Using Complex Deep Learning Neural Architectures for Sentiment Mining and Analysis across Text Corpora

Enzo, Nico
University of Brescia, Brescia, Italy

Abstract: In order to address the issue of irony and sarcasm identification for the Italian language, we examine and contrast five deep learning neural architectures. To determine the optimal trade-off between complexity and performance, we quickly examine the model architectures. The reported findings, which in the best scenario achieved 93% of the F1-Score, demonstrate how well such systems handle the challenge. As a case study, we also demonstrate how neural systems might be integrated into a cloud computing infrastructure to take advantage of the computational benefits of doing so when dealing with large data.
Keywords: Sentiment mining; Sentiment analysis; Text corpora; Deep learning; Neural architectures; Complex architectures.

1. Introduction

A bigger and more significant role for sentiment analysis has been performed by Web 2.0 and social media development. The opinions shared via reviews, comments, feedback, and other forms of communication can serve as useful benchmarks for a variety of applications, from political concerns to marketing.

The study of sentiment analysis has been extensively researched in the past. Its primary goal is to analyze the thoughts, feelings, attitudes, and emotions that people express through written words. In actuality, users can share their thoughts on a wide range of subjects, offer recommendations, and convey favorable or unfavorable feelings about everyday items via microblogs (like Twitter). As the number of active web users has increased, sentiment analysis has become more relevant since writing down one's ideas enables one to investigate the general consensus on a variety of topics, including politics and advertisements. Fake news (e.g., [8]) that can appear in such posts is another pertinent issue in this context. Even while this issue is intriguing right now, it is not at the heart of our research and will be addressed in the future.

Generally speaking, the sentiment can be separated into good and negative categories. According to this viewpoint, it can be viewed as a classification task with three possible values: neutral, negative, and positive.

Researchers have been concentrating on creating effective methods to automate this work because the vast volume of data has rendered manual text analysis impracticable. When linguistic strategies like irony and sarcasm conceal the text's true meaning, basic methods that analyze the text literally encounter problems. "Saying the opposite of what you mean" is the traditional definition of such figures of speech [7]. There is a slight distinction between sarcasm and irony because the latter is typically employed to make fun of or degrade someone; hence, it might be seen as a subclass of irony.

Two intricate linguistic phenomena that occur in daily interactions are irony and sarcasm [16]. Irony and sarcasm are rather similar, however irony lacks a target, whereas sarcasm has a definite target who is the object of the sarcastic statement [25].

However, from a computational standpoint, they are challenging to address for automatic text comprehension and human-machine interaction.
Irony and sarcasm detection is influenced by the situation as well as the speaker's mental state, including attitude, knowledge, and intentions. These factors should be considered in order to comprehend sarcastic content in a sentence [29] and may help to increase the efficacy of conversational agents that possess associative and intuitive abilities [38]. Furthermore, even people find it more challenging to detect sarcasm in written texts [39] because written texts lack the tone of speech that typically conveys sarcastic intent.
Because it impacts the functionality of systems that provide sentiment analysis, automatically identifying irony and sarcasm has thus become a crucial task to address.

In this work, we use Deep Learning (DL) techniques to address the irony-sarcasm identification problem. Because there are, as far as we are aware, very few publications on the subject, our study is concentrated on the Italian language. By merging designs like Recurrent Neural Network (RNN), Bi-direction RNN, Convolutional Neural Network (CNN), and Attention mechanism, we are able to compare a large number of DL systems. In order to assess the extent to which these components affect system performance, we first exploited a DL system [35] designed to handle the sarcasm detection task. Next, we modified the network architecture by removing layers. A corpus of ironic and non-ironic samples taken from Spinoza and seven well-known newspapers and Twitter accounts, respectively, is used to train the presented algorithms. In order to balance the dataset, the experiments are calculated by repeatedly utilizing the K-Fold cross-validation methodology. By obtaining 93% of the F1-Score in the best scenario, the results demonstrate the great adaptability of such systems to handle the challenge.

The structure of the paper is as follows. Section 1 presents a case study on a potential integration into a cloud computing infrastructure; Section 2 provides a brief overview of the relevant literature; Section 3 introduces the key components (pre-processing and the model) and parameters that define the systems that are being presented; Sections 4 and 5 each explain the methodology used to test our systems and the results analysis. Section 6 concludes with findings and future directions.

2. CASE STUDY: CLOUD COMPUTING INFRASTRUCTURES WITH FRAMEWORK IMPLEMENTATION

In the context of big data, the system can find a natural area of application. Building efficient, ad-hoc big data architectures to support sarcasm detection using a neural technique can be accomplished by integrating the system within the core layer of well-known big data processing platforms, such Hadoop, Spark, etc. A reference architecture designed to do this is shown in Figure 1. The following levels of the suggested architecture can be distinguished when mapping it as a conventional multi-layer software solution:

The big data sources that generate the input that is given to one or more instances of the suggested system are located in the data source layer. Document repositories, social networks, websites, and blogs are a few examples of data sources that provide actual application scenarios where the architecture can be used efficiently. Cloud-Based Prediction Layer: this layer makes use of the flexibility of contemporary cloud computing infrastructures and large data processing platforms by deploying classifier systems on top of a cloud environment. In order to impose distributed and parallel processing paradigms, a classifier instance is replicated on each Cloud node. Each node in this layer produces a graph, where each node is the chunk of text associated with the label "sarcastic" or "non-sarcastic." If a semantic similarity, as determined by a sub-symbolic score function, is greater than a predetermined experimentally set threshold, two nodes are connected by an edge.

The software layer known as the "Data Staging Layer" is where huge data repositories are staged using well-known and widely utilized big data processing platforms. It also provides the appropriate big data services to support the analytics architecture as a whole. At this layer, the underlying big data processing platform supports the next layer of the architecture where appropriate big data analytics tasks are delivered, in addition to supporting the collection of big data sources, proper prediction via the classifier system (thanks to a core integration with the Cloud computing infrastructure), and semantic similarity between text items; The software components that are specifically designed to facilitate big data analytics jobs over the irony/sarcasm prediction data produced by the corresponding layer are found in the Big Data Analytics Layer. Numerous options, such as visual big data analytics, can be investigated in this situation. By utilizing the cloud computing paradigm, the suggested cloud-based architecture enables the realization of more effective applications that may be used on very big datasets. The most important factor in doing this is figuring out how to make the categorization method "Cloud-able." The system can thus attain a better degree of scalability over massive data that populates real-life applications because of the Cloud support, which is an important element to consider.

Each source's content is filtered based on a predetermined semantic query, and a node with a particular neural network trained to identify irony or sarcasm in the examined text subsequently processes the textual data. It is possible to train, or update, each network separately from the other networks found in the other nodes. To create, for instance, a recommendation system or decision support system, the results of the categorization process are then sent and examined via cloud infrastructure.

3. RELATED WORK

The expansion of internet users has resulted in an enrichment of web textual materials due to the growing options for accessing technological facilities. The emergence of social media has made the billions of different types of writings that are produced on the Internet every day more accessible [17] [36]. By gathering web resources to create datasets, academics have examined the value of internet data in recent years. Such information has been utilized to build models that can complete various tasks, leading to systems that can perform well. [37] describes one of the earliest attempts to detect sarcasm using web data. Amazon reviews show how well automatically-crawled data can handle the work, and authors use data from Twitter to their advantage. Since hashtags can be used to filter messages with specific ironic or sarcastic content, the majority of the following strategies share the usage of the Twitter platform. Twitter lets users write up to 280 characters of text, which can be enhanced with hashtags to help focus on the message's main points (e.g., irony, sarcasm, critique, approval). Initial findings on sarcasm detection using Twitter data have demonstrated how challenging the problem is, even for humans. A corpus of 2700 tweets that have been evenly classified as sarcastic, positive, and negative emotion was constructed in [22]. By showing a portion of this corpus to a panel of human judges, it tested the capacity of humans to identify sarcasm. Low accuracy and low agreement were also evident in the results.

In the categorization task by demonstrating the task's intricacy. Additionally, in [30], a group of users who have not included the hashtag are shown a collection of tweets that are illustrative of a sarcastic circumstance (i.e., convey a positive emotion in a negative event) and tagged as #sarcastic. Another indication of the task's complexity was the discovery that human judges identified less than half of the tweets as sarcastic. According to a study on sarcasm detection effort in [32], documents find the task easier since context analysis yields additional information that makes the task simpler.

Several strategies have been used to address the automatic detection of sarcasm. A system based on hashtags and a set of guidelines for sarcasm recognition has been suggested in [28]. The experiments yielded a precision of 96.41% using a corpus of tweets with general topics. The effects of adding user features and an author description on system performance have been examined in [4]. By achieving a notable increase in accuracy, the experiments demonstrated a beneficial contribution to the augmented data. In [16], a Support Vector Machine (SVM) was used to classify sarcasm using Latent Semantic Analysis (LSA) and the Truncated Single Value Decomposition (T-SVD). The system demonstrated a precision of up to 71% when evaluated on five distinct corpora. As a subtask of sentiment analysis, which has been extensively researched in the past, the problem has been extensively studied. Three classifiers—SVM, multinomial Navies Bayes, and random forest—were used in [2] to construct an ML model for the sentiment analysis of football-related tweets. In [31], stock market prices and Twitter data are analyzed to predict car sales using support vector regression. A multi-class sentiment analysis method is suggested in [6] with the goal of accurately identifying the sentiment contained in the tweet by linking labels that represent all of the sentiments that are now present.

Researchers have been motivated to employ such approaches to address the topic task by the pertinent outcomes that Deep Learning has produced in Natural Language Processing (NLP). It has been demonstrated in [20] that a particular kind of neural network with convolutional and recurrent layers can outperform recursive SVM in classification tasks. The approach introduced in [35] to enhance sentiment analysis and sarcasm detection was developed in part because of this work. In order to create a sarcasm detection system, sentiment, mood, and personality traits are extracted using a pre-trained DL model in [19]. The majority of research on the Italian language has focused on the automatic identification of irony, a linguistic construct that is similar to sarcasm. One could presume that sarcasm is a subset of irony. In [5], a Decision Tree method for irony detection is trained using characteristics like word frequency, synonyms, word ambiguity, and parts of speech. Researchers have been challenged by EVALITA-2018 to create an automated system that can: a) recognize irony on its own; and b) recognize sarcasm as a subclass of irony. A bidirectional recurrent neural network with convolutional layers and a Gated Recurrent Unit was introduced for the competition in [21]. For task a), it receives an F1-Score of 61%, whilst for job b), it receives a score of 46.5%.

4. A NEW TECHNIQUE BASED ON DEEP LEARNING NEURAL ARCHITECTURES TO SUPPORT SENTIMENT MINING AND ANALYSIS

4.1 Preprocessing

According Every ML-based system must have a preprocessing phase, whose primary goal is to prepare the input for the model's analysis. Such a task involves removing biased parts, keeping as much of the original information as feasible, and changing the input data to arrange it coherently with the model features.
One popular method in NLP is to arrange the input text as a series of tokens. A token can be considered the text's fundamental unit, and the selection policy may have an impact on the system's performance. To enable the system to accomplish its goal, an effective tokenization process should only designate as tokens portions of the input that are information-rich. We have decided to use words and punctuation as tokens in this instance. By

making this decision, we are able to fully capture the input text while simultaneously accounting for its syntactical structure.
Following tokenization, a suitable padding operation was carried out to ensure that the tweet's length was consistent with the average length, which is 15 tokens. After that, the input data is vectorially represented by satisfying the core model's requirements. A pre-trained version of the effective embedding tool FastText [24], a library for the effective learning of word representations and sentence categorization, is used to uncover such representation. Each token is arranged as a series of multidimensional vectors and mapped to a 300-dimensional space for each input sentence. Ultimately, the preprocessing phase yields a matrix of dimension L x N for each input phrase, where L is the number of tokens and N is the embedding dimension (i.e., 300).

4.2. Primary Model

The primary model (BILSTM+CONV from now on) used to address the issue is comparable to the one presented in [35]. It is a Deep Neural Network that combines the attention mechanism with both recurrent and convolutional architectures. The model is unique in that it considers a certain collection of calculated punctuation-based attributes that have been shown to enhance system performance. A figurative text and symbolic clues that help determine whether a tweet is sarcastic are often indicated by features like the number of exclamation points (!) in the tweet, the number of question marks (?) in the tweet, the number of periods (.), the number of capital letters in the tweet, and the number of uses of or in the tweet. The input layer of the model architecture receives the matrices of real numbers that were produced during the preprocessing stage. This layer arranges the data such that the following layer, which is based on Long Short Term Memory (LSTM) units, can analyze it [23].

An RNN called LSTM was developed to address the shortcomings of traditional RNN systems. The LSTM formulation was created by researchers to address the well-known disappear gradient issue [23], which causes network training to stall. A sequence is examined element by element by the LSTM unit.

4.3. Secondary Frameworks

In order to continue performance analysis, the main model has been compared with a collection of DL systems that were inspired by it but had some DL layers removed. By examining the environment that enables the network to infer the input space as effectively as possible, in addition to the contribution of layers, we have developed four alternative architectures at varying levels of complexity. Our suggested architectures don't take into account further features; they just look at the tokens mentioned in section 4.1. Because we want to examine how many DL models may approach this task by relying solely on words and punctuation marks, we refrained from using auxiliary features in secondary models. We then formulated the study question to determine whether the auxiliary features are useful. Below is a description of secondary models.

S-LSTM: S-LSTM, which stands for simple LSTM, is a two-layer DL architecture. The first one, an RNN made up of LSTM neural units, examines the input sequence by producing a vector that is calculated using both the elements and their configurations. Utilizing the internal feedback loops of the LSTM, which enable the network to take into account both long-term (i.e., among long-distance components) and short-term (i.e., among short-distance elements) interactions, the final output of such a layer is obtained. A fully linked neural network made up of two neurons that are triggered by the softmax function makes up the second layer. When applied to the network's output, the softmax function translates its value within a range between 0 and 1 so that its total is equal to the unit. This indicates that the likelihood that the input text falls into one of the two classifications is the network's ultimate output.

Attentional LSTM is referred to as ATT-LSTM. The attention mechanism that is added after the LSTM layer is the only difference between this architecture and the S-LSTM one. In order to help the network weigh the input sequence items effectively and complete the task at hand, the attention layer calculates attention weights. Only the internal states of the recurrent neural network are utilized in this instance, and the attention mechanism is the same as that used for the main model (see sec.) [18]. An architecture called ATT-LSTM has been used in the past to address a number of tasks, including autonomous text complexity evaluation for the English and Italian languages [34][33].

BI-LSTM: The acronym for bidirectional LSTM is BI-LSTM. It is made up of two layers. The first is a bidirectional LSTM, which is made up of two LSTM layers that relate input sequence items in a forward and backward manner, respectively. The amount of input data that can be analyzed can be increased by looking at the input sequence in these opposite orientations. Additionally, it helps the recurrent neural network remember knowledge about the past and future that the network might otherwise find difficult to recall. In this instance, we have decided to concatenate the two vectors that the bidirectional LSTM produces, which must be combined by selecting a merger policy. The softmax function will then be used to activate the computed output once it has been provided as input to a dense layer made up of two neurons. The acronym for bidirectional attentional LSTM is BI-ATT-LSTM, which is a BI-LSTM enhanced by an attention layer. Although it may result in overfitting, this network design is the most potent of the secondary models and can accurately infer the input space. In order to help the network overcome its memory limitations, the BI-ATT-LSTM relates the BI-LSTM layer methodology with the attention mechanism that weights the internal state taken by the LSTMs during the sequence analysis. It has already been utilized to address issues in other application areas [18], and rigorous training is necessary to enable the network to effectively generalize the solution for the task at hand.

4.4. Characteristics

The primary DL system's parameters have been calculated empirically, drawing inspiration from [35]. The parameters of the secondary models are identical to those of the primary models.
Specifically, there are 500 LSTM cells in the BiLSTM. A Gaussian distribution of the random values to be generated is used to initialize the network's weights. A 0.01 standard deviation has been established. 20% of the weights can be removed from the recurrent network using the dropout regularization technique, compared to 40% for the convolutional layers. We employ Adam with a learning rate of 0.0001 for the optimization.
They are listed in the table for simplicity's sake.

5. EXPERIMENTAL EVALUATION AND OUTCOMES

5.1. The corpus

Despite being a pertinent task, automatic irony detection for the Italian language is a relatively new field of study with little prior research. The study field's slow growth, which is typified by a lack of resources for creating and testing categorization algorithms, is the result of the topic's lack of attention. We use a dataset designed for the automatic detection of irony in the Italian language, which was reported in [5], to complete the task. It is a compilation of 3,185 humorous texts taken from the Spinozait Twitter account and Spinoza blog. Spinozabis is a well-known blog with satirical political content that is widely accepted as a source of irony. In contrast, the corpus includes 22.295 non-ironic tweets that were taken from the accounts of seven well-known daily newspapers: La Stampa, Il Messaggero, Repubblica, Il Resto del Carlino, Il Sole 24 ore, Gazzetta dello Sport, and Corriere della Sera. The decision to use newspaper tweets as bad examples stems from the fact that Spinoza and the media both address comparable subjects, primarily news and politics. Additionally, Spinoza writes in a style that is similar to that of newspapers.

5.2. Tests

The goal of the testing phase is to gauge how well the network performs in linking input texts to the appropriate classes. The dataset's imbalance and the limited quantity of tweets available for training and testing systems are the two primary issues it has considered. We have chosen to utilize the entire dataset, despite the fact that there are significantly fewer positive examples (3,185) than negative ones (22.295). We accomplished this by connecting subsets of negative cases with positive examples in a series of tests. In specifics, we repeat tests by matching the initial C negative example to the C positive example, where C is the number of positive cases. By matching the second C negative examples to the complete set of positive examples, we generate a second subset of data. Following i steps, we pair the negative components from positions $i * C$ to $i * C + C - 1$ with the complete set of positive cases to construct the subset.

In order to address the limited quantity of accessible texts, we have utilized the popular cross-validation technique known as K-Fold, which has demonstrated the ability to generalize the system's learned knowledge in situations where data is inadequate. The dataset is divided into K subsets using the K-Fold method. Alternatively, the remaining data is used to train the model, and each subset is used as a test set. The testset is used to calculate the performance of each trained model. The measurements are then averaged.

Here, a 10-Fold is applied to each pair of negative instances and positive examples, and the F1-Score, Precision, Recall, and Accuracy metrics are calculated. Over the course of the runs, partial results are averaged.

5.3. Conversation

The outcomes displayed in Table 2 demonstrate the systems' generally strong performance. Nearly every evaluated model achieves similar F1-Score and Accuracy values. The most sophisticated system we have utilized is the BIL-STM+CONV neural network, which was developed to particularly address sarcasm detection. It integrates all of the DL architectures discussed in the paper and also looks at auxiliary characteristics that are generated on the fly. Despite its complexity, investigations show that it produces a slightly higher F1-Score and the same accuracy value as previous systems. The best rival is the BI-ATT-LSTM, which uses a simpler design yet obtains the closest accuracy and F1-Score values. The latter just employs a dense layer and does not take use of the CNN. By increasing the number of training epochs, other systems can attain outcomes that are close to the top ones, with the exception of the S-LSTM, which performs the poorest. The BI-LSTM and ATT-LSTM require 40 and 48 epochs of training, respectively, but the BI-ATT-LSTM and the BILSTM+CONV require only 37 epochs.

The findings indicate that the network's performance is unaffected by the quantity of exclamation points, question marks, periods, capital letters, and other punctuation used in the tweets. In fact, even though the BILSTM+CONV model solely considers these auxiliary features, the comparable designs accomplish the same outcomes by utilizing just words and punctuation. The outcomes are comparable across all calculated measures, not just when taking the F1-Score into account. Stated otherwise, we can confirm that their performances are nearly identical.

We may draw the conclusion that, in the Italian context, the experimental that makes use of the Spinoza corpus and tweets extracted from a number of newspapers demonstrates that auxiliary traits are insufficient to assist the categorization of ironic-sarcastic texts.

The compared performances in Table 2 indicate that the ATT-LSTM model should be selected as the preferred model when considering computing effort. In fact, its architecture is simpler even if a larger number of epochs are required to attain the greatest outcomes. By selecting a lighter architecture, we can consider apps that can be integrated into mobile devices without requiring the use of traditional rest-based services and a main workstation. These apps could help users by connecting the proposed DL system for irony-sarcasm detection with other apps, such social media apps. By providing real-time assistance to those who struggle to understand such linguistic strategies, sarcasm-irony may be automatically identified.

A decision tree was trained and trials were carried out by utilizing the 10-Fold approach in [5]. Even though the methods used for system testing varied slightly, the outcomes are still same. The authors of [5] also use a 10-Fold approach to calculate F1-Scores, recall, and precision. According to the comparison, our top model's F1-Score was +22%.

6. CONCLUSIONS AND FUTURE WORK

In this work, we have examined and contrasted five deep learning approaches for automatically detecting irony and sarcasm. To compare such systems, an accurate testing phase has been discovered. It demonstrates that nearly every system can achieve high performance and highlights the role that DL layers play in addressing this issue. The findings indicate that while a basic model like an S-LSTM is insufficient to address the issue, a combination of a fully linked layer, a bidirectional LSTM, and an attention mechanism offers a decent balance between complexity and performance. Future research will look at using these findings for text complexity evaluation in order to improve state-of-the-art outcomes and enhance the automatic text complexity evaluation system [26][27]. Considering data approximation paradigms (e.g., [11, 15, 14, 13, 12, 10, 9, 1]) is another study avenue that could be helpful in addressing the complexity of newly emerging big data environments.

References

1. Rulkov. Ahn, S., Couture, S. V., Cuzzocrea, A., Dam, K., Grasso, G. M., Leung, C. K., McCormick, K. L., and Wodi, B. H. A fuzzy logic based machine learning tool for supporting big data business analytics in complex artificial intelligence environments. In 2019 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2019, New Orleans, LA, USA, June 23-26, 2019 (2019), IEEE, pp. 16.
2. Aloufi, S., and El Saddik, A. Sentiment identification in football-specific tweets. *IEEE Access* 6 (2018), 7860978621.
3. Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, 2014.
4. Bamman, D., and Smith, N. A. Contextualized sarcasm detection on twitter. In *Ninth Interna- tional AAAI Conference on Web and Social Media* (2015).
5. Barbieri, F., Ronzano, F., and Saggion, H. Italian irony detection in twitter: a first approach. In *The First Italian Conference on Computational Linguistics CLiC-it (2014)*, vol. 28.
6. Bouazizi, M., and Ohtsuki, T. Multi-class sentiment analysis in twitter: What if classification is not the answer. *IEEE Access* 6 (2018), 6448664502.
7. Butler, H. E., et al. *The Instituto Oratoria of Quintilian: With an English Translation*, vol. 1. W. Heinemann, 1953.

8. Campan, A., Cuzzocrea, A., and Truta, T. M. Fighting fake news spread in online social networks: Actual trends and future research directions. In *2017 IEEE International Conference on Big Data*, BigData 2017, Boston, MA, USA, December 11-14, 2017 (2017), IEEE Computer Society, pp. 44534457.
9. Ceci, M., Cuzzocrea, A., and Malerba, D. Effectively and efficiently supporting roll-up and drill-down OLAP operations over continuous dimensions via hierarchical clustering. *J. Intell. Inf. Syst.* 44, 3 (2015), 309333.
10. Cuzzocrea, A. Improving range-sum query evaluation on data cubes via polynomial approximation. *Data Knowl. Eng.* 56, 2 (2006), 85121.
11. Cuzzocrea, A., and Chakravarthy, S. Event-based lossy compression for effective and efficient OLAP over data streams. *Data Knowl. Eng.* 69, 7 (2010), 678708.
12. Cuzzocrea, A., Furfaro, F., and Sacca, D. ‘ Hand-olap: A system for delivering OLAP services on handheld devices. In *6th International Symposium on Autonomous Decentralized Systems (ISADS 2003)*, 9-11 April 2003, Pisa, Italy (2003); IEEE Computer Society, pp. 8087.
13. Cuzzocrea, A., and Matrangolo, U. Analytical synopses for approximate query answering in OLAP environments. In *Database and Expert Systems Applications, 15th International Conference*, DEXA 2004 Zaragoza, Spain, August 30-September 3, 2004, Proceedings (2004); F. Galindo, M. Takizawa, and R. Traunmuller, Eds., vol. 3180 of *Lecture Notes in Computer Science*, Springer, pp. 359370.
14. Cuzzocrea, A., Sacca, D., and Serafino, P. ‘ A hierarchy-driven compression technique for advanced OLAP visualization of multidimensional data cubes. In *Data Warehousing and Knowledge Discovery, 8th International Conference*, DaWaK 2006, Krakow, Poland, September 4-8, 2006, Proceedings (2006); A. M. Tjoa and J. Trujillo, Eds., vol. 4081 of *Lecture Notes in Computer Science*, Springer, pp. 106119.
15. Cuzzocrea, A., and Serafino, P. LCS-hist: taming massive high-dimensional data cube compression. In *EDBT 2009, 12th International Conference on Extending Database Technology*, Saint Peters- burg, Russia, March 24-26, 2009, Proceedings (2009);
16. M. L. Kersten, B. Novikov, J. Teubner, V. Polutin, and S. Manegold, Eds., vol. 360 of *ACM International Conference Proceeding Series*, ACM, pp. 768779.
17. Di Gangi, M., Lo Bosco, G., and Pilato, G. Effectiveness of data-driven induction of semantic spaces and traditional classifiers for sarcasm detection. *Natural Language Engineering* 25 (2019), 257285.
18. DAvanzo, E., Pilato, G., and Lytras, M. Using twitter sentiment and emotions analysis of google trends for decisions making. *Program* (2017).
19. Felbo, B., Mislove, A., Sgaard, A., Rahwan, I., and Lehmann, S. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceed- ings of the 2017 Conference on Empirical Methods in Natural Language Processing (Copenhagen, Denmark, Sept. 2017)*, Association for Computational Linguistics, pp. 16151625.
20. Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99*. (Conf. Publ. No. 470) (1999), vol. 2, pp. 850855 vol.2.
21. Ghosh, A., and Veale, T. Fracking sarcasm using neural network. In *Proceedings of the 7th work- shop on computational approaches to subjectivity, sentiment and social media analysis* (2016), pp. 161169.
22. Giudice, V. Aspie96 at ironita (evalita 2018): Irony detection in italian tweets with character-level convolutional rnn. *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA18)* (2018), 160165.
23. Gonzalez-Ib anez, R., Muresan, S., and Wacholder, N. Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (2011), pp. 581586.
24. Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.
25. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. Learning word vectors for 157 languages. *CoRR* abs/1802.06893 (2018).
26. Kreuz, R. J., and Glucksberg, S. How to be sarcastic: The echoic reminder theory of verbal irony. *Journal of experimental psychology: General* 118, 4 (1989), 374.
Lo Bosco, G., Pilato, G., and Schicchi, D. A neural network model for the evaluation of text complexity in italian language: a representation point of view. *Procedia computer science* 145 (2018), 464470.
27. Lo Bosco, G., Pilato, G., and Schicchi, D. A sentence based system for measuring syntax complex- ity using a recurrent deep neural network. In *2nd Workshop on Natural Language for Artificial Intelligence, NL4AI 2018*, (2018), vol. 2244, CEUR-WS, pp. 95101.
28. Maynard, D. G., and Greenwood, M. A. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *LREC 2014 Proceedings* (2014), ELRA.
29. McDonald, S. Exploring the process of inference generation in sarcasm: A review of normal and clinical studies. *Brain and language* 68, 3 (1999), 486506.
30. Oraby, S., Harrison, V., Reed, L., Hernandez, E., Riloff, E., and Walker, M. Creating and characterizing a diverse corpus of sarcasm in dialogue. arXiv preprint arXiv:1709.05404 (2017).
31. Pai, P.-F., and Liu, C.-H. Predicting vehicle sales by sentiment analysis of twitter data and stock market values. *IEEE Access* 6 (2018), 5765557662.
32. Reyes, A., and Rosso, P. On the difficulty of automatically detecting irony: beyond a simple case of

negation. *Knowledge and Information Systems* 40, 3 (2014), 595614.
33. Schicchi, D., Pilato, G., and Lo Bosco, G. Attention-based model for evaluating the complexity of sentences in english language. In *20th ieee mediterranean eletrotechnical conference* (2020), p. in press.
34. Schicchi, D., Pilato, G., and Lo Bosco, G. Deep neural attention-based model for the evaluation of italian sentences complexity. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)* (2020), pp. 253256.
35. Son, L. H., Kumar, A., Sangwan, S. R., Arora, A., Nayyar, A., and Abdel-Basset, M. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. *IEEE Access* 7 (2019), 2331923328.
36. Terrana, D., Augello, A., and Pilato, G. Facebook users relationships analysis based on sentiment classification. In *2014 IEEE International Conference on Semantic Computing* (2014), IEEE, pp. 290296.
37. Tsur, O., Davidov, D., and Rappoport, A. Icwsma great catchy name: Semi-supervised recogni- tion of sarcastic sentences in online product reviews. In *fourth international AAAI conference on weblogs and social media* (2010).
38. Vassallo, G., Pilato, G., Augello, A., and Gaglio, S. Phase coherence in conceptual spaces for conversational agents. *Semantic computing* (2010), 357371.
39. Wallace, B. C., Kertz, L., Charniak, E., et al. Humans require context to infer ironic intent (so computers probably do, too). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers) (2014), pp. 512516.
40. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chap- ter of the Association for Computational Linguistics: Human Language Technologies (San Diego, California, June 2016)*, Association for Computational Linguistics.