

Article

Revealing People's Sentiment in Natural Italian Language Sentences

Andrea Calvagna , Emiliano Tramontana *  and Gabriella Verga 

Dipartimento di Matematica e Informatica, University of Catania, 95125 Catania, Italy;
andreamario.calvagna@unict.it (A.C.); gabriella.verga@unict.it (G.V.)

* Correspondence: tramontana@dmi.unict.it; Tel.: +39-095-7383008

Abstract: Social network systems are constantly fed with text messages. While this enables rapid communication and global awareness, some messages could be aptly made to hurt or mislead. Automatically identifying meaningful parts of a sentence, such as, e.g., positive or negative sentiments in a phrase, would give valuable support for automatically flagging hateful messages, propaganda, etc. Many existing approaches concerned with the study of people's opinions, attitudes and emotions and based on machine learning require an extensive labelled dataset and provide results that are not very decisive in many circumstances due to the complexity of the language structure and the fuzziness inherent in most of the techniques adopted. This paper proposes a deterministic approach that automatically identifies people's sentiments at the sentence level. The approach is based on text analysis rules that are manually derived from the way Italian grammar works. Such rules are embedded in finite-state automata and then expressed in a way that facilitates checking unstructured Italian text. A few grammar rules suffice to analyse an ample amount of correctly formed text. We have developed a tool that has validated the proposed approach by analysing several hundreds of sentences gathered from social media: hence, they are actual comments given by users. Such a tool exploits parallel execution to make it ready to process many thousands of sentences in a fraction of a second. Our approach outperforms a well-known previous approach in terms of precision.

Keywords: natural language processing; sentiment analysis; information extraction



Citation: Calvagna, A.; Tramontana, E.; Verga, G. Revealing People's Sentiment in Natural Italian Language Sentences. *Computers* **2023**, *12*, 241. <https://doi.org/10.3390/computers12120241>

Academic Editor: Paolo Bellavista

Received: 8 October 2023

Revised: 14 November 2023

Accepted: 18 November 2023

Published: 21 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The interpretation of sentences expressed in natural language is a very important research field because it allows us to achieve a variety of results, as shown by several widely used applications [1]. E.g., many companies need to analyse opinions that users have expressed on some products, and in many cases, social network systems need an automatic means to limit the spread of hateful messages that could hurt readers. This complex and useful work is called Sentiment Analysis (SA) and is the computational study of people's opinions, evaluations, attitudes and emotions [2]. This field has gained more and more attention because thousands of people tend to have web-based relationships (e.g., being members of internet social networks and forums), both for business and/or pleasure, as an integral part of their lives. Hence, many Internet pages reflect people's thoughts, feelings and opinions. As a consequence, researchers have focused on web forums to extract text and show the usefulness of their techniques for classifying feelings in documents or sentences [3].

SA tools can be applied to different levels of context granularity: document level, sentence level and aspect-based level [2]. At the document level, the task is to classify a whole document, by expressing an overall positive or negative opinion [4]. This level of analysis assumes that the document under examination expresses opinions on a single entity (e.g., a product). At the sentence level, the task is to classify whether each sentence expresses a positive, negative or neutral opinion. Before analysing the polarity (orientation)

of the sentence, it is necessary to determine whether the sentence is subjective or objective because the objective ones do not express opinions, unlike subjective ones [5]. By firstly separating sentences into one of the two categories, then the subjective sentences can be labelled as positive or negative. At the aspect-based level, the goal is to look directly at the opinion itself. A document or a sentence may refer to different entities; thus, this level focuses on the recognition of all sentiments and the aspects to which they refer. It is based on the idea that an opinion consists of a sentiment (positive or negative) and a target of the opinion [6].

English and a few other languages have been extensively analysed, and for them, large annotated datasets have been produced, which are instrumental in the consequent machine learning (ML) approaches. However, for languages exhibiting a scarcity of annotated datasets, the efficacy of ML-based approaches is much reduced, and such approaches yield a low level of precision and recall [7–9]. Moreover, ML-based approaches, or other approaches based on large datasets (such as, e.g., transformers [10]), make extensive use of computational and storage resources both for training and when having to produce a result by processing a request. This makes them unsuitable when dealing with a high rate of requests, when cost-effective solutions are needed, or when a client-side solution is appropriate.

We propose a new approach for Italian language SA that processes text by means of a dictionary for significant words and properly devised automata where each automaton represents a grammar-legal category of sentences. Our proposed automata quickly analyses sentences by interpreting underlying Italian grammar; then, it classifies them as positive, neutral or negative. Three finite-state automata, embedding Italian grammar rules, are designed and embedded into our developed tool to automatically identify sentences that express an opinion. Despite the complexity of the Italian language, we have obtained high-quality results in terms of precision and recall for the identification of sentence orientation. Unlike most recent approaches that are ML-based, our approach does not need an annotated dataset nor a large amount of computational resources for training and execution [11]. Both the dictionary used by our approach and our tool are very small, making the tool prone to be used on any hardware, even on the client side. Moreover, our results are explainable, unlike those of ML-based models; hence, the classification of sentences can be examined and possibly improved by tuning the used dictionary or adapting the automata to new shapes of sentences. Compared with other approaches [12], our approach shows better performances in terms of precision and recall due to the careful analysis of grammar in our proposed automata. We have developed the said tool to execute according to the MapReduce style in Java to easily achieve parallelism, hence speeding up execution as much as possible [13].

This paper is organized as follows. Section 2 discusses the related works. Section 3 explains the proposed approach to performing SA. Section 4 describes the experiments and results as well as a comparison of results with a well-known algorithm for SA (Serendio). Section 5 reports the parallel version of the algorithm using Java 8 support for MapReduce. Section 6 gives remarks on the provided approach and tool. We draw our conclusions in Section 7.

2. Related Works

A large number of papers in the SA field have been published in the literature over the years and have implemented various approaches [6,14–16]. This research field has shown rapid growth: almost doubling the number of documents per year every two years. In recent systematic literature reviews [8,17,18], more than five thousands primary studies were counted that were conducted from 2010 to 2021. SA approaches can be grouped according to the following classification: lexicon-based approaches, ML-based approaches and hybrid approaches [14].

Lexicon-based approaches use a collection of known terms and a set of rules to determine the orientations of opinions in a phrase; therefore, they are based on text analysis.

They can be approximately divided into two groups: dictionary-based and corpus-based. Dictionary-based approaches rely on finding opinion seed words and then consulting a dictionary for their synonyms and antonyms. For the latter support, the most widely used dictionary is WordNet [19]. One of the algorithms for WordNet's synonym-finding defines a distance $d(t1, t2)$ as the length of the shortest path between two terms $t1$ and $t2$ in WordNet [20]. Corpus-based approaches begin with a seed list of opinion words and then find other opinion words in a large corpus to identify the most-similar context and then the orientation. They use statistical or semantic methods to find sentiment polarity.

Machine learning (ML) has been the traditional and mainstream approach to classify the emotional tone of textual input [21,22]. ML algorithms use linguistic features, starting with a set of training records wherein each record is labelled as belonging to a class, to derive a model that is then used to predict a class label for a given instance of an unknown class [23]. Text classification proposals using ML can be approximately divided into two groups: supervised learning and unsupervised learning. Supervised learning methods depend on the existence of extensive labelled training datasets. In the research literature, there are many kinds of such classifiers, such as Naïve Bayes [24–26], Convolutional Neural Networks [27,28], Support Vector Machines (SVM) [26] and Long Short-Term Memory [29], that have been applied to SA. Unsupervised learning methods do not require prior training to analyse data and aim to measure how far a word is inclined towards positive or negative; hence, a labelled dataset is still needed for such a priori knowledge. E.g., one such method divides the documents into sentences and finds a category for each sentence using keyword lists and for each sentence computes a similarity measure [30].

These inherently data-driven learning approaches need an extensive curated dataset and long training, and results are not always accurate and can be affected by bias due to potentially unbalanced training data [14,31]. As no single tool has been found to be sufficiently reliable on its own, some SA solutions use an ensemble approach by combining predictions from multiple models into hybrid tools to improve performance and achieve better accuracy [16,32–35].

However, in contrast to high-resourced languages like English, developing models for under-resourced languages (or for multilingual contexts) is harder, as the scarcity of annotated datasets limits progress in this area [7–9]. In fact, initially, the scientific literature of SA has almost exclusively focused on well-studied languages such as English, German, Chinese, etc. Nevertheless, over time, many researchers have started SA tasks in low-resource languages, having recognised the need to identify the sentiment for text written in languages that are not the mostly used ones. Approaches for several low-resources languages of the Asian geographic area have recently been studied by several authors like El-Haj et al. [36] for the Arabic language, Le et al. [37] for Indonesian, Rajabi et al. [38] on the Persian language, and Gangula and Mamidi [39] for Telugu language, which is one of the six languages designated as a classical language by the Government of India and the 14th most-spoken native language in the world. Nasib et al. [40] in 2018 obtained good accuracy results converting natural Bengali language to text with an adequately trained model based on an open-source text-to-speech Java framework. A very recent work by Altaf et al. [41] exploits linguistic features of Urdu language (a major Pakistani language) for sentence-level SA and classifies idioms and proverbs using classical machine learning techniques. The above approaches use either machine learning or deep learning; accordingly, they need an ample training set and relevant computational resources [11].

With the advancements in natural language processing (NLP) in recent years, researchers have shifted their focus toward advanced artificial intelligence (AI) techniques like deep learning and transfer learning [18,42–45]. Particularly, the *transformer* model introduced by Vaswani et al. [10] provided the groundwork for a new line of language *generative* models, including the GPT model, which marked a milestone in this field. Recently, Zhong et al. [46] investigated the ability to apply ChatGPT, a GPT model variant, to SA. However, we note that while GPT tools are based on a general-purpose language model capable of a wide range of language-related tasks, natural language SA tools are specialised for the

specific task of analysing sentiment in text data. They can be tailored to a specific domain or language, fine-tuned on relevant data, and ultimately outperform a general language model. They can offer specialised features, like aspect-based SA (analysing sentiment toward specific aspects or subjects within a text) and are often faster and scale better if processing large volumes of text in real-time or near-real-time is required.

In fact, there are many very specific challenges that are peculiar to SA, such as those related to finding the proper meaning according to the position of keywords in the sentence [6], finding the correct association between negation and sentiment [6]—that is, the application scope of negations [47–49]—and/or classifying a comparative sentence. For this, specific solutions and tools for SA show better results than general AI-based tools.

As for specific issues that can be met when analysing a sentence, one of the cases is sentences using comparatives and ones using superlatives. In the first case, the subject expresses a comparison, while in the second, the subject expresses an opinion [50]. In [51], the authors analysed all comparative opinions (including comparatives and superlatives). Moreover, it can be difficult to find the correct association between the sentiment and the subject to which it refers [52]. Generally, such difficulties mostly derive from the pre-assigned maximum number of steps separating two meaningful keywords in a sentence, which is commonly given in existing approaches. A well-known algorithm is Serendio [12], which limits the distance between negation and sentiment to two steps (two words) so as to have high reliability of the bond between negation and sentiment; however, this causes the removal of a great portion of sentences. The inconvenience is mainly found in the Italian language, where sentences are usually more complex than those in the English language. This limitation leads to an incorrect association between two words. Instead, our devised approach is much more robust, i.e., regardless of the distance in terms of words between the product and the sentiment, orientation will be correctly determined.

In this paper, we target sentence-level classification and identify direct opinions. In addition, we succeed in overcoming the challenge of correctly associating negation and sentiment.

3. Proposed Approach

3.1. Overview

In our approach, sentences written in the Italian language are automatically processed to identify a sentiment (if it exists in the sentence) and attribute a label such as: positive, neutral or negative. Our approach consists of several steps, which are all performed automatically. Firstly, each sentence is parsed to identify words at the input text and classify them according to their grammatical category (i.e., verb, article, adverb, etc.); such an identification is performed by finding matches between the input text and a dictionary. Secondly, the resulting sequence of categories is passed to a set of three automata. Each automaton has been designed according to grammar rules for the Italian language and recognises the meanings of several types of sentences adhering to proper grammar rules. Each automaton has two final states: one for identifying positive sentiment and the other for negative sentiment. When an analysed sentence is partially accepted by an automaton, however, if a final state (among the two identified) cannot be reached, then the sentence is associated with a neutral sentiment. Finally, when a sentence cannot be accepted (because it cannot enter the first state) by an automaton, then it is determined that such an automaton is not appropriate for recognising the sentence's sentiment; no label is given in this case.

Figure 1 shows an overview of our approach, where each column in the diagram represents a relevant activity. Data collection refers to the gathering of sentences from web sites, which will be analysed; preprocessing is the mapping of words in sentences to grammar categories (see Section 3.2); feature extraction recognises the form of a sentence (according to grammar rules embedded into automata); classification gives the orientation according to the final state reached by an automata (see Section 3.3); finally, evaluation is performed only to validate the approach when data labels are compared with the results of the automatic characterisation of the sentiment.

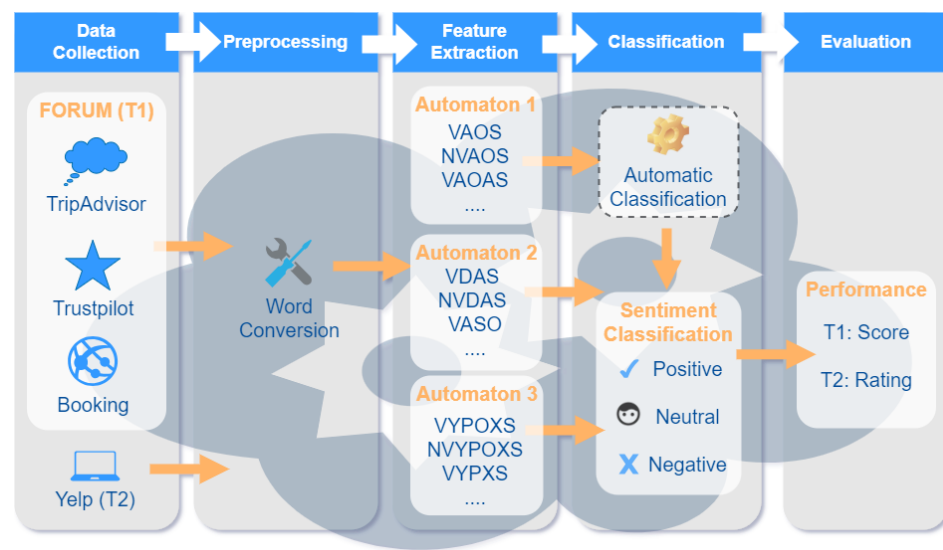


Figure 1. The flow of steps performing text analysis and sentiment recognition.

3.2. Preliminary Lexicon Analysis

To determine a phrase's sentiment, firstly, it is necessary to parse the text and perform the identification of meaningful words. Therefore, the first step of the proposed approach extracts all the words from a sentence and classifies each of them according to the following categories. Such a classification is automatically performed by looking up the words found in sentences in a dictionary mapping words to corresponding categories. The following are the categories of words.

- **Verbs (V):** a category for the Italian verbs; a subset of all the Italian verbs has been gathered, comprising the verbs that are mostly used to express an opinion. Such a subset includes seven verbs: *essere* (be) and *avere* (have) with their synonyms *trovare* (find), *sembrare* (seem, look like), *restare* (stay), *vedere* (see), and *rimanere* (remain, stay).
- **Articles (A):** a category for the entire list of articles for the Italian language, such as, e.g., *il*, *lo* (the), etc., and *un* (a), etc. Such a list consists of nine articles.
- **Sentiments (S):** a category for the words related to emotions, which are not effected by the context (except when using negations), e.g., *bello* (beautiful) and *brutto* (ugly). This list consists of two subsets: a set expressing positive sentiments, such as, e.g., *carino* (nice), *buono* (good, delicious), *dolce* (sweet), etc.; and a set for negative sentiments, such as e.g., *cattivo* (bad) or *pesante* (heavy), etc. Each of the two subsets is used to guide the automaton's transitions towards one of the two corresponding final states (for positive or negative orientation, respectively). The list of positive sentiments consists of 111 words, whereas the list of negative sentiments consists of 43 words. Table 1 shows a partial list of words expressing a sentiment (positive or negative) that we used in our approach.
- **Negations (N):** only contains the word *non* (not). Negation is tricky because the same word accompanied by negation may change the orientation of a sentence. E.g., 'Il mio nuovo computer è potente' (my new computer is powerful) versus 'Il mio nuovo computer non è potente' (my new computer is not powerful). We have included in the proposed automata the ability to process negation in such a way that a negation and a positive sentiment makes it transition towards the end state expressing a negative orientation and vice versa for negation and negative sentiment.
- **Adverbs (D):** a list of adverbs such as *veramente* (really), *poco* (little), and *molto* (a lot). This list is made up of two subsets of positive adverbs, such as, e.g., *molto* (very) or *davvero* (really), and negative adverbs such as *poco* (little) or *purtroppo* (unfortunately), etc. As for the two lists of words for positive and negative sentiments, also, the two lists of adverbs guide towards one of the two final states expressing positive or

negative orientation. Table 2 shows a partial list of words functioning as adverbs and a partial list of prepositions. The list of positive adverbs consists of 104 words, whereas the list of negative adverbs consists of 16 words.

- Pronoun (Y): only contains the word *uno* (one).
- Superlative (X): only contains the word *più* (most).
- Prepositions (P): a list of prepositions, such as *da* (from), *dentro* (into), *in* (in), etc., and with articles, such as, e.g., *dagli* (from the), etc. The collected prepositions are 42 (Table 2 shows a partial sample).
- Other (O): it consists of words not belonging to any of the above lists (we need not form a list for such words).

Table 1. Lists of sample words suggesting positive or negative sentiments.

Positive Sentiment	Negative Sentiment
affascinante (charming)	brutto (ugly)
ampio (ample)	cattivo (bad, nasty)
attraente (attractive)	guerra (war)
bello (beautiful)	inquietante (disturbing)
buono (good, delicious)	odioso (hateful)
carino (pretty)	pesante (heavy)
dolce (sweet)	problema (problem)
gentile (dear, gentle)	scarso (poor)
gradevole (nice)	sporco (dirty)
grazioso (pretty)	
migliore (best)	
potente (powerful)	
pulito (clean)	
resistente (durable)	

The Sentiment (S) and Adverb (D) categories have two subsets because each subset conveys a meaning (and consequently an orientation for the sentence) that is opposite that of the other subset. I.e., each subset guides the automaton towards a final state that is different than the final state that the other subset would determine.

Table 2. Lists of sample words for adverbs and prepositions.

Positive Adverb	Negative Adverb	Preposition
esatto (exact)	per nulla (in no way)	ad (for)
pure (also)	niente (anything)	in (in)
certamente (sure)	quasi (almost)	dalla (from)
assolutamente (absolutely)	purtroppo (unfortunately)	alla (at the)
semplicemente (simply)	neanche (neither)	agli (at the)
molto (very)	neppure (neither)	sulle (on)
sempre (always)	appena (just)	nello (in)
davvero (really)		
veramente (really)		

Let us analyse the sentence ‘Il ragazzo è amichevole’ (The boy is friendly). By performing slight modifications to the above sentence, we can change its orientation, e.g., ‘Il ragazzo è molto amichevole’ (The boy is very friendly); both are positive sentences. Instead, ‘Il ragazzo non è molto amichevole’ (The boy is not very friendly) and ‘Il ragazzo è poco amichevole’ (The boy is unfriendly) are both negative sentences.

For the correct interpretation of the sentence, firstly, our approach associates each word in the sentence with the letter identifying the category of the word, e.g., ‘Il ragazzo è amichevole’ is associated with AOVVS because there is an article (A), a not identified word (O), a verb (V), and a sentiment (S), respectively. Table 3 lists some sample sentences and their associated sequences of categories found by recognising each word in them.

Table 3. Sample sentences and the associated sequences of word categories, e.g., *Il* (the) becomes A, *nuovo* (new) becomes O, etc.

Sentence	Category Sequence
Il nuovo iPhone è il telefono più bello di quest'anno	AOOVAOXSPPO
The new iPhone is the most beautiful phone this year	
La villa di Catania è molto affascinante	AOPOVAS
The villa in Catania is very charming	
L'hotel non ha un aspetto gradevole	AONVAOS
The hotel does not look nice	

3.3. Performing Sentence Analysis

Once the preliminary step has been performed, which maps each word into its own grammar category, each sequence of categories is given to an automaton, which mechanically performs some state transitions following the input sequence and can end up with a final state.

We have implemented three finite state automata; each one corresponds to a class of sentences adhering to Italian grammar rules. For each automaton, we have two final states: one for positive sentiment and one for negative sentiment. When the automaton has used all the categories (letters) of the input sequence and one of the two final states has been reached, then the final state determines whether the sentiment expressed in the original sentence is positive or negative. Specifically, a sentence is given a score of +1 if it is accepted by a final state that indicates a positive orientation, and it is given score of −1 if it is accepted by a final state that indicates a negative orientation. When the automaton has used all the parts of the sequence without reaching one of the final states (hence, partially accepting the given sentence), then a score equal to 0 is given, which identifies a neutral sentiment. Typically, each sentence is recognised by only one automaton, as each automaton expresses a distinct Italian grammar rule.

When the sequence of letters identifying categories is given as input to an automaton, its initial state could happen to be unsuitable to accept the first letter of the sequence, as sentences could have many words that are not relevant for the identification of the sentiment; hence, the design of the automaton had not considered that they could be present. Then, the sequence is stripped of the first letter, and the automaton is tried again giving as input the remaining part of the sequence. Stripping the first letter is performed many times until the sequence is accepted or it becomes empty. In the last case, the result is that the sequence was not recognised by the automaton.

E.g., the sentence 'Huawei Mate 60 Pro è lo smartphone migliore del 2023' (Huawei Mate 60 Pro is the best smartphone for 2023) gives the sequence OOOOVAOSPO. When feeding such a sequence to our automata, let us say that the first letter cannot be accepted by the first state of any automata (in the following, both Automaton 2 and Automaton 3 cannot accept it). Then, by stripping the first four Os, one at a time, we obtain VAOSPO. The sequence VAOS is accepted by the first automaton and the final state is q6, giving a score equal to +1 and indicating a positive sentiment.

As a consequence, an exceptional ability to scale is clearly inherent to this approach since all rules can be checked in parallel, irrespective of their number. Overall computation ends normally as soon as any automaton corresponding to a grammar rule reaches a final state. Moreover, another significant performance speed-up has been easily achieved with respect to the number of sentences that have to be analysed (there could be up to several millions in some real cases encompassing server-side social network systems) thanks to a parallel design adopted for the automaton algorithm (see Section 5).

3.3.1. Automaton 1

The first finite state automaton represents all sentences having the form: Verb + Article + Other + Sentiment, which is a sequence VAOS, named according to the initials of the categories of words. Many possible variations are also recognised by design, since if we

identify additional words of different categories occurring within the said sequence, the automaton can recognise and skip them to find the expressed sentiment orientation. The variations to be recognised are a set that is controlled by the automaton itself thanks to the labels that regulate the transitions among some states and allow some controlled repetitions of words.

Figure 2 shows the automaton we designed to recognise the sequence VAOS and its equivalent variations: the initial state is q_0 and the final states are q_6 and q_7 . Every sentence accepted by this automaton and ending in state q_7 is labelled with +1 (positive sentiment), whereas when ending in state q_6 , it is labelled with −1 (negative sentiment). The automaton scans the category of words in a given sentence and remains in state q_0 (initial state) until a verb or a negation has been found. Each transition from one state to another is labelled with the marker of the identified word category that triggers the change of state (in a few transitions, the symbol ϵ is used to identify exiting from the origin state in case any category is present rather than the one identifying transiting toward another state from the same origin state).

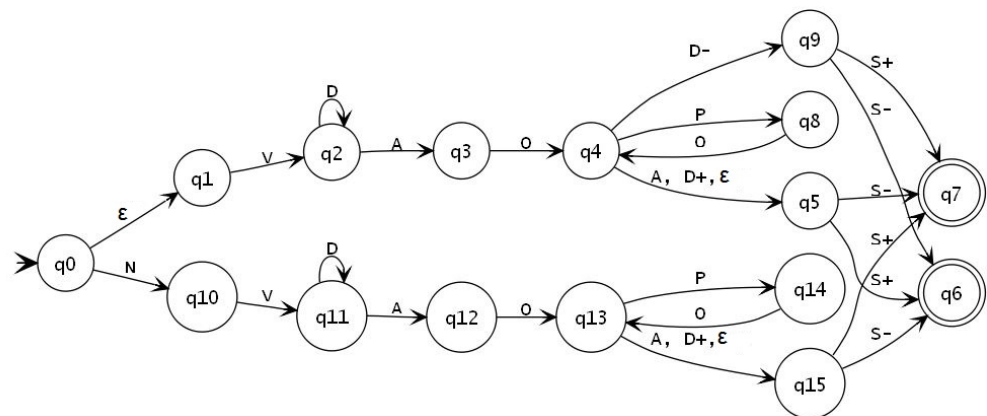


Figure 2. Automaton 1 recognising sentiments for sentences having form Verb + Article + Other + Sentiment, i.e., sequence VAOS and some variations (ϵ indicates anything else rather than the letter for other outgoing transitions from the origin state).

One example of sentences having such a form is: ‘L’elefante è un animale molto bello’ (The elephant is a very beautiful animal). According to our analysis, each word is mapped into a category (see Tables 1 and 2); hence, for the above sentence, we obtain the following sequence (of categories) **AOVAODS**. For this sentence, the automaton successfully recognised the sequence VAOS within the longer one given as input even though the VAO part is followed by DS and not simply by S. We marked in bold the meaningful categories of words in the sequence that present more Other word categories. Therefore, the automaton can recognise a whole set of variations for the initial grammar-correct sentence that we have analysed and coded in the automaton.

Another example for such a form of sentences is: ‘Il telefono ha uno schermo davvero molto resistente’ (The phone has a very durable screen), which would give the sequence of word categories **AOVAODDS**. Again, this sequence is successfully recognised by the automaton.

3.3.2. Automaton 2

This automaton handles sentences wherein the sentiment is expressed before the name to which it refers, i.e., the name can be mentioned after the sentiment or can be missing. Moreover, no words expressing the names can be in between the verb and the sentiment. Therefore, Automaton 2 successfully recognises the sentences having the form Verb* + Sentiment + Other*, which is the sequence V^*SO^* , and where * indicates that it is optional.

Figure 3 shows the automaton handling such a sequence. As for the previous automaton, this one scans categories of words for a given sentence and remains in state q_0 (initial

state) until a Verb or Negation (not) has been found. Then, the following words can trigger a transition towards the next states. The final states are q4 for negative sentiment and q7 for positive sentiment.

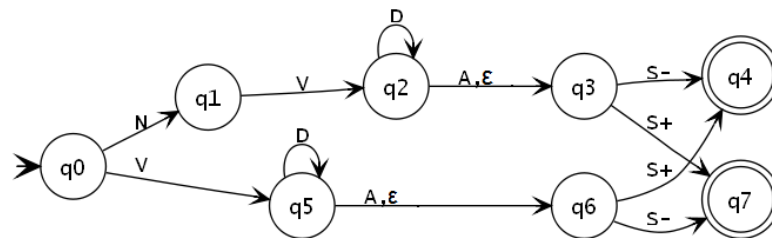


Figure 3. Automaton 2 recognising sentiments for sentences having form Verb + Sentiment + Other, i.e., sequence VSO and its variations.

E.g., the sentence ‘La tua moto ha un bel colore’ (Your motorbike has a beautiful colour) gives sequence AOOVASO, which is recognised by Automaton 2 as having a positive orientation. On the contrary, the sentence ‘La tua moto ha un colore bello’ (Your motorbike has a beautiful colour) gives the sequence AOOVAOS, which is recognised by Automaton 1 as having a positive orientation.

Let us take another sentence, e.g., ‘Il film non è stato molto interessante’ (The film was not very interesting), which, when processed, gives sequence AONVDS. This is a variation of the previous VSO sequence and is accepted by Automaton 2 thanks to the design allowing transitions, since D is a modifier (adverb) for the sentiment S (hence, VS or VDS are considered similar). Moreover, as said above, when O is missing, recognition can be performed anyway.

3.3.3. Automaton 3

The third automaton we considered is the one that handles the phrases that use superlatives, as they express an opinion (differently than sentences using comparatives, wherein the user is not really expressing his/her opinion but only making a comparison). It represents all sentences having the form: Verb + Article/Pronoun + Superlative + Sentiment, which is a sequence VAXS or a sequence VYXS, and their variations according to the transitions controlled by the automaton, which have been carefully designed.

Figure 4 shows that the automaton that accepts the said sequence form has final states q8 and q15 for positive and negative orientations, respectively. As with previous automata, categories of words are scanned, and if labelled transitions can be triggered, an exit state is reached until one final state or the end of the sequence is reached.

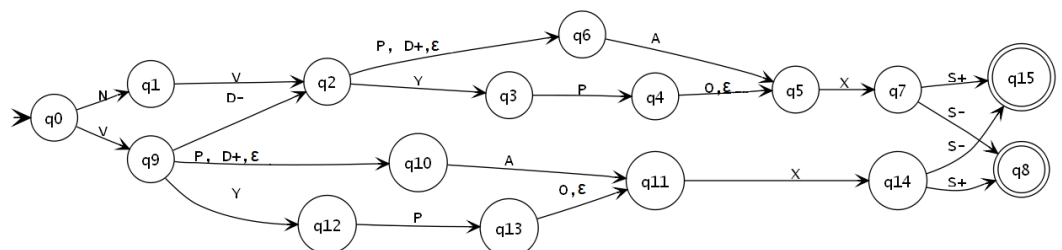


Figure 4. Automaton 3 recognising sentiments for sentences having form Verb + Article/Pronoun + Superlative + Sentiment, i.e., sequences VAXS or VYXS and their variations.

Given the sentence ‘L’auto è la più bella del mondo’ (The car is the most beautiful in the world), the resulting sequence AOVAXS would be recognised and accepted by Automaton 3 as having a positive orientation.

3.3.4. Sequence Generation from Automata

A list of possible grammar-legal variations to the form of a sentence accepted by the above automata can be generated by following the possible paths from the initial state to the end states for each of the described automata. Given that some transitions are labelled as anything else (ϵ) and that some transitions allow the formation of loops, the possible sequences can theoretically be infinite. We have thus limited the generation and considered: (i) a few cycles for each transition being a loop and (ii) all possible variations in case of (ϵ). We have obtained more than one hundred sequences for the automata and observed in the experiments that such sequences sufficed. Table 4 shows a partial list of accepted sequences.

Table 4. Sequences generated from the three proposed automata (V is for verb, A for article, O for other, S for sentiment, D for adverb, N for negation, X for superlative and P for pronoun).

Automaton 1	Automaton 2	Automaton 3
VAOS	VDAS	VYPOXS
NVAOS	NVDAS	NVYPOXS
VAOAS	VASO	VYPXS
NVAOAS	NVASO	NVYPXS
VDAOS	VDASO	VPAXS
NVDAOS	NVDASO	NVPAXS
VAODS	VDS	VAXS
NVAODS	NVDS	NVAXS
VAOPOAS	VDDAS	VDAXS
NVAOPOAS	VDADS	NVDAXS
VAOPOS		
NVAOPOS		

4. Experiments and Results

4.1. Dataset Selection

We have considered the following two collections of data. Data were gathered by selecting as targets restaurants, hotels and mobile phones in both data collections.

- Collection T1: a set of 921 sentences that have been randomly extracted from various web forums, such as Tripadvisor ([tripadvisor.com](https://www.tripadvisor.com), last accessed on 4 October 2023). After gathering the sentences, each sentence was manually labelled with a score that identifies the expressed sentiment: the score -1 identifies a negative sentiment; the score 0 identifies a neutral sentiment; the score $+1$ identifies a positive sentiment. Table 5 shows some examples of the analysed sentences, the sequences of categories of words automatically determined, the automaton (1 to 3) that reached the final state, and the final state of the automaton (q6 or q7 for Automaton 1, q4 or q7 for Automaton 2, or q8 for Automaton 3).
- Collection T2: a set of 780 sentences that have been extracted from the Yelp platform. We collected data using Yelp's API (<https://www.yelp.com/developers>, last accessed on 4 October 2023). Yelp is a review forum that allows users to post their comments on local restaurants and to associate a number of stars ranging from 1 to 5 to give a rating representing the user's satisfaction; of course a low number of stars corresponds to a poor rating (and vice versa). We assume that for each user comment, the number of stars is a measure for the sentiment expressed in the review—that is: if the number of stars is less than three, then it is considered a negative sentiment; if the number of stars is equal to three then it is considered as neutral sentiment; if the number of stars is greater than three, then it is considered as positive sentiment. Table 6 shows some analysed sentences, the number of stars and the resulting score.

Table 5. A sample of sentences extracted from web forums and analysed using our approach. Column ‘Sentence’ gives the extracted text; column ‘Pattern’ shows the list of categories for the words in the sentence; column ‘Result’ shows the sentiment identified by the proposed algorithm (positive if score is greater than zero, neutral if score is equal of zero, and negative if score is less than zero), the final state and the number of the automaton that accepted the sentence.

Sentence	Pattern	Result (a, st)
Huawei P50 è stato lo smartphone migliore del 2021 (Huawei P50 was the best smartphone for 2021)	OOVAOS+PO	positive (1, q6)
Lo Zenfone 9 ha un prezzo molto competitivo (Zenfone 9 is inexpensive)	AOOVAOD+S+	negative (1, q6)
Lo Zenfone 9 ha un prezzo poco competitivo (Zenfone 9 is expensive)	AOOVAOD-S+	negative (1, q7)
OnePlus 8 ha una versione di Android molto leggera (OnePlus 8 has a very light Android version)	OVAOPD+S+	positive (1, q6)
OnePlus 8 ha una versione di Android molto pesante (OnePlus 8 has a very heavy Android version)	OVAOPD+S-	negative (1, q7)
L’hotel Verdi non ha un aspetto grazioso (Verdi hotel does not look pretty)	AOONVAOS+	negative (1, q7)
L’iPhone è un bel telefono (iPhone is a beautiful phone)	AOVASO	positive (2, q4)
La telecamera dell’iPhone non è per nulla competitiva (iPhone camera is by no means competitive)	AODONVD-D-S+	negative (2, q7)
L’iPhone è un ottimo dispositivo sia dal punto di vista hardware che software (iPhone is an excellent device both from a hardware and software point of view)	AOVASO	positive (2, q4)
L’iPhone è davvero molto bello (iPhone is really beautiful)	AOVD+D+S+	positive (2, q4)
La camera dell’hotel Verdi è la più bella del mondo (Verdi hotel’s room is the most beautiful in the world)	AODOOVAXS	positive (3, q8)
L’iPhone è uno dei più potenti (iPhone is one of the most powerful)	AOVYPXP	positive (3, q8)

Table 6. A sample of sentences taken from Yelp and used for SA. Column ‘Post’ shows the extracted sentences; column ‘Stars’ shows the number of stars given in the post; column ‘Score’ gives the score given by our approach (both results, 4 stars and score 1 and 1 star and score −1, are true positives).

Post	Stars	Score
Un locale davvero bello. Perfetto per le cene romantiche. L’ambiente è raffinato, ma i proprietari sanno metterti a tuo agio. Cibo ottimo e presentato in. (A really nice place. Perfect for romantic dinners. The environment is refined, but the owners know how to put you at ease. Great food and presented in.)	4	+1
Esperienza pessima. Scegliamo questo ristorante per la notte di capodanno. Prezzo: 80 a persona, una cifra non bassa e che aveva creato in noi aspettative. (Bad experience. We choose this restaurant for New Year’s Eve. Price: 80 per person, a non-low figure that had created expectations in us.)	1	−1

Sentences for collection T1 were each associated (manually) with an orientation (positive, neutral or negative) only for the sake of the validation step (no other manual analysis was performed). For collection T2, the orientation in each sentence was automatically determined from the number of stars. The orientation values for sentences formed a ground truth, which, in our tests, was used to compute precision (the ratio between relevant points and retrieved points) and recall (the fraction of relevant points retrieved) as detailed in Section 4.3. The tests (and comparison) were performed using our proposed approach and Serendio algorithm (see description in Section 4.2).

4.2. Serendio Algorithm

Serendio is a practical lexicon-based approach to SA that extracts and analyses sentiments for products [12]. The Serendio approach created a lexicon using the following categories: common or default sentiment words (positive and negative sentiment words that have the same sentiment value in different domains, i.e., good), negation words (words that reverse the polarity of sentiment, i.e., not), blind negation words (words that point out the absence or presence of some sense that is not desired in a product feature, i.e., need), and split words (words that are used for splitting sentences into clauses, i.e., but).

The Serendio algorithm is based on a sentiment calculation (SentiSum), which is the aggregation of the sum of the sentiment-bearing emotions of the sentence. The algorithm assigns to the analysed sentence a SentiSum as follows: words in the sentence are filtered according to the sentiment list (including positive and negative sentiments); then, sentiment polarity is changed if the word occurs in proximity of a negative word (two-word distance). E.g., ‘The restaurant is not good’, the sentiment word *good* becomes a negative word. Therefore, for each word in the sentiment list (common or default sentiment word) that is in the sentence, if the word is a positive sentiment, then +1 is added to the SentiSum; if the word is a negative sentiment, then −1 is added to the SentiSum. In addition, the presence of blind negation words indicates a negation of the sentiment. Finally, if SentiSum is greater than 0, then the sentence is tagged as having positive sentiment; if SentiSum is equal to 0, then the sentence is tagged as neutral; otherwise, if SentiSum is less than 0, then the sentence is negative.

For the comparison with our approach, we have implemented the Serendio algorithm in Java and given the same lexicon as for our approach.

4.3. Results and Comparison

The above proposed automata embedding grammar rules for finding the sentiment of a sentence (see Section 3) guided the development of our tool for processing text and determining the sentiment of sentences.

For the validation of our approach, the tool we developed was given the sentences to be analysed and, as a result, obtained the sentiment for each sentence; then, this was compared with the label score of the datasets (collections T1 and T2). Finally, the numbers of true positives, true negatives, false positives and false negatives were computed. If the label in the data and the automata expressed the same sentiment (negative, neutral or positive), we considered the result a true positive (TP). If our approach identified an orientation when the sentence was neutral, then this was considered a false positive (FP); whereas when our approach identified a sentence as neutral when the orientation was positive or negative, then this was counted a false negative (FN). When the identified sentiment orientation (e.g., positive) was opposite that of the label (e.g., negative), then this was counted as a false positive (FP). Subsequently, precision and recall were computed.

Table 7 gives the numbers of true positives (TPs), false positives (FPs) and false negatives (FNs) and precision, recall and F-score for our approach and Serendio for each dataset (collections T1 and T2). Precision was computed as $\frac{TP}{TP+FP}$, and recall was computed as $\frac{TP}{TP+FN}$. F-score was computed as $\frac{2}{1/r+1/p}$ and gives a measure of a test’s accuracy that combines precision (p) and recall (r).

Table 7. Overview of the results and comparison of the proposed approach and Serendio.

Source	Algorithm	TP	FP	FN	Precision	Recall	F-Score
T1 Forum	Our approach	628	16	182	0.975	0.775	0.864
	Serendio	302	230	26	0.568	0.921	0.702
T2 Yelp	Our approach	405	90	216	0.818	0.652	0.726
	Serendio	402	92	217	0.813	0.649	0.722

For collection T1, our SA approach detected the sentiment for 90% of the sentences given as input (the other 10% were unclassified) and with a high degree of precision (0.975). Some low rates of recall were due to the limited list of words in our dictionary, which can be extended to reflect a larger variety of sentiments, as well as the presence of very short user comments that do not use any grammar while expressing an opinion.

Table 8 shows the number of analysed sentences grouped by orientation (positive, neutral and negative). Column ‘Unclassified’ gives the number of sentences that were not recognised by any automata and for which an orientation was not assigned; column ‘TP’ gives the number of correctly identified orientations, and columns ‘FP’ and ‘FN’ give the mistaken identifications (counted as described above). The last row gives the number of unclassified, correct identification (TP) and mistaken identification overall for the T1 collection of sentences.

Table 8. Detailed results for collection T1 analysed by the proposed approach.

Orientation	Total	Unclass'd	TP	FP	FN	Precision	Recall	F-Score
Positive	504	53	292	0	159	1.000	0.647	0.786
Neutral	324	38	276	10	0	0.965	1.000	0.982
Negative	93	4	60	6	23	0.909	0.723	0.805
T1	921	95	628	16	182	0.975	0.775	0.864

Figure 5 shows the comparison between the results when analysing collections T1 and T2 with both our approach and Serendio. It can be seen that our approach gave better precision. This can be explained by the use of our proposed careful text analysis that allows some degree of text comprehension according to the programmed grammar rules. We successfully identified more sentiment orientations than Serendio, as we can see when comparing the number of true positives and false positive for the two approaches. Although Serendio has simple rules to determine orientation, for collection T1 having 921 sentences, 363 were left unclassified and only 302 were correctly labelled. As a comparison, with our approach, only 95 were unclassified and 628 were correctly labelled.

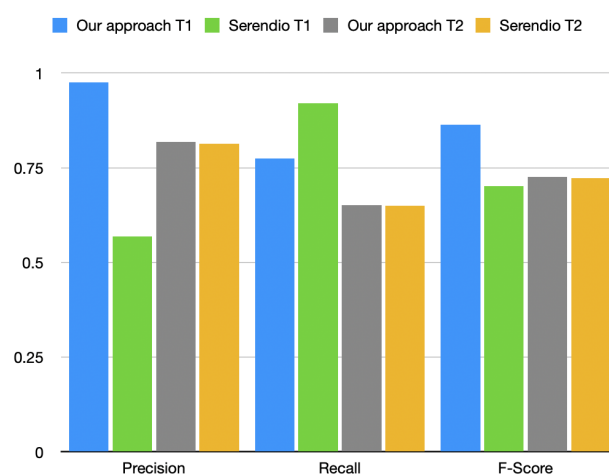


Figure 5. Results in terms of precision, recall and F-score obtained when processing sentences in T1 and T2 collections.

5. Parallelization Approach

Given the huge amount of text sentences that are available and that are generated on the web continuously, we have implemented a parallel version of the proposed algorithm using the Java Stream API, available from version 8. The Java Stream type provides significant support inspired by the MapReduce paradigm [13] and introduces handy support for parallelism for properly using modern multicore architectures. Stream supports functional-style operations on elements, typically initially held in a Collection, such as

map(), filter(), reduce(), etc. Such operations are executed in parallel by leveraging the available multicore availability of the underlying hardware and without requiring specific effort from the developer for the aim of parallel execution except from the call to operation parallel().

Algorithm 1 shows the algorithm that splits the text in sentences (line 2) then splits each sentence into a list of words while associating each word with a key (k) identifying the position of the word (line 5). Furthermore, the word is transformed into an identifier for its category (line 8); then, all the identifiers for the categories in a sentence are connected in the proper order (line 9). Finally, the sequence of identifiers is matched against known sequences representing the automata, and the final state of automata is revealed (line 10).

Algorithm 1 Algorithm for finding the sentiment (positive or negative) in a text

```

1: procedure SENTIMENTANALYSIS(text)
2:   sentenceList ← splitSentences(text)
3:   sentenceList.stream()
4:     .parallel()
5:     .map(sentence → splitWords(sentence))
6:     .stream()
7:     .parallel()
8:     .map(word → replace(word))
9:     .reduce("", (m, v) → concatenate(m, v))
10:    .forEach(seq → checkPattern(seq))

```

The above algorithm calls the following methods.

- splitWords(): given a text, splits it into its constituting words and associates with each word its position in the sentence;
- replace(): replaces a word with the corresponding category, such as article, sentiment, adverb, etc. (see Tables 1 and 2);
- concatenate(): joins the letters of categories together according to their initial positions in the original sentence;
- checkPattern(): takes a sequence of identifiers for categories and computes a score according to the automaton recognising it (as discussed in Section 3).

Table 9 shows the times in seconds for sequential and parallel executions for several millions (from 11 to 26) of sentences processed. The speed-up (computed as sequential/parallel) was up to 2.127. Parallel execution was performed by launching two threads by means of the Java Stream library. The hardware running the tool was equipped with four cores, 16 GB RAM and a i7-1185G7 CPU@3.00 GHz.

Table 9. Execution times in seconds for the sequential and parallel versions using two threads for several millions of sentences analysed.

n. Sentences (Millions)	Sequential Time (s)	Parallel Time (s)	Speed-Up
11	280	150	1.866
16	405	233	1.740
19	478	270	1.770
22	657	350	1.878
26	836	393	2.127

6. Discussion

A characteristic of our proposed approach is that it can be easily extended to reveal sentiment orientation when additional forms of sentences are revealed. A proper automaton that expresses some further grammar rules can be designed and coded in the format expressed above, i.e., a list of sequences of categories of words to be recognised. Then, the

list of sequences can be given to the same algorithm, e.g., using a configuration file, and this will check whether a sequence is accepted and a final state will be reached.

For our approach, a lexicon consisting of a list of words for each category has been prepared. Such lists are then used to recognise whether a word expresses a sentiment or is a relevant verb, an article, etc. The used lexicon has a limited number of words, i.e., a few dozen for sentiments, verbs, etc. By further expanding the lists to consider a more-comprehensive classification of words—some common variations of words (e.g., expressing emphasis, ‘urban’ terms, common misspelling, etc.) and widely used words that are imported from English or French and are used in informal Italian sentences—then the ability of automata to recognise the sentences will be improved, and in turn, precision and recall will be higher.

Moreover, our approach can be used when dealing with languages that have a similar structure to Italian, such as Latin-based languages and English. For each language, a set of rules can be represented by a corresponding automaton; then from this, we can enumerate all the legal sequences, and finally, the sequences are given to the same algorithm that finds whether a final state can be reached for the analysed sentence.

Our approach is very lightweight in terms of data needed and execution time when comparing it with typical ML approaches. The initial data needed in our approach are the lists of words for each category to be identified, which are constant for a given natural language. No training is needed for our algorithm to use such lists. Execution time when analysing a sentence is very short, as simple operations (e.g., string substitution and comparison) are performed, which are prone to being performed in parallel. Given the low amount of initial data needed and the performed computation, the developed tool could even be embedded in a mobile platform (having a limited amount of computational resources) to filter sentences, e.g., coming from social networks, according to topics and to avoid more-fragile people having contact with offensive or negative sentences.

7. Conclusions

In our work, we have proposed a solution that performs sentiment analysis on text using a very lightweight approach that can achieve both precision and short execution time. The proposed approach finds sentiment orientation in a sentence by using Italian grammar rules and a lexicon. We have analysed sentences and represented the underlying grammar by means of three proposed finite-state automata. These three automata resulted in being general enough and adequate to successfully recognise the orientation of several thousands of sentences extracted from web sites.

The proposed approach is both robust and accurate. From the analysis of thousand of sentences, we had: a very low number of false positives (lower than other approaches), very high precision (higher than other approaches) and a higher number of true positive. Comparisons were made with the Serendio algorithm, which is considered among the state-of-the-art in this field. Moreover, the approach was adequate to properly process reinforcing or denying expressions, such as ‘poco attraente’ (not very attractive) and ‘tanto attraente’ (very attractive), as well as many variations of sentences that can be obtained when duplicating adjective or adverbs (to reinforce), when changing the order of subject and sentiment, etc. We could also successfully attain the correct association between the use of a negation and the word of the sentiment it refers to, achieving the correct result when identifying the orientation of the sentence.

The tool corresponding to our approach is very lightweight as it need not have training and only has a small lexicon; hence, execution time is very low. By not relying on a large annotated dataset, our approach can be very useful for sentences using languages for which such datasets are unavailable: hence, for which a machine learning approach would be infeasible or very expensive.

The proposed approach can be expanded for analysing sentences written in English or other languages having a structure resembling Italian. The required work would be to derive the grammar rules describing how some typical sentences expressing opinions

can be legally formed and give a lexicon consisting of words for each category that we have identified. For grammar rules to be coded, it just suffices to enumerate the legally accepted sequences for word categories. Then, giving such data to the same tool we have developed, the tool would find the potential match between such sequences and the analysed sentences.

Author Contributions: Conceptualization, E.T. and G.V.; methodology, E.T. and G.V.; software, G.V.; validation, A.C., E.T. and G.V.; writing—original draft preparation, A.C., E.T. and G.V.; writing—review and editing, A.C. and E.T.; supervision, A.C. and E.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Restrictions apply to the availability of data, as they were obtained from third parties.

Acknowledgments: The authors acknowledge the support of the University of Catania PIACERI 2020/22 Project “TEAMS”.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Feldman, R. Techniques and applications for sentiment analysis. *Commun. ACM* **2013**, *56*, 82–89. [\[CrossRef\]](#)
2. Pang, B.; Lee, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135. [\[CrossRef\]](#)
3. Abbasi, A.; Chen, H.; Salem, A. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.* **2008**, *26*, 12. [\[CrossRef\]](#)
4. Beineke, P.; Hastie, T.; Manning, C.; Vaithyanathan, S. An exploration of sentiment summarization. In Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications, Palo Alto, CA, USA, 22–24 March 2004; Volume 3, pp. 12–15.
5. Wiebe, J.M.; Bruce, R.F.; O'Hara, T.P. Development and use of a gold-standard data set for subjectivity classifications. In Proceedings of the Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, MD, USA, 20–26 June 1999; pp. 246–253.
6. Liu, B. Sentiment analysis and opinion mining. *Synth. Lect. Hum. Lang. Technol.* **2012**, *5*, 1–167.
7. Naik, M.V.; Vasumathi, D.; Siva Kumar, A. A Novel Approach for Extraction of Distinguishing Emotions for Semantic Granularity Level Sentiment Analysis in Multilingual Context. *Recent Adv. Comput. Sci. Commun.* **2022**, *15*, 77–87. [\[CrossRef\]](#)
8. Mercha, E.M.; Benbrahim, H. Machine learning and deep learning for sentiment analysis across languages: A survey. *Neurocomputing* **2023**, *531*, 195–216. [\[CrossRef\]](#)
9. Catelli, R.; Pelosi, S.; Esposito, M. Lexicon-based vs. Bert-based sentiment analysis: A comparative study in Italian. *Electronics* **2022**, *11*, 374. [\[CrossRef\]](#)
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [\[CrossRef\]](#)
11. Meetei, L.S.; Singh, T.D.; Borgohain, S.K.; Bandyopadhyay, S. Low resource language specific pre-processing and features for sentiment analysis task. *Lang. Resour. Eval.* **2021**, *55*, 947–969. [\[CrossRef\]](#)
12. Palanisamy, P.; Yadav, V.; Elchuri, H. Serendio: Simple and Practical lexicon based approach to Sentiment Analysis. In Proceedings of the Second Joint Conference on Lexical and Computational Semantics, Atlanta, GA, USA, 14–15 June 2013; pp. 543–548.
13. Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters. *Commun. ACM* **2008**, *51*, 107–113. [\[CrossRef\]](#)
14. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Eng. J.* **2014**, *5*, 1093–1113. [\[CrossRef\]](#)
15. Birjali, M.; Kasri, M.; Beni-Hssane, A. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowl.-Based Syst.* **2021**, *226*, 107134. [\[CrossRef\]](#)
16. Tan, K.L.; Lee, C.P.; Lim, K.M. A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research. *Appl. Sci.* **2023**, *13*, 4550. [\[CrossRef\]](#)
17. Fatima, N.; Imran, A.S.; Kastrati, Z.; Daudpota, S.M.; Soomro, A. A Systematic Literature Review on Text Generation Using Deep Neural Network Models. *IEEE Access* **2022**, *10*, 53490–53503. [\[CrossRef\]](#)
18. Cobo, M.; Perez, I.; Velez-Estevez, A.; Cabrerizo, F. Uncovering the conceptual evolution of sentiment analysis research field during the period 2017–2021. In Proceedings of the 2022 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS), Larnaca, Cyprus, 25–26 May 2022; Volume 2022. [\[CrossRef\]](#)
19. Miller, G.A.; Beckwith, R.; Fellbaum, C.; Gross, D.; Miller, K.J. Introduction to WordNet: An on-line lexical database. *Int. J. Lexicogr.* **1990**, *3*, 235–244. [\[CrossRef\]](#)

20. Kamps, J.; Marx, M.; Mokken, R.J.; De Rijke, M. Using WordNet to Measure Semantic Orientations of Adjectives. In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal, 26–28 May 2004; Volume 4, pp. 1115–1118.
21. Zhu, H. Sentiment Analysis of 2021 Canadian Election Tweets. In Proceedings of the International Conference on Artificial Intelligence, Virtual Reality, and Visualization (AIVRV 2022), Chongqing, China, 23–25 September 2022; Volume 12588. [\[CrossRef\]](#)
22. Nalini, C.; Dharani, B.; Baskar, T.; Shanthakumari, R. Review on Sentiment Analysis Using Supervised Machine Learning Techniques. *Lect. Notes Netw. Syst.* **2023**, *715*, 166–177. [\[CrossRef\]](#)
23. Ye, Q.; Zhang, Z.; Law, R. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Syst. Appl.* **2009**, *36*, 6527–6535. [\[CrossRef\]](#)
24. Go, A.; Bhayani, R.; Huang, L. Twitter sentiment classification using distant supervision. *CS224N Proj. Rep. Stanf.* **2009**, *1*, 2009.
25. Devi, J.S.; Nandyala, S.P.; Reddy, P.V.B. A Novel Approach for Sentiment Analysis of Public Posts. In *Innovations in Computer Science and Engineering*; Springer: Singapore, 2019; pp. 161–167.
26. Buntoro, G.A. Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter. *INTEGER J. Inf. Technol.* **2017**, *2*, 1. [\[CrossRef\]](#)
27. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Kerrville, TX, USA, 2014; pp. 1746–1751. [\[CrossRef\]](#)
28. Jian, Z.; Chen, X.; Wang, H.S. Sentiment classification using the theory of ANNs. *J. China Univ. Posts Telecommun.* **2010**, *17*, 58–62.
29. Yadav, V.; Verma, P.; Katiyar, V. Long short term memory (LSTM) model for sentiment analysis in social data for e-commerce products reviews in Hindi languages. *Int. J. Inf. Technol.* **2023**, *15*, 759–772. [\[CrossRef\]](#)
30. Ko, Y.; Seo, J. Automatic text categorization by unsupervised learning. In Proceedings of the 18th Conference on Computational Linguistics—Volume 1 (COLING'00), Saarbrücken, Germany, 31 July 2000; Association for Computational Linguistics (ACL): Kerrville, TX, USA, 2000; pp. 453–459.
31. Suhaeni, C.; Yong, H.S. Mitigating Class Imbalance in Sentiment Analysis through GPT-3-Generated Synthetic Sentences. *Appl. Sci.* **2023**, *13*, 9766. [\[CrossRef\]](#)
32. Jain, K.; Ghosh, P.; Gupta, S. A Hybrid Model for Sentiment Analysis Based on Movie Review Datasets. *Int. J. Recent Innov. Trends Comput. Commun.* **2023**, *11*, 424–431. [\[CrossRef\]](#)
33. Palomino, M.A.; Varma, A.P.; Bedala, G.K.; Connelly, A. Investigating the Lack of Consensus Among Sentiment Analysis Tools. *Lect. Notes Artif. Intell.* **2020**, *12598*, 58–72. [\[CrossRef\]](#)
34. Prabowo, R.; Thelwall, M. Sentiment analysis: A combined approach. *J. Inf.* **2009**, *3*, 143–157. [\[CrossRef\]](#)
35. Mudinas, A.; Zhang, D.; Levene, M. Combining lexicon and learning based approaches for concept-level sentiment analysis. In Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining, Beijing China, 12 August 2012; ACM: New York, NY, USA, 2012; p. 5.
36. El-Haj, M.; Kruschwitz, U.; Fox, C. Creating language resources for under-resourced languages: Methodologies, and experiments with Arab. *Lang. Resour. Eval.* **2015**, *3*, 549–580. [\[CrossRef\]](#)
37. Le, T.A.; Moeljadi, D.; Miura, Y.; Ohkuma, T. Sentiment analysis for low resource languages: A study on informal Indonesian tweets. In Proceedings of the 12th Workshop on Asian Language Resources (ALR12), Osaka, Japan, 12 December 2016; pp. 123–131.
38. Rajabi, Z.; Valavi, M. A Survey on Sentiment Analysis in Persian: A Comprehensive System Perspective Covering Challenges and Advances in Resources and Methods. *Cogn. Comput.* **2021**, *13*, 882–902. [\[CrossRef\]](#)
39. Gangula, R.R.R.; Mamidi, R. Resource Creation Towards Automated Sentiment Analysis in Telugu (a low resource language) and Integrating Multiple Domain Sources to Enhance Sentiment Prediction. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; European Language Resources Association (ELRA): Paris, France, 2018.
40. Nasib, A.U.; Kabir, H.; Ahmed, R.; Uddin, J. A Real Time Speech to Text Conversion Technique for Bengali Language. In Proceedings of the International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Rajshahi, Bangladesh, 8–9 February 2018; pp. 1–4. [\[CrossRef\]](#)
41. Altaf, A.; Anwar, M.W.; Jamal, M.H.; Bajwa, U.I. Exploiting Linguistic Features for Effective Sentence-Level Sentiment Analysis in Urdu Language. *Multimed. Tools. Appl.* **2023**, *82*, 41813–41839. [\[CrossRef\]](#)
42. Chauhan, G.S.; Nahta, R.; Meena, Y.K.; Gopalani, D. Aspect based sentiment analysis using deep learning approaches: A survey. *Comput. Sci. Rev.* **2023**, *49*, 100576. [\[CrossRef\]](#)
43. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F. Sentiment analysis based on deep learning: A comparative study. *Electronics* **2020**, *9*, 483. [\[CrossRef\]](#)
44. Prottasha, N.J.; Sami, A.A.; Kowsher, M.; Murad, S.A.; Bairagi, A.K.; Masud, M.; Baz, M. Transfer learning for sentiment analysis using BERT based supervised fine-tuning. *Sensors* **2022**, *22*, 4157. [\[CrossRef\]](#) [\[PubMed\]](#)
45. Singh, C.; Imam, T.; Wibowo, S.; Grandhi, S. A deep learning approach for sentiment analysis of COVID-19 reviews. *Appl. Sci.* **2022**, *12*, 3709. [\[CrossRef\]](#)
46. Zhong, Q.; Ding, L.; Liu, J.; Du, B.; Tao, D. Can ChatGPT Understand Too? A Comparative Study on ChatGPT and Fine-tuned BERT. *arXiv* **2023**, arXiv:2302.10198.

47. Ikeda, D.; Takamura, H.; Ratinov, L.A.; Okumura, M. Learning to shift the polarity of words for sentiment classification. In Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I, Asian Federation of Natural Language Processing, Hyderabad, India, 7–12 January 2008.
48. Jia, L.; Yu, C.; Meng, W. The effect of negation on sentiment analysis and retrieval effectiveness. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, China, 2–6 November 2009; ACM: New York, NY, USA, 2009; pp. 1827–1830.
49. Morante, R.; Schrauwen, S.; Daelemans, W. Corpus-based approaches to processing the scope of negation cues: An evaluation of the state of the art. In Proceedings of the Ninth International Conference on Computational Semantics, Oxford, UK, 12–14 January 2011; Association for Computational Linguistics: Kerrville, TX, USA, 2011; pp. 350–354.
50. Liu, B. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing*, 2nd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2010; pp. 627–666.
51. Jindal, N.; Liu, B. Identifying comparative sentences in text documents. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, 6–11 August 2006; ACM: New York, NY, USA, 2006; pp. 244–251.
52. Nasukawa, T.; Yi, J. Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the International Conference on Knowledge Capture, Sanibel Island, FL, USA, 23–25 October 2003; ACM: New York, NY, USA, 2003; pp. 70–77.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.