of the opinion expressed in the short review. While the use of a temporal quantifier (*3 zile/3 days*) makes the presence of the *durability/reliability* aspect expected, the presence of *battery life* is less so. A laptop not turning on may indeed involve an issue with the battery, which is knowledge the word2vec model likely learned by seeing the verb *a aprinde/turn on* in contexts which also involved discussions about the battery performance.

For an in-depth evaluation of the proposed approach's performance with respect to the length of reviews, we examine specific instances, namely reviews R_{l_2}, R_{l_3}, and R_{l_7}. The succinct information provided in R_{l_2} aligns with categories exhibiting the highest scores: *price* and *build quality/design*. The user's phrase "good for this money" effectively alludes to the laptop's value for money and overall quality. Longer reviews are addressed with equal proficiency, and increased references to discussed aspects may even contribute to a clearer distinction between aspect categories. For example, R_{l_3} is exclusively assigned to the *operating system* and *software components* categories. In contrast, R_{l_7}, which contains a profoundly implicit reference to the product's *durability* and *battery life*, is attributed to every aspect category to varying degrees. These observations lead us to the conclusion that the length of the considered text has a lesser role than the clarity with which aspects are referenced in the precise identification of aspect categories.

Lastly, while the results obtained provide encouraging results, there are cases in which the proposed method encounters difficulties, such as R_{l_9}. The *display* aspect is explicitly mentioned in the review, but it is unclear how *connectivity* and *durability/reliability* are discussed. Moreover, in a review such as R_{l_4}, it can be argued that *battery life* should have a higher score.

Table 13. Aspect category detection results with respect to a set of reviews from product category *laptop*.

	Review Text	Durability/ Reliability	Battery Life	Price	Connectivity	Build Quality/ Design	Display	Operating System	Software Components
R_{l_1}	***Un laptop de buget se poate folosii pentru varsnici sau copii. Pentru banii ceruti este un produs foarte bun.*** *A budget laptop can be used for seniors or children. For the money asked, it's a very good product.*	0.004	0	0.792	0.001	0.203	0	0	0
R_{l_2}	***Bun ptr bani astia*** *Good for this money*	0.015	0.007	0.497	0.003	0.470	0.006	0.002	0
R_{l_3}	***Instalarea Windows-ului la laptopurile HP cu procesoare Intel de generatie 11 sau 12 necesita drivere speciale pentru fiecare model in parte, altfel masina nu vede hardul. Este un bag de fabricatie. Luati-le mai bine direct pe cele cu Windows-ul preinstalat.*** *Installing Windows on HP laptops with 11th or 12th generation Intel processors requires special drivers for each model; otherwise, the system doesn't recognize the hard drive. It's a manufacturing glitch. It's better to get the ones with pre-installed Windows.*	0	0	0	0	0	0	0.384	0.616
R_{l_4}	***Nu încarcă bateria. Nu recomand decât dacă va doriți un laptop fix, gen PC*** *It doesn't charge the battery. I only recommend it if you want a desktop-like laptop.*	0.286	0.100	0.011	0.151	0.412	0.030	0.004	0
R_{l_5}	***Fraților, nu vă sfătuiesc să vă zgârciți la câteva sute de lei pentru că acest produs este foarte slab! Îl am de o lună și deja s-a desfăcut toată rama din împrejurul display ului… Foarte slab…*** *Brothers, I advise you not to skimp on a few hundred lei because this product is very weak! I've had it for a month, and the frame around the display has already come apart… Very poor…*	0.090	0.001	0.210	0	0.695	0.003	0	0
R_{l_6}	***Nemulțumit. Îl voi returna cât de curând. Se tot actualizează, ba se blochează. Are Windows-ul 10 instalat. Păcat de firma hp și de HDD de 1T.*** *Unsatisfied. I will return it as soon as possible. It keeps updating, and it even freezes. It has Windows 10 installed. It's a shame for the HP brand and the 1TB HDD.*	0.002	0.001	0.007	0.007	0.002	0.014	0.502	0.466
R_{l_7}	***Dupa a 3 zi nu s-a mai aprins.*** *After 3 days, it didn't turn on anymore*	0.220	0.198	0.046	0.080	0.220	0.089	0.067	0.081
R_{l_8}	**L. Am luat pentru gaming și deși are rtx 3050 ti in jocuri cu ray tracing nu depășește 25–30 cadre pe full hd, 2k/4k nu mai discutam..** *I got it for gaming, and even though it has an RTX 3050 Ti, in games with ray tracing, it doesn't go beyond 25–30 frames per second at full HD. Let's not even discuss 2K/4K.*	0.001	0	0.003	0	0.003	0.991	0	0.002
R_{l_9}	***Laptopul este performant dar display-ul are probleme…*** *The laptop is performant, but the display has issues…*	0.438	0.026	0.019	0.119	0.039	0.243	0.070	0.046

6. Discussion

In this section, we present the results of a comparison between our approaches for document-level sentiment analysis and two existing approaches from the literature, as well as an overall analysis of the obtained results in order to provide insights into the research questions formulated in the Introduction.

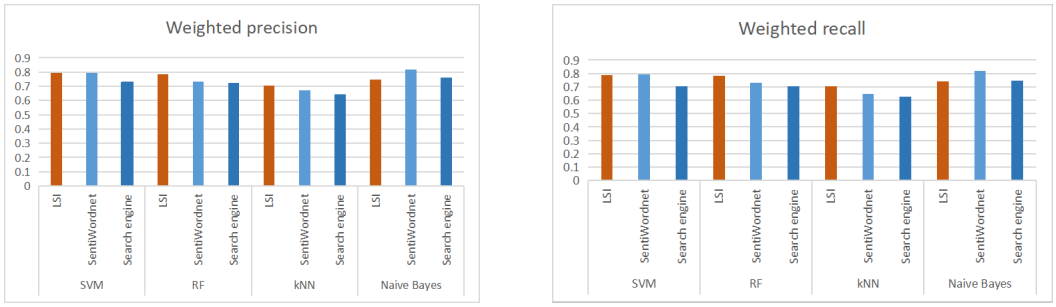
6.1. Comparison to Related Work

In this study, we have also compared our approaches for document-level sentiment analysis with two existing approaches: one based on SentiWordnet and one based on searches using a search engine, proposed by Russu et al. in [28]. In addition, we have also evaluated our approaches on the movie reviews dataset Russu et al. have employed in their paper.

For a fair comparison, focused on the document representations, we have employed the same classifiers as in [28], namely RF, kNN, NB, and SVM, the same implementation for them (as offered by Weka) and the same evaluation methodology, that is, 10-fold cross-validation. We repeated 10-fold cross-validation ten times and report 95% confidence intervals for the performance measures.

The only two performance measures the authors report values for are weighted precision and weighted recall, so we have computed the same performance indicators for the LSI-based approach.

The experimental results are numerically presented in Table 14 and visually represented in Figure 5a,b. In Table 14, the best performances are highlighted.



(**a**) (**b**)

Figure 5. Comparison to related work: LSI-based versus SentiWordnet-based [28] and search-engine-based [28] document polarity binary classification with respect to two performance indicators: (**a**) weighted precision and (**b**) weighted recall.

It can be observed that, when using naïve Bayes as a classifier, the LSI-based approach is outperformed by the approaches proposed by Russu et al. [28], but when using support vector machines (SVM), the LSI-based approach outperforms the search-engine-based approach [28], while it is slightly outperformed by the SentiWordnet-based approach. However, when using both random forest and k-nearest neighbors as automatic classification algorithms, the LSI-based approach we propose outperforms both the SentiWordnet-based and the search-engine-based approaches proposed by Russu et al. [28].

When averaging the values for the performance indicators over the different classifiers employed, the LSI-based approach leads to an overall weighted precision of 0.757, compared to 0.755 for the SentiWordnet-based approach and 0.715 for the search-engine-based approach, and an overall weighted recall of 0.755, compared to 0.748 for the SentiWordnet-based approach and 0.694 for the search-engine-based approach. So, both performance indicators confirm that the performance of using LSI-based embeddings for representing review documents written in Romanian as a basis for automatic sentiment polarity classification leads to an overall slightly superior performance when compared to the SentiWordnet-based and search-engine-based approaches proposed by Russu et al. [28] for the considered movie reviews dataset.

As for the deep learning approach, dense embedding was integrated into the best-performing model up to this point, so CNN was used for classification, and the same evaluation methodology as in [28] was employed (namely, 10-fold cross-validation). The last line in Table 14 presents the results obtained for the two performance indicators utilized. While we notice the weighted precision 0.756 is comparable with the other approaches, the CNN model performs better than all search-engine-based approaches and kNN approaches, but it is outperformed by LSI-based representation used with SVM or RF. As for recall, this performance indicator is the weakest of all with a value of 0.534. In this case, the weaker performance of the deep learning approach in some cases could be explained by the small number of training instances in the dataset ($n = 1000$ instances), which limits the deep learning model's capacity to learn. For the previous experiments, there were 2067 instances for SLSA and 6434 instances for DLSA.

Table 14. Comparison to related work: LSI-based versus SentiWordnet-based [28] and search-engine-based [28] document polarity binary classification. The highest value for each performance indicator is marked in bold.

Classifier	Approach	Weighted Precision	Weighted Recall
SVM	LSI-based	0.794 ± 0.003	0.789 ± 0.002
	SentiWordnet-based [28]	**0.795**	**0.795**
	Search engine-based [28]	0.729	0.705
RF	LSI-based	**0.784 ± 0.005**	**0.783 ± 0.004**
	SentiWordnet-based [28]	0.735	0.732
	Search engine-based [28]	0.723	0.703
kNN	LSI-based	**0.704 ± 0.005**	**0.704 ± 0.005**
	SentiWordnet-based [28]	0.671	0.646
	Search engine-based [28]	0.645	0.625
NB	LSI-based	0.746 ± 0.005	0.743 ± 0.005
	SentiWordnet-based [28]	**0.818**	**0.818**
	Search engine-based [28]	0.763	0.744
CNN	Dense Embedding	0.756	0.534

6.2. Analysis

In this section, we provide an analysis of the results obtained in our study, considering the research questions we have started from.

RQ1: Is latent semantic indexing (LSI) in conjunction with conventional machine learning classifiers suitable for sentiment analysis of documents written in Romanian?

The results obtained in the sentiment analysis task at the document level for the RoProductReviews dataset are presented in Section 5.1. The results provided in Tables 4 and 5 and the analysis of the performance of standard machine learning classifiers used in conjunction with an LSI representation indicate an affirmative answer to **RQ1**: using an LSI representation for documents written in Romanian as input for conventional machine learning classifiers leads to good results in our sentiment analysis task. The comparison with two existing approaches (presented in Section 6.1) also reinforces our conclusion.

RQ2: Can deep-learned embedding-based approaches improve the performance of document- and/or sentence-level sentiment analysis, as opposed to classical natural language processing (NLP) embedding-based non-deep-learning approaches?

In our study, we have experimented with deep learning approaches for sentiment analysis at both the document and sentence levels, in a binary and a multi-class setting. The obtained results at the document level are presented in Tables 6 and 7 and those obtained at sentence level are shown in Table 10. We have also compared them with the best results obtained using ML classifiers. The obtained results clearly show that deep learning approaches can improve performance compared to a classical ML classifier at the document level. For shorter texts, the improvement is less clear. Our experiments also point out the drawbacks of deep learning approaches, namely the higher cost in terms of resources such as running time and the need for a large dataset for training.

RQ3: What is the relevance of different textual representations in the task of sentence polarity classification, and what impact do additional preprocessing steps have in this task?

In our study, we have also examined different textual representations for sentence-level sentiment analysis to determine if the representation used affects the obtained results. In Tables 8 and 9, we have shown the results obtained by using conventional machine learning classifiers in conjunction with two representations (TF-IDF and LSI). From these results, we conclude that, while LSI is suitable for document-level analysis, its dimensionality reduction component is not improving the sentence polarity classification; on the contrary, thus, the TF-IDF method alone suffices for this granularity.

Regarding the impact of the preprocessing step, our results have shown that the additional step of stop words removal negatively influences the classification results. We consider that this may be due to the smaller dimension of the sentences, compared to the dimension of documents.

RQ4: In terms of aspect extraction, is it feasible for a clustering methodology relying on learned word embeddings to delineate groups of words capable of serving as aspect categories identified within a given corpus of documents?

We have experimented with clustering-based approaches for aspect term extraction and aspect category detection, the results obtained being presented in Section 5.3. The performance of these two approaches for aspect term extraction is presented in Tables 11 and 12, respectively, for a specific product category. From the results obtained, we can conclude that the proposed methodology produces coherent aspect clusters for given product types (namely, *laptop* and *monitor* in our experiments), resulting in interpretable and easy-to-label aspect categories. The approach used has the ability to identify aspect clusters (and, thus, aspect categories) with strong relevance to their respective product types.

RQ5: How can the aspect categories discussed within a document be identified, if an aspect category is given through a set of words?

For the aspect category detection task, we have used in our experiments a simple and completely unsupervised method based on word similarity in an embedding space, results for which are shown in Table 13. From the obtained results, we can conclude that a simple approach, like the one we have used, manages to correctly identify aspect categories in units of texts of varying lengths containing both implicit and explicit mentions of them.

In terms of the full aspect term extraction-aspect category detection pipeline, we have observed that the approach used demonstrates remarkable versatility, as it can be applied in order to analyze aspects of a single product type in depth, or it can be scaled up to handle more extensive categories of products. For instance, it can be effectively employed to explore and categorize aspects within the product type category of *peripherals*, making it a valuable tool for comprehensive product analysis. Moreover, the technique used for identifying aspects that are discussed in a review can be modified to address text units of varying lengths (e.g., sentences, sentence parts), which can then be assigned a sentiment label using the appropriate model.

While the approach holds promise, it is not without its limitations. The quality and effectiveness of the generated aspect clusters are directly influenced by the quality of the preprocessing pipeline. Elements such as part-of-speech tagging and lemmatization play a crucial role in the accuracy and relevance of the results. Additionally, the readability and complexity of the language used in the corpus can impact the quality of the clusters. Another limitation is the manual assignment of category labels, which can introduce subjectivity and potential inaccuracies in the analysis.

6.3. Potential Challenges and Limitations

Data accuracy and accessibility. A first challenge in implementing a sentiment analysis system may refer to the availability and quality of data gathered from online sources. The utility of such a sentiment analysis model is dependent on the representativeness of the training data, which should encompass a comprehensive set of diverse examples that cover the sentiments and language patterns that may be encountered in the target domain or application. Additionally, the dataset should be balanced, providing the model with sufficient information to capture the relevant patterns for each of the target classes. We have addressed these aspects in Section 4.1, specifically Sections 4.1.1 and 4.1.2, which describe our data collection process and its result.

Resistance to machine learning approaches. Another potential challenge in implementing an automated sentiment analysis system is the lack of transparency in the decision-making process of some models, as well as the hesitation to rely on machine predictions, especially for a task like sentiment analysis. Sentiment and emotion are complex concepts, and their interpretation and evaluation are at times difficult even for humans. In terms of

the latter, we highlight the characteristics of the type of data we employ in our experiments. In product reviews, users evaluate a product, describing either their satisfaction or their dissatisfaction with the product (or, in some cases, both), thus purposefully expressing a valenced opinion. This often leads to a straightforward expression of sentiment, with rare use of ambiguous language or complex syntactic constructions, which may make it easier for models to learn the particularities of sentiment expression, and thus, lead to more confidence in the resulting predictions.

Generalization and adaptability. A limitation that follows from a focus on a specific type of data, such as reviews, is the SA system's decreased ability to handle other types of texts (e.g., tweets, news, etc). However, the aim of this study is an in-depth, multi-faceted analysis of the data considered, from which we hope to gain insights that may lead to building robust, general models in future work. As far as dependency on a domain, we have shown that, while we use a dataset of reviews about electronic devices as a case study, the proposed approach also provides good performances for other domains, such as movie reviews, as underlined in Section 6.1.

Ethical and Privacy Considerations. A crucial consideration in the analysis of user-generated content pertains to ethical and privacy concerns associated with potentially sensitive information. Notably, the RoProductReviews dataset utilized in all experiments within this study consists exclusively of publicly available data. Furthermore, no details regarding the identity of reviewers or any other personal information are included.

7. Conclusions and Further Work

In this study, an extensive examination of the performance of various machine learning approaches for sentiment analysis on Romanian-language texts was conducted, addressing multiple textual levels.

At the document level, the obtained results indicate that the LSI-based embedding is relevant for an automatic sentiment analysis of review documents written in Romanian, when feeding them into standard ML classifiers. Deep learning approaches, on the other hand, may provide a boost in performance when the available training dataset is sufficiently large, but at a higher cost with respect to resource utilization. Comparative studies using a range of dataset sizes would be necessary for future research in order to establish the precise contexts in which deep learning techniques outperform standard ML classifiers. Additionally, performing hyperparameter tuning would allow assessing the maximum potential of both conventional ML and deep learning classifiers.

At the sentence level, the results obtained for the task of sentiment analysis lead to the conclusion that, as opposed to the analysis at the document level, the dimensionality reduction step of the LSI algorithm hinders performance in the case of sentences, with the TF-IDF representation used in conjunction with standard ML classifiers resulting in higher performance. What is more, after examining the performance of a deep learning model for the sentiment analysis task at the sentence level, and taking into consideration the costs of deep learning methods, we conclude that standard ML algorithms are preferable for solving the task. As for future work, we intend to validate our conclusions on other datasets in Romanian and, additionally, to perform hyperparameter tuning so as to further improve the results.

In terms of the unsupervised extraction of aspect terms and categories, results show that the proposed technique based on word clustering manages to identify easily interpretable groups of words that can be viewed as aspect terms that form an aspect category. Additionally, a simple aspect category detection technique, based on word similarity in an embedding space, provides information regarding the aspect categories discussed in a review. Results for this task also reflect human interpretation to a high degree. To enhance the aspect term extraction-aspect category detection approach further, one avenue is the exploration of alternative word embeddings, such as BERT, which can potentially lead to a more precise and insightful analysis of product aspects. Finally, in future work, we aim to eliminate the manual aspect category label assignment step.

We also point out a future direction for research at all analysis levels: looking at language patterns used to express sentiment over time. This is because understanding the dynamics of sentiment expression would greatly improve the potential applicability of sentiment analysis systems, like the one this paper proposes, in real-world settings.

Author Contributions: Conceptualization, A.B., A.-D.C., D.-L.M. and C.M.-D.; Data curation, A.-D.C. and C.M.-D.; Formal analysis, A.B. and D.-L.M.; Investigation, A.B., A.-D.C., D.-L.M. and C.M.-D.; Methodology, D.-L.M. and V.P.; Resources, G.D.; Software, A.B., A.-D.C. and G.D.; Supervision, G.D.; Visualization, D.-L.M. and C.M.-D.; Writing—original draft, A.B., A.-D.C., D.-L.M., C.M.-D. and V.P.; Writing—review and editing, A.B., A.-D.C., C.M.-D. and V.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant of the European Regional Development Fund through the Competitiveness Operational Program 2014–2020, project "Produs software inovativ pentru analiza sentimentelor din textele în limba Română—SENTITEXT", MySMIS code 156284.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset collected and used in our experiments is not publicly available.

Conflicts of Interest: The study was conducted within the framework of a collaborative project involving Babeș-Bolyai University, Faculty of Mathematics and Computer Science, and T2 S.R.L., as outlined in the funding section. The research was carried out by a combined team from the Department of Computer Science of the Faculty of Mathematics and Computer Science and T2 S.R.L., sharing common objectives in exploring sentiment analysis models for the Romanian language. As a result, there is no conflict of interest in the collaboration, the investigation being performed without any commercial or financial affiliations outside of those declared in the article and that could thus be perceived as a possible conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolution Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DT	Decision Trees
GAP	Global Average Pooling
GRU	Gated Recurrent Unit
kNN	k-Nearest Neighbors
LR	Logistic Regression
LSI	Latent Semantic Indexing
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron
NB	Naïve Bayes
RF	Random Forest
RNN	Recurrent Neural Network
SOM	Self-Organizing Maps
SVM	Support Vector Machines
TF-IDF	Term Frequency–Inverse Document Frequency
VP	Voted Perceptron

Appendix A

In this appendix, we include results for another product category, namely *monitors*, in order to showcase the adaptability of the proposed approach to aspect term extraction and aspect category detection with respect to identifying relevant aspect categories for different product types.

The word groups $\mathcal{A}_w$ obtained in one example run for the category *monitor* are presented in Table A1 in order of the cumulative frequency of the containing terms.

As it can be seen, the obtained noun clusters are, as it was the case for the *laptop* category, easily interpretable. $\mathcal{A}_{w_{m1}}, \mathcal{A}_{w_{m5}}, \mathcal{A}_{w_{m6}}$ describe various characteristics of the *display* of a monitor: image quality, display technology (IPS stands for in-plane switching technology, a type of display panel technology, while TN is short for twisted nematic, a type of LED panel display technology), and other display characteristics (HD stands for *high definition*). As it is a visual output device, having more than one aspect cluster that encompasses a larger, more general aspect category *display* is to be expected, especially as far as the technology used and the performance of the monitor in terms of color accuracy, brightness and contrast in different lighting conditions. *connectivity* is also a crucial aspect when buying a monitor, as a user will want to ensure that it has the necessary ports to connect to their device ($\mathcal{A}_{w_{m2}}$). The third aspect cluster is concerned with *price*, while the fourth is with the durability and reliability of the product.

Table A1. Example clusters obtained using SOM for product type *monitor*.

	Terms	**Assigned Label**	**NPMI**	**LCH**
$\mathcal{A}_{w_{m1}}$	**culoare, intensitate, intuneric, scenecadră, expunere** *color, intensity, darkness, scenes/frames, exposure*	Display (Image Quality)	−1.000	1.441
$\mathcal{A}_{w_{m2}}$	**mufă, adaptor, cutie, cablu, usb** *socket, adapter, box, cable, USB*	Connectivity	−1.000	2.009
$\mathcal{A}_{w_{m3}}$	**asteptar, pret, produs, leu** *expectation, price, product, Romanian leu (RON)*	Price	−0.384	1.000
$\mathcal{A}_{w_{m4}}$	**săptămână, problemă, achiziție, an, saptaman, lună** *week, problem, purchase, year, month*	Durability	−0.905	2.081
$\mathcal{A}_{w_{m5}}$	**vizibilitate, pixel, ips, visualizare, pixă, unghi, tn** *visibility, pixel, IPS, visualization, angle, TN*	Display (Technology)	−0.382	1.493
$\mathcal{A}_{w_{m6}}$	**imagine, monitor, hd, display, ecran** *image, monitor, HD, display, screen*	Display (Characteristics)	0.258	1.279
$\mathcal{A}_{w_{m7}}$	**medie, calitate, pro, ok, rest, bun, super** *average, quality, pro, ok, otherwise, good, great*	Quality	−0.648	1.417

It is interesting to note the common aspect categories between the two types of products: *monitors* and *laptops*: *durability* ($\mathcal{A}_{w_{m4}}$), *price* ($\mathcal{A}_{w_{m3}}$), *connectivity* ($\mathcal{A}_{w_{m2}}$), and *display* ($\mathcal{A}_{w_{m6}}$). However, there are some differences in the aspect terms used for the categories for each product type. A stark contrast can be observed between the level of detail the *display* aspect category implied in the *monitor* reviews as opposed to the *laptop* reviews, which is an intuitive distinction, as for laptops, the display is only one of the components, while for a monitor, it can be argued that it is the most important one. Alternatively, the aspect of *durability/reliability* tends to be characterized by temporal words (i.e., *year*, *month*, *duration*, *time*, *beginning*) accompanied by synonyms of the word *usage* for both product types.

Table A2 provides a series of example reviews from the product category *monitors*, to exemplify the performance of the proposed aspect category detection technique on a different product category. In general, for this product type, we observe that short reviews such as R_{m_7}, which do not reference any particular aspects of the product, are dominated by the *quality/general* category. Other similar reviews with a marked presence of this aspect are: *"Este destul de bun dar nu il recomand./It's decent, but I don't recommend it."* (0.903), *"Un monitor bun, claritate buna/A good monitor, good clarity."* (0.988), *"E chiar bun imi place/It's actually good, I like it"* (0.936) or even simply *"Bun/Good"* (0.797). Alternatively, high *quality/general* scores are also obtained for long reviews in which no particular aspects are discussed. For example, a review consisting of approximately 50 words through which indications about a workaround for an issue (lack of component) was assigned a score of 0.999 for the *quality/general* aspect category.

However, in reviews such as R_{m_2} or R_{m_8}, the *quality/general* aspect category is present to a significant extent (for instance, *bun pentru birou/good for the office* in review R_{m_2} is a general evaluation), but so are other factors like connectivity (R_{m_2}—*conexiune VGA/VGA*

connection, R_{m_8}—*port HDMI, DisplayPort, USB*), which are identified by the proposed method accordingly.

In addition, as it can be seen, aspects that are not explicitly referred (e.g., *display* in R_{m_4} or *durability/reliability* in R_{m_1} and R_{m_3} are also indicated by our approach, which supports the conclusion drawn regarding the wide applicability of the proposed aspect term extraction-aspect category detection pipeline.

Table A2. Aspect category detection results with respect to a set of reviews from product category *monitor*.

	Review Text	Display (Image Quality)	Connectivity	Price	Durability/ Reliability	Display (Technology)	Display Features	Quality/ General
R_{m_1}	***După o luna au apărut dungi pe ecran!!!*** *After a month, stripes appeared on the screen!*	0.059	0.010	0.011	0.672	0.064	0.157	0.027
R_{m_2}	***Doar conexiune VGA și atât. Bun pentru birou.*** *Only VGA connection, and that's it. Good for the office*	0.067	0.179	0.065	0.055	0.065	0.080	0.490
R_{m_3}	***Am monitorul de mai mult de 3 ani si sunt foarte multumit de el. Il folosesc doar pt gaming si se ridica așteptărilor. Cumpărați cu încredere*** *I've had the monitor for more than 3 years, and I am very satisfied with it. I use it exclusively for gaming, and it meets expectations. Buy with confidence.*	0	0.001	0.105	0.817	0.002	0.006	0.068
R_{m_4}	***Are ghosting destul de urat.*** *Is ghosting quite ugly*	0.176	0.057	0.082	0.060	0.178	0.183	0.264
R_{m_5}	***Pret calitate, DEZAMAGITOR!*** *Price quality, DISAPPOINTING!*	0.005	0.003	0.147	0.037	0.020	0.017	0.771
R_{m_6}	***Nu am fost atent la detalii si am comandat unul cu port serial in loc de hdmi. Are doar o singura intrare si depinde de model…*** *I wasn't careful with the details, and I ordered one with a serial port instead of HDMI. It has only one input, and it depends on the model.*	0.006	0.839	0.003	0.003	0.113	0.027	0.009
R_{m_7}	***Super ok! Se comporta bine!*** *Super ok! It performs well!*	0	0	0.001	0.007	0	0	0.992
R_{m_8}	***Business as usual de la Dell. Un monitor excelent. ii dau totusi 4 stele pentru ca folosit cu doua deviceuri, dureaza foarte mult functia de autoscan, este mai rapid sa selectez manual input source cand am nevoie sa trec de la un PC la celalalt. E destul de incomod si faptul ca are doar un singur port HDMI si unul singur DisplayPort. USB-urile sunt excelente pentru cei fara docking station. Evident ca cei care au un singur device nu sunt catusi de putin incomodati de micile inconveniente sus mentionate.*** *Business as usual from Dell. An excellent monitor. However, I'm giving it four stars because when used with two devices, the autoscan function takes a long time. It's faster to manually select the input source when I need to switch from one PC to the other. It's quite inconvenient that it has only one HDMI port and one DisplayPort. The USB ports are excellent for those without a docking station. Clearly, those with only one device aren't bothered at all by the minor inconveniences mentioned above.*	0.012	0.473	0	0	0.002	0	0.512

References

1. Liu, B. *Sentiment Analysis and Opinion Mining*; Springer Nature: Berlin/Heidelberg, Germany, 2022.
2. Vernikou, S.; Lyras, A.; Kanavos, A. Multiclass sentiment analysis on COVID-19-related tweets using deep learning models. *Neural Comput. Appl.* **2022**, *34*, 19615–19627. [CrossRef]
3. Hasib, K.M.; Habib, M.A.; Towhid, N.A.; Showrov, M.I.H. A Novel Deep Learning based Sentiment Analysis of Twitter Data for US Airline Service. In Proceedings of the 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh, 27–28 February 2021; pp. 450–455. [CrossRef]
4. Nagamanjula, R.; Pethalakshmi, A. Twitter sentiment analysis using Dempster shafer algorithm based feature selection and one against all multiclass SVM classifier. *Int. J. Adv. Res. Eng. Technol.* **2020**, *11*, 163–185. [CrossRef]
5. Mukta, M.S.H.; Islam, M.A.; Khan, F.A.; Hossain, A.; Razik, S.; Hossain, S.; Mahmud, J. A comprehensive guideline for Bengali sentiment annotation. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *21*, 1–19. [CrossRef]
6. Elbagir, S.; Yang, J. Twitter sentiment analysis using natural language toolkit and VADER sentiment. In Proceedings of the International Multiconference of Engineers and Computer Scientists, Hong Kong, 13–15 March 2019; Volume 122, p. 16.
7. Su, J.; Chen, Q.; Wang, Y.; Zhang, L.; Pan, W.; Li, Z. Sentence-level Sentiment Analysis based on Supervised Gradual Machine Learning. *Sci. Rep.* **2023**, *13*, 14500. [CrossRef] [PubMed]
8. Liu, B. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*; Cambridge University Press: Cambridge, UK, 2020.
9. Chebolu, S.U.S.; Dernoncourt, F.; Lipka, N.; Solorio, T. Survey of Aspect-based Sentiment Analysis Datasets. *arXiv* **2023**, arXiv:cs.CL/2204.05232.
10. Zhang, W.; Li, X.; Deng, Y.; Bing, L.; Lam, W. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 11019–11038. [CrossRef]
11. He, R.; Lee, W.S.; Ng, H.T.; Dahlmeier, D. An Unsupervised Neural Attention Model for Aspect Extraction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 388–397. [CrossRef]
12. Shi, T.; Li, L.; Wang, P.; Reddy, C.K. A simple and effective self-supervised contrastive learning framework for aspect detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 13815–13824.
13. Chebolu, S.U.S.; Rosso, P.; Kar, S.; Solorio, T. Survey on aspect category detection. *ACM Comput. Surv.* **2022**, *55*, 1–37. [CrossRef]

14. Luo, L.; Ao, X.; Song, Y.; Li, J.; Yang, X.; He, Q.; Yu, D. Unsupervised Neural Aspect Extraction with Sememes. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, 10–16 August 2019; pp. 5123–5129.
15. Tulkens, S.; van Cranenburgh, A. Embarrassingly Simple Unsupervised Aspect Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3182–3187. [CrossRef]
16. Dumitrescu, S.D.; Rebeja, P.; Lorincz, B.; Gaman, M.; Avram, A.; Ilie, M.; Pruteanu, A.; Stan, A.; Rosia, L.; Iacobescu, C.; et al. LiRo: Benchmark and leaderboard for Romanian language tasks. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, Online, 6–14 December 2021.
17. Tache, A.; Gaman, M.; Ionescu, R.T. Clustering Word Embeddings with Self-Organizing Maps. Application on LaRoSeDa—A Large Romanian Sentiment Data Set. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 19–23 April 2021; pp. 949–956. [CrossRef]
18. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
19. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8440–8451. [CrossRef]
20. Dumitrescu, S.D.; Avram, A.M.; Pyysalo, S. The birth of Romanian BERT. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 4324–4328. [CrossRef]
21. Masala, M.; Ruseti, S.; Dascalu, M. RoBERT—A Romanian BERT Model. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6626–6637. [CrossRef]
22. Masala, M.; Iacob, R.C.A.; Uban, A.S.; Cidota, M.; Velicu, H.; Rebedea, T.; Popescu, M. jurBERT: A Romanian BERT Model for Legal Judgement Prediction. In Proceedings of the Natural Legal Language Processing Workshop 2021, Punta Cana, Dominican Republic, 10 November 2021; Association for Computational Linguistics: Punta Cana, Dominican Republic, 2021; pp. 86–94. [CrossRef]
23. Avram, A.; Catrina, D.; Cercel, D.; Dascalu, M.; Rebedea, T.; Pais, V.F.; Tufis, D. Distilling the Knowledge of Romanian BERTs Using Multiple Teachers. *arXiv* **2021**, arXiv:2112.12650.
24. Nicolae, D.; Yadav, R.; Tufis, D. A Lite Romanian BERT:ALR-BERT. *Computers* **2022**, *11*, 57. [CrossRef]
25. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Valencia, Spain, 3–7 April 2017; Association for Computational Linguistics: Valencia, Spain, 2017; pp. 427–431.
26. Burlăcioiu, C.; Boboc, C.; Mitre, B.; Dragne, I. Text Mining in Business. A Study of Romanian Client's Perception with Respect to Using Telecommunication and Energy Apps. *Econ. Comput. Econ. Cybern. Stud. Res.* **2023**, *57*, 221–234.
27. Jones, K.S. A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* **2021**, *60*, 493–502. [CrossRef]
28. Russu, R.M.; Dinsoreanu, M.; Vlad, O.L.; Potolea, R. An opinion mining approach for Romanian language. In Proceedings of the 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 4–6 September 2014; pp. 43–46. [CrossRef]
29. Esuli, A.; Sebastiani, F. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In Proceedings of the International Conference on Language Resources and Evaluation, European Language Resources Association (ELRA), Genoa, Italy, 22–28 May 2006.
30. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.
31. Echim, S.V.; Smădu, R.A.; Avram, A.M.; Cercel, D.C.; Pop, F. Adversarial Capsule Networks for Romanian Satire Detection and Sentiment Analysis. In *Lecture Notes in Computer Science*; Springer Nature: Cham, Switzerland, 2023; Volume 13913, pp. 428–442. [CrossRef]
32. Neagu, D.C.; Rus, A.B.; Grec, M.; Boroianu, M.A.; Bogdan, N.; Gal, A. Towards Sentiment Analysis for Romanian Twitter Content. *Algorithms* **2022**, *15*, 357. [CrossRef]
33. Istrati, L.; Ciobotaru, A. Automatic Monitoring and Analysis of Brands Using Data Extracted from Twitter in Romanian. In *Intelligent Systems and Applications*; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 55–75. [CrossRef]
34. Coita, I.F.; Cioban, S.; Mare, C. Is Trust a Valid Indicator of Tax Compliance Behaviour? A Study on Taxpayers' Public Perception Using Sentiment Analysis Tools. In *Digitalization and Big Data for Resilience and Economic Intelligence*; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 99–108. [CrossRef]
35. Buzea, M.C.; Trăuşan-Matu, Ş.; Rebedea, T. A three word-level approach used in machine learning for Romanian sentiment analysis. In Proceedings of the 2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet), Galați, Romania, 10–12 October 2019; pp. 1–6.
36. Roșca, C.M.; Ariciu, A.V. Unlocking Customer Sentiment Insights with Azure Sentiment Analysis: A Comprehensive Review and Analysis. *Rom. J. Pet. Gas Technol.* **2023**, *4*, 173–182. [CrossRef]
37. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 168–177. [CrossRef]
38. Popescu, A.M.; Etzioni, O. Extracting Product Features and Opinions from Reviews. In *Natural Language Processing and Text Mining*; Springer: London, UK, 2007; pp. 9–28. [CrossRef]

39. Wu, Y.; Zhang, Q.; Huang, X.J.; Wu, L. Phrase dependency parsing for opinion mining. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3, Singapore, 6–7 August 2009; pp. 1533–1541. [CrossRef]
40. Hai, Z.; Chang, K.; Kim, J.j. Implicit feature identification via co-occurrence association rule mining. In *Computational Linguistics and Intelligent Text Processing*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 393–404. [CrossRef]
41. Schouten, K.; Van Der Weijde, O.; Frasincar, F.; Dekker, R. Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis with Co-occurrence Data. *IEEE Trans. Cybern.* **2017**, *48*, 1263–1275. [CrossRef] [PubMed]
42. Titov, I.; McDonald, R. Modeling Online Reviews with Multi-Grain Topic Models. In Proceedings of the 17th International Conference on World Wide Web, Beijing, China, 21–25 April 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 111–120. [CrossRef]
43. Brody, S.; Elhadad, N. An Unsupervised Aspect-Sentiment Model for Online Reviews. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, CA, USA, 2–4 June 2010; Association for Computational Linguistics: Los Angeles, CA, USA, 2010; pp.804–812.
44. García-Pablos, A.; Cuadros, M.; Rigau, G. W2VLDA: Almost unsupervised system for Aspect Based Sentiment Analysis. *Expert Syst. Appl.* **2018**, *91*, 127–137. [CrossRef]
45. Ghadery, E.; Movahedi, S.; Faili, H.; Shakery, A. An Unsupervised Approach for Aspect Category Detection Using Soft Cosine Similarity Measure. *arXiv* **2018**, arXiv:1812.03361.
46. Sia, S.; Dalmia, A.; Mielke, S.J. Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too! In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 1728–1736. [CrossRef]
47. Viegas, F.; Canuto, S.; Gomes, C.; Luiz, W.; Rosa, T.; Ribas, S.; Rocha, L.; Gonçalves, M.A. CluWords: Exploiting Semantic Word Clustering Representation for Enhanced Topic Modeling. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 753–761. [CrossRef]
48. Comito, C.; Forestiero, A.; Pizzuti, C. Word Embedding Based Clustering to Detect Topics in Social Media. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Thessaloniki, Greece, 14–17 October 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 192–199. [CrossRef]
49. Boroș, T.; Dumitrescu, S.D.; Burtica, R. NLP-Cube: End-to-End Raw Text Processing With Neural Networks. In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Brussels, Belgium, 31 October–1 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 171–179. [CrossRef]
50. Lupea, M.; Briciu, A. Studying emotions in Romanian words using Formal Concept Analysis. *Comput. Speech Lang.* **2019**, *57*, 128–145. [CrossRef]
51. Deerwester, S.C.; Dumais, S.T.; Landauer, T.K.; Furnas, G.W.; Harshman, R.A. Indexing by Latent Semantic Analysis. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 391–407. [CrossRef]
52. Řehůřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 22 May 2010; pp. 45–50. [CrossRef]
53. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv* **2018**, arXiv:1811.03378.
54. Nwankpa, C.E.Advances in optimisation algorithms and techniques for deep learning. *Adv. Sci. Technol. Eng. Syst. J.* **2020**, *5*, 563–577. [CrossRef]
55. Farhadloo, M.; Rolland, E. Multi-class sentiment analysis with clustering and score representation. In Proceedings of the 2013 IEEE 13th International Conference on Data Mining Workshops, Dallas, TX, USA, 7–10 December 2013; pp. 904–912.
56. Tache, A.M.; Gaman, M.; Ionescu, R.T. Clustering word embeddings with self-organizing maps. application on laroseda—A large romanian sentiment data set. *arXiv* **2021**, arXiv:2101.04197.
57. Bouma, G. Normalized (Pointwise) Mutual Information in Collocation Extraction. *Proc. Bienn. GSCL Conf.* **2009**, *30*, 31–40.
58. Lau, J.H.; Newman, D.; Baldwin, T. Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, 26–30 April 2014; Association for Computational Linguistics: Gothenburg, Sweden, 2014; pp. 530–539. [CrossRef]
59. Leacock, C. Combining local context and WordNet similarity for word sense identification. In *WordNet: A Lexical Reference System and Its Application*; The MIT Press: Cambridge, MA, USA, 1998; pp. 265–283.
60. Dumitrescu, S.D.; Avram, A.M.; Morogan, L.; Toma, S.A. RoWordNet—A Python API for the Romanian WordNet. In Proceedings of the 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Iasi, Romania, 28–30 June 2018; pp. 1–6. [CrossRef]
61. Freund, Y.; Schapire, R.E. Large Margin Classification Using the Perceptron Algorithm. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, Madison, WI, USA, 24–26 July 1998; Association for Computing Machinery: New York, NY, USA, 1998; pp. 209–217. [CrossRef]