Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 1086 – 1106
© TÜBİTAK
doi:10.3906/elk-1907-11

Satire identification in Turkish news articles based on ensemble of classifiers

Aytuğ ONAN[1], **Mansur Alp TOCOGLU**[2,*]
[1]Department of Computer Engineering, Faculty of Engineering and Architecture,
Izmir Katip Çelebi University, Izmir, Turkey
[2]Department of Software Engineering, Faculty of Technology, Manisa Celal Bayar University, Manisa, Turkey

Received: 02.07.2019 • **Accepted/Published Online:** 30.10.2019 • **Final Version:** 28.03.2020

Abstract: Social media and microblogging platforms generally contain elements of figurative and nonliteral language, including satire. The identification of figurative language is a fundamental task for sentiment analysis. It will not be possible to obtain sentiment analysis methods with high classification accuracy if elements of figurative language have not been properly identified. Satirical text is a kind of figurative language, in which irony and humor have been utilized to ridicule or criticize an event or entity. Satirical news is a pervasive issue on social media platforms, which can be deceptive and harmful. This paper presents an ensemble scheme for satirical news identification in Turkish news articles. In the presented scheme, linguistic and psychological feature sets have been utilized to extract the feature sets (i.e. linguistic, psychological, personal, spoken categories, and punctuation). In the classification phase, accuracy rates of five supervised learning algorithms (i.e. naive Bayes algorithm, logistic regression, support vector machines, random forest, and k-nearest neighbor algorithm) with three widely utilized ensemble methods (i.e. AdaBoost, bagging, and random subspace) have been considered. Based on the results, we concluded that the random forest algorithm yielded the highest performance, with a classification accuracy of 96.92% for satire detection in Turkish. For deep learning-based architectures, we have achieved classification accuracy of 97.72% with the recurrent neural network architecture with attention mechanism.

Key words: Satire detection, figurative language, machine learning, classifier ensembles

1. Introduction

Recent advances in information technologies made it possible to reach immense quantities of unstructured online text documents. The online content available on the Web consists of figurative and nonliteral linguistic elements, including metaphor, analogy, irony, and satire. In figurative language, irony, sarcasm, or satire may be employed to express more complex issues. Hence, the identification of figurative language can be a challenging problem for not only computers, but also for human beings. The identification of figurative language is a fundamental task for sentiment analysis. The classification accuracy of sentiment analysis schemes may be degraded if elements of figurative language have not been properly addressed [1, 2].

Satire is a way of criticizing or ridiculing people, events, or issues with the use of humor, irony, and exaggeration (https://en.oxforddictionaries.com/definition/satire) [3]. Satirical text is a kind of nonliteral/figurative communication employed by people to expose and criticize people, issues, and contemporary political events in a humorous way. With the progress in Internet technology, information sharing in forms of news has also increased. The forms of figurative language can be employed in news articles shared on social media platforms.

[*]Correspondence: mansur.tocoglu@gmail.com

News satire (fake news) is a kind of expression encountered especially in journalism, in which the elements of figurative languages, such as satire, irony, and sarcasm, have been employed [4]. News satire is a pervasive phenomenon on the Web.

Many research studies on natural language processing have been conducted to identify figurative language elements, such as sarcasm and irony [5–7]. Compared to the automatic identification schemes for sarcasm and irony, machine learning-based identification of satirical text has been addressed in a few studies [8]. In addition, former research works on satire identification have been concentrated on the English language. In this regard, we present a classifier ensemble approach for satirical news identification in Turkish news articles. We have gathered 17,857 satirical news articles and 17,348 nonsatirical news articles from two distinct sources to examine the performance of machine learning algorithms and feature sets on satirical text detection in terms of predictive efficiency. In the feature extraction, LIWC (Linguistic Inquiry and Word Count) has been utilized [9]. LIWC is exploratory text mining software to extract psychological and linguistic feature sets from text documents. LIWC can be successfully utilized for identification of figurative language on social media platforms [10]. Psycholinguistic feature sets have been already employed in identification of several forms of figurative language, such as humorous text identification [11], sarcasm, and nastiness identification [12]. Features can be obtained in five major dimensions with LIWC software: language processes, psychological procedures, personal issues, spoken issues, and punctuation marks. Five LIWC dimensions and their ensemble combinations were taken into consideration. In the classification phase, accuracy rates of five supervised learning algorithms (i.e. naive Bayes algorithm, logistic regression, support vector machines, random forest, and k-nearest neighbor algorithm) with three widely utilized ensemble methods (i.e. AdaBoost, bagging, and random subspace) have been considered.

In this article, we propose an approach to detect satire in Turkish. The study's primary contributions can be summarized as follows: To the best of our knowledge, it is the first study of satire detection in Turkish based on machine learning. The present system empirically assesses the efficacy of satire detection based on psychological and linguistic feature sets. In addition, we have examined the predictive performance of five deep learning-based architectures in conjunction with conventional word embedding schemes for satire identification.

The rest of this article is organized as follows: in Section 2, we address briefly the related research on automatic identification of figurative language with emphasis on detection of satire. Section 3 provides the study's methodology. We present preprocessing methodology, feature extraction techniques, classification algorithms, and ensemble techniques. Experimental results are presented in Section 4. The last section concludes the paper, providing a summary of the study.

2. Related work

The automatic identification of figurative language in text documents can be viewed as a challenging task [3]. The previous research on automatic identification of figurative language with emphasis on satire detection is provided in this section.

In [13], Ahmad et al. introduced an approach based on machine learning to filter satirical or real news from journals. In the presented scheme, four feature weighting schemes (namely binary feature weighting, TF-IDF method, binormal separation method, and TF-IDF-BNS) were employed for feature engineering. The experimental procedure, conducted on support vector machines, indicated that the utilization of the TF-IDF method in conjunction with binormal separation can result in promising outcomes for news corpus satire identification. In [14], Barbieri et al. introduced a satirical text identification strategy based on machine learning for advertising news in Spanish. In the presented scheme, several feature engineering schemes (namely frequency of a word of each tweet, ambiguity of words, parts of speech tags, frequency of synonyms, and sentiment orientations) have been taken into account. Empirical findings on separate feature engineering schemes indicated that linguistic feature sets on satirical text classification can produce higher predictive efficiency compared to standard representation schemes, such as bag-of-word representation. In another study, Barbieri et al. [15] evaluated the classification accuracy of machine learning-based schemes for satirical news identification on Twitter in a multilinguistic context, where English, Spanish, and Italian messages were considered. The classification accuracies across different languages have also been examined. Rubin et al. [11] focused on the predictive performance of ensemble feature sets for satirical text detection in another study by taking absurdity, humor, grammar, negative affect, and punctuation into account. The experimental results on the support vector machine algorithm indicated that the ensemble feature sets can provide higher predictive performance in the automatic detection of fake and real news documents. Similarly, for satirical news identification, Perez-Rosas et al. [16] implemented a machine learning strategy. Various feature engineering schemes (i.e. unigrams, bigrams, punctuation marks, psycholinguistic feature sets, readability characteristics, and syntax characteristics) were regarded in the empirical assessment. Similarly, Ravi and Ravi [17] implemented an ensemble classification strategy on customer reviews and news. Linguistic, semantic, psychological, and unigram-based features were extracted. The ensemble classification framework used several supervised learning algorithms, including support vector machines, logistic regression, the random forest algorithm, the naive Bayes algorithm, and multilayer perceptron. The experimental findings indicated that feature sets based on LIWC yielded promising performance in detection of satire and irony. In another work, Ahmed et al. [18] examined the predictive efficiency of various linguistic systems, such as unigram, bigram, trigram, and four-gram models and two weighting systems, i.e. term-frequency and TF-IDF-based representation, in identifying fake and valid online news. In another study, Yang et al. [19] introduced a deep learning-based architecture to detect satirical news. Hierarchical neural networks and linguistic word embedding systems were used in the presented architecture. Recently, Ravi and Ravi [20] presented an ensemble scheme for irony identification, which integrates syntactic, semantic, and psycholinguistic feature sets.

The earlier works on satire identification focused on conventional linguistic and psycholinguistic feature sets in the English language. In the classification phase, conventional classification algorithms, which are support vector machines, naive Bayes, and C4.5, have been used. This paper differs from the earlier literature in the field in terms of several aspects. In this paper, we present a comprehensive analysis on ensemble feature sets obtained by different psycholinguistic feature sets. In addition, conventional learning methods and ensemble learners have been evaluated in the classification phase. The empirical results also report the predictive performance of deep learning-based algorithms on Turkish satire identification. All the empirical analysis has been conducted on Turkish news articles.

3. Methodology

The research methodology is provided in this section. The presented scheme consists of three phases: collection and preprocessing of the dataset, extraction of features based on LIWC, and classification based on supervised learners and ensemble methods. The general structure of the ensemble scheme based on LIWC features and classifiers has been presented in Figure 1. This section introduces briefly the techniques used in the system.

3.1. Dataset collection and preprocessing

In this study, we created a dataset for the detection of the existence of satirical text in Turkish. To do so, we collected Turkish news from two sources where the related news is categorized as satirical and nonsatirical. We gathered the satirical news from the archives of Zaytung, which is a well-known online satirical newspaper in Turkey. On the other hand, to gather the nonsatirical news, we collected tweets of seven newspaper Twitter accounts, which are Milliyet, BBC Turkish, CNNTÜRK, Habertürk, Hürriyet, NTV, and TRTHaber. Table 1 shows the overall number of documents and vocabularies of the collected news based on two categories individually.

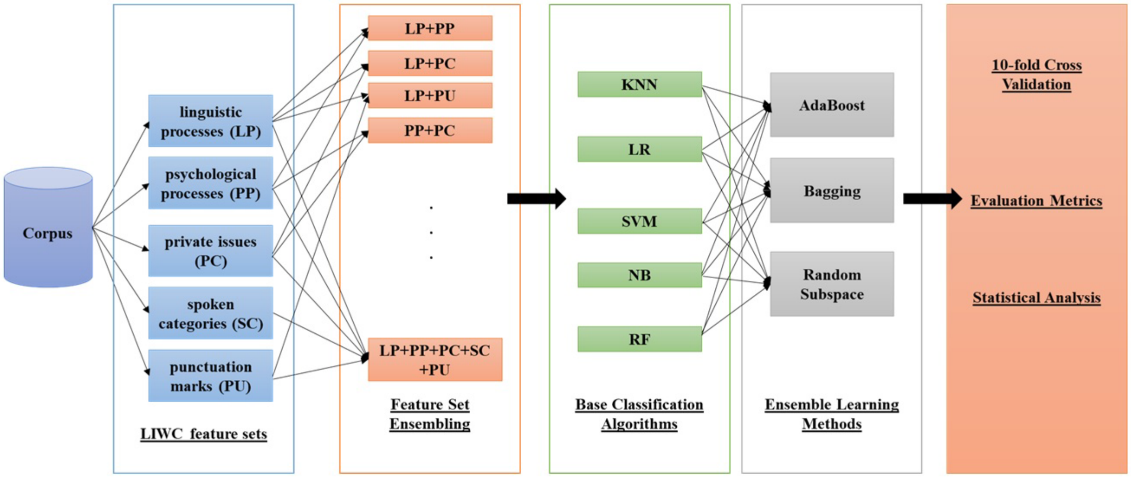

Figure 1. The general structure of the ensemble satire identification framework.

Table 1. Descriptive information of the dataset used in analysis.

Category	# of documents	Total vocabulary	Total unique vocabulary
Nonsatirical	17,348	235,481	36,478
Satirical	17,857	683,300	95,405
Total	35,205	918,781	111,361

To examine the effect of different preprocessing strategies on the corpus, we performed empirical analysis with four different configurations. Initially, we removed all punctuation marks, numeric characters, and extra spaces (denoted by “with preprocessing” in Figure 2). For this scheme, we have not included smileys, emojis, mentions, and hashtags. In addition, we have also examined the predictive performance of classifiers when emojis are added, mentions and hashtags are added, and punctuation marks are added. The average predictive performances (in terms of classification accuracy) of different strategies are summarized in Figure 2. Since Twitter-specific symbols (i.e. mentions and hashtags) may potentially affect the meaning of tweets, they have not been eliminated for the empirical results listed in Section 4. In addition, smileys and emojis have been taken as the feature set. Since punctuation marks and numeric characters may increase or invert the meaning of tweets, punctuation marks have not been eliminated during the preprocessing phase.

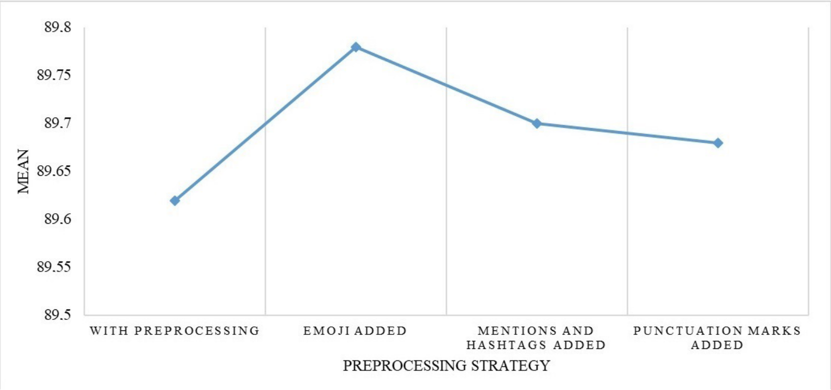

Figure 2. The main effects plot for preprocessing strategies on the corpus.

3.2. Feature extraction

In this research, psycholinguistic sets were obtained using LIWC (Linguistic Inquiry and Word Count). For satirical text identification, the psycholinguistic function subsets were empirically assessed. Psycholinguistic feature sets based on LIWC have already been used in several computational linguistic applications, including sarcasm detection [12], satire identification [9, 16], and humorous text detection [21].

LIWC is an application for text analysis to extract emotional, cognitive, and structural properties from samples of verbal and written speech. The first version of LIWC was launched in 1993 and the last version of LIWC, which is called LIWC2015, was published in 2015 [22, 23]. LIWC software comprises dictionaries in various languages such as English, Turkish, Spanish, German, Dutch, Norwegian, Italian, and Portuguese. The dictionaries for the other languages were acquired from the English dictionary translations. The LIWC dictionary consists of about 6,400 phrases, stems of phrases, and emoticons. Each dictionary entry corresponds to one or more categories of words or subdictionaries. It is possible to assign each word into one or more categories. Furthermore, categories can be split into five primary aspects (subsets), which are linguistic processes, psychological processes, personal concerns, spoken categories, and punctuation [23]. The LIWC feature sets and categories are provided in Table 2. In Table 3, the traditional LIWC dimension values for satirical and nonsatirical news articles are presented, where traditional LIWC dimensions denote the percentage of total words within the text. In addition, four summary variables (i.e. analytic, clout, authenticity, and emotional tone scores) have been provided. In addition to the basic descriptive information for satirical and nonsatirical news articles, Table 3 contains four instances with the corresponding LIWC values. Regarding the summary variables for news articles on the corpus, the average authenticity values for nonsatirical news are higher, while the emotional tone values are higher for satirical news.

3.3. Supervised learning methods

In this study, we used five supervised learning algorithms, the naive Bayes algorithm, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm, which are applied to assess the predictive efficiency of various function sets on satire detection.

3.3.1. Naive Bayes algorithm

Naive Bayes (NB) is a statistical learning algorithm based on Bayes' rule and conditional independence assumption. The conditional hypothesis of independence considers that, given the class, the attributes are conditionally independent. The assumption simplifies the required computations involved in the algorithm. In response, the algorithm is efficient and can scale well. Though the assumption may not be valid for every real-world case, the naive Bayes algorithm can yield promising results in machine learning tasks, including text classification and sentiment analysis [24, 25].

Table 2. LIWC feature sets and categories [22, 23].

Feature set	Categories
Linguistic Processes	Number of words, number of pronouns, number of personal pronouns, articles, prepositions, auxiliary verbs, adverbs, conjunctions
Psychological Processes	Affective words, positive emotions, negative emotions, social words, cognitive words, perceptual words
Personal Concerns	Work, leisure, home, money
Spoken Categories	Assent, Nonfluencies, fillers
Punctuation	Number of punctuation marks, periods, commas, colons, semicolons, question marks, exclamation marks, dashes

Table 3. LIWC dimension values for satirical and nonsatirical news.

Traditional LIWC dimensions	Instance 1	Instance 2	Instance 3	Instance 4	Average for nonsatirical news	Average for satirical news	Average for social media: Twitter, Facebook, blogs
I-Words (I, ME, MY)	0.00	0.00	0.00	0.00	0.00	0.20	5.51
Social words	11.80	9.10	11.10	13.00	7.50	9.60	9.71
Positive emotions	0.00	0.00	0.00	0.00	1.00	2.80	4.57
Negative emotions	5.90	9.10	3.70	0.00	3.10	3.00	2.10
Cognitive processes	0.00	9.10	7.40	8.70	5.60	6.60	10.77
Summary variables							
Analytic	99.00	99.00	95.80	92.80	98.80	98.60	55.92
Clout	88.00	81.80	64.50	95.90	75.70	80.70	55.45
Authenticity	66.30	85.20	15.10	9.30	48.00	21.00	55.66
Emotional tone	1.00	1.00	1.00	25.80	4.40	22.10	63.35
Instance 1	Adana’nın ceyhan ilçesinde bir evde çıkan yangında 5 yaşındaki çocuk hayatını kaybetti						
Instance 2	Edirne’de evde çıkan yangında dumandan etkilenen yacslı kadın hayatını kaybetti						
Instance 3	Yerel seçimlere doğru: 56 yıldır Çaycuma Belediye başkanlığını sürdüren Rıza Sarıkaya (92): “Koltuklar kimsenin babasının malı değil...”						
Instance 4	Tatil İçin Türkiye’yi tercih eden ABD’li Richards ailesi, ülkemize bayıldı: “1 ay daha kalırsak oteli satın alabiliyoruz...”						

3.3.2. Logistic regression classifier

Logistic regression (LR) [25, 26] is a supervised learning algorithm. The algorithm provides a scheme to apply linear regression to classification tasks. It employs a linear regression model and transformed target variables have been utilized to construct a linear classification scheme.

3.3.3. Support vector machines

Support vector machines (SVMs) [25, 27] are linear algorithms that can be used for classification and regression. For the two-class classification tasks, the algorithm finds a hyperplane in the higher-dimensional space such that the instances of two classes have been separated as much as possible. In response, the algorithm has a good generalization ability on newly encountered instances. The algorithm can build suitable learning models in the case of a large amount of data.

3.3.4. Random forest algorithm

Random forest (RF) is a supervised learning algorithm, combining the bagging algorithm and random method of subspace [25, 28]. Decision trees were used as the base learning algorithm in this algorithm. Each tree was constructed based on training data bootstrap samples. A random selection of features was used to provide the variety among the base learners. In response, the model can yield promising learning models on datasets with noisy or irrelevant data.

3.3.5. K-nearest neighbor algorithm

K-nearest neighbor (KNN) is a learning algorithm for supervised learning tasks, including classification and regression tasks [26]. In this scheme, the class label for an instance has been determined based on the similarity of the instance to its nearest neighbors in the training set. In this scheme, all the instances have been stored, and at the time of classification, the class label has been identified based on the examination of the k-nearest neighbors.

3.4. Ensemble methods

Ensemble learning refers to the process of combing the predictions of multiple supervised learning algorithms and treating the algorithms as a committee of decision makers [25]. Ensemble learning schemes seek to identify a more accurate classification model. In this study, we used the ensembles of the five supervised learning algorithms with three well-known ensemble learning methods, which are AdaBoost, bagging, and random subspace.

3.4.1. AdaBoost algorithm

AdaBoost is an ensemble learning algorithm based on boosting [29]. The base learning algorithms were trained sequentially in the algorithm and at each round a new learning model was built. The weight values allocated to misclassified samples will be increased at each round, while the weight values allocated to properly categorized cases will be reduced. In reaction, the algorithm aims to devote more rounds to cases that are more difficult to learn and to compensate for classification mistakes produced in previous models [25].

3.4.2. Bagging algorithm

Bagging (bootstrap aggregating) [30] is another technique of constructing the ensemble. In this system, from the initial training set by bootstrap sampling, distinct training subsets were acquired. The projections produced by the algorithms of base learning were combined with the use of majority voting [25].

3.4.3. Random subspace algorithm

Random subspace [31] is another technique for constructing the ensemble. In this system, the variety between the ensemble members was accomplished in terms of the partition based on feature space. Each classification algorithm operates on various random subsets of the feature space in this algorithm. The technique thus decreases overfitting while at the same time enhancing predictive efficiency [25].

4. Experimental results and discussions

In this section, we describe the experimental procedure and the results.

4.1. Experimental procedure

In this section, we report experimental findings as the 10-fold average outcomes for each classifier. Five primary LIWC feature sets, namely linguistic processes (LP), psychological processes (PP), private issues (PC), spoken categories (SC), and punctuation marks (PM), have been taken into consideration to extract feature sets. We used feature sets and their ensemble combinations in the empirical analysis. Thirty-one function sets were acquired in this manner. To evaluate the predictive performance of LIWC-based feature sets compared to conventional text representation schemes, evaluation measures were also presented for the unigram model based on term frequency representation. We used five supervised learning algorithms, the naive Bayes algorithm, logistic regression, support vector machines, random forests, and k-nearest neighbor algorithm, and their ensembles with three learning techniques of the ensemble, namely AdaBoost, bagging, and the random subspace algorithm. In this manner, we evaluated 20 distinct algorithms of classification (5 basic learning and 15 ensembles) by using WEKA (Waikato Environment for Knowledge Analysis) version 3.9. In Table 4, the basic parameter sets for the conventional classifiers and ensemble learning methods have been presented.

Table 4. Parameter sets for classification algorithms utilized in the empirical analysis.

Classification algorithms	Parameter set
Naive Bayes algorithm	Batch size: 100, use kernel estimator: False
Logistic regression classifier	Batch size: 100, number of decimal places: 4, ridge: 1.0E-8
Support vector machines	Batch size: 100, cache size: 40.0, cost: 1, degree: 3, eps: 0.001, gamma: 0.0, loss: 0.1, kernel type: linear, number of decimal places: 2
Random forest algorithm	Size of each bag: 100, batch size: 100, number of decimal places: 2, seed: 1, number of iterations: 100
K-nearest neighbor algorithm	Number of neighbors to use: 1, batch size: 100, distance weighting: none applied, nearest neighbor search algorithm: brute force search, distance function: Euclidean measure
AdaBoost algorithm	Batch size: 100, number of decimal places: 2, number of iterations: 10, weight threshold: 100
Bagging algorithm	Size of each bag: 100, batch size: 100, number of decimal places: 2, seed: 1, number of iterations: 10
Random subspace algorithm	Batch size: 100, number of decimal places: 2, number of execution slots: 1, number of iterations: 10, seed: 1, subspace size: 0.50

For evaluation tasks on deep-learning-based architectures, the text corpus has been represented by three word embedding schemes (namely word2vec, fastText, and GloVe) in conjunction with five deep-learning architectures (namely convolutional neural network, recurrent neural network, bidirectional recurrent neural network with attention mechanism, gated recurrent unit, and long short-term memory). To implement and train deep-learning architectures, we have utilized Tensorflow and Keras. To obtain the optimal predictive performance from the models, we have employed hyperparameter optimization based on Bayesian optimization using a Gaussian process. For word2vec and fastText word embeddings, continuous skip-gram and continuous bag of words (CBOW) methods have been evaluated with varying vector sizes (vector size of 200 and 300) and different dimensions for projection layers (dimension size of 100 and 200). For the corpus, 80% of data have been utilized as the training set, whereas the rest of data are utilized as the testing set.

4.2. Experimental results and discussions

In Table 5, the accuracy and AUC (area under ROC curve) values acquired on distinct feature sets by standard supervised learning algorithms are provided. In Table 6, precision, recall, and F-measure values obtained by conventional classifiers are presented.

Regarding the predictive performance values acquired by distinct psychological and linguistic feature sets on satire identification with supervised learning algorithms presented in Table 5, classification accuracies and AUC values of random forest usually outperform the other classification algorithms. Among the five classifiers, generally the naive Bayes algorithm has the second highest predictive performance in terms of accuracy and AUC values. It is followed by support vector machines. The K-nearest neighbor algorithm achieved the lowest predictive output. The experimental evaluation aims to analyze the predictive output of LIWC-based feature sets (i.e. linguistic processes, psychological procedures, private issues, spoken categories, and punctuation) and their ensembles as feature sets for Turkish satire identification. Besides LIWC-based function sets, classification and AUC values for standard text representation schemes have been given.

As can be seen from the empirical outcomes described in Table 5 and Table 6, the use of LIWC-based feature sets outperforms the term frequency for satire identification with the unigram model (referred to as TF). With respect to the distinct feature subsets of LIWC-based sets, linguistic processes (referred to as LP) have accomplished the greatest predictive efficiency. Psychological process-based characteristics acquired the second largest predictive output, and private concerns-based feature sets reached the third largest predictive output. With respect to the predictive performance of individual feature sets, punctuation-based feature sets have obtained the lowest predictive performance. Therefore, the experimental evaluation of LIWC-based feature sets indicates that the use of characteristics, such as word count, complete pronouns, personal pronouns, articles, prepositions, etc. can be more informative to identify satirical text. We achieved the highest predictive performance among the five individual feature sets with a classification accuracy of 89.21 percent. In this system, we used language procedures in combination with random forest algorithms as function sets.

We examined the LIWC-based ensemble feature sets for Turkish satire detection as the second concern of the study. Table 5 and Table 6 present the use of ensemble feature sets combining different LIWC-based feature to provide better predictive performance compared to individually taken LIWC-based feature sets. The ensemble feature set, which integrates five LIWC-based feature sets, namely linguistic procedures, psychological procedures, private issues, spoken categories, and punctuation, has acquired the highest classification precision among the 31 feature sets considered. Using this set function in combination with the random forest algorithm results in a 93.21 percent classification and a 0.94 F-measure value.

To improve the predictive output of standard supervised learning techniques, the paradigm of ensemble learning can be used. The empirical analysis' third issue is to define whether or not ensemble learners can

Table 5. Accuracy and AUC values of conventional classification algorithms.

	Accuracy					AUC value				
Feature set	KNN	LR	SVM	NB	RF	KNN	LR	SVM	NB	RF
TF	84.57	85.26	86.78	87.72	87.8	0.85	0.857	0.873	0.883	0.884
LP	84.59	85.57	87.54	87.89	88.04	0.85	0.861	0.881	0.885	0.886
PP	84.83	86.11	87.67	88.64	88.46	0.853	0.866	0.882	0.892	0.89
PC	84.9	86.97	87.7	88.73	88.96	0.854	0.875	0.882	0.893	0.895
SC	85.21	87.08	87.8	88.9	89.12	0.857	0.876	0.884	0.895	0.897
PU	85.39	87.1	87.84	88.96	89.21	0.859	0.876	0.884	0.895	0.898
LP+PP	85.41	87.25	88.04	89.04	89.34	0.859	0.878	0.886	0.896	0.899
LP+PC	85.54	87.37	88.11	89.21	89.62	0.86	0.879	0.887	0.898	0.902
LP+SC	85.73	87.42	88.4	89.23	89.86	0.862	0.88	0.89	0.898	0.905
LP+PU	85.73	87.48	88.59	89.23	89.87	0.862	0.88	0.892	0.898	0.905
PP+PC	85.76	87.69	88.8	89.31	90.01	0.862	0.882	0.894	0.899	0.906
PP+SC	86.1	87.74	88.85	89.44	90.48	0.866	0.883	0.894	0.9	0.911
PP+PU	86.11	87.75	88.87	89.46	90.5	0.866	0.883	0.894	0.9	0.911
PC+SC	86.14	87.85	88.93	89.49	90.87	0.866	0.884	0.895	0.901	0.915
PC+PU	86.14	87.85	88.98	89.54	90.89	0.866	0.884	0.895	0.901	0.915
SC+PU	86.41	87.89	89.02	89.83	90.91	0.869	0.884	0.896	0.904	0.915
LP+PP+PC	86.57	87.97	89.43	89.89	91.06	0.871	0.885	0.9	0.905	0.917
LP+PP+SC	87.18	88.12	89.47	90.01	91.06	0.877	0.887	0.9	0.906	0.917
LP+PP+PU	87.19	88.17	89.48	90.03	91.1	0.877	0.887	0.901	0.906	0.917
LP+PC+SC	87.41	88.21	89.65	90.15	91.13	0.879	0.887	0.902	0.907	0.918
LP+PC+PU	87.41	88.29	89.67	90.34	91.17	0.879	0.888	0.902	0.909	0.918
LP+SC+PU	87.59	88.42	89.73	90.35	91.17	0.881	0.89	0.903	0.91	0.918
PP+PC+SC	87.62	88.7	90	90.36	91.28	0.881	0.892	0.906	0.91	0.919
PP+PC+PU	87.87	88.75	90.08	90.64	91.29	0.884	0.893	0.907	0.912	0.919
PP+SC+PU	87.97	88.86	90.14	90.85	91.55	0.885	0.894	0.907	0.915	0.922
PC+SC+PU	88.22	88.94	90.16	90.88	91.71	0.887	0.895	0.907	0.915	0.923
LP+PP+PC+ SC	88.41	89.01	90.34	90.91	91.83	0.889	0.896	0.909	0.915	0.925
LP+PP+PC+ PU	88.54	89.29	90.47	91.13	91.87	0.891	0.898	0.911	0.917	0.925
LP+PP+SC+ PU	88.7	89.34	90.55	91.14	91.88	0.892	0.899	0.911	0.917	0.925
LP+PC+SC+ PU	89.43	89.41	90.6	91.55	92.12	0.9	0.899	0.912	0.922	0.927
PP+PC+SC+ PU	89.52	90.13	91.59	91.78	92.58	0.901	0.907	0.922	0.924	0.932
LP+PP+PC+ SC+PU	89.61	90.77	91.97	92.67	93.21	0.901	0.913	0.926	0.933	0.939

attain better predictive output for satire detection. In this respect, we took into consideration three well-known learners of the ensemble (i.e. bagging (B), AdaBoost (A), and random forest (RS)). Tables 7-11 provide predictive performance values for ensemble learners in combination with standard learners and ensembles. As can be seen from the outcomes of predictive performance summarized in Tables 7-11, evaluation metric values

Table 6. Precision, recall, and F-measure of conventional classification algorithms.

	Precision					Recall					F-measure values				
Feature set	KNN	LR	SVM	NB	RF	KNN	LR	SVM	NB	RF	KNN	LR	SVM	NB	RF
TF	0.872	0.879	0.895	0.904	0.905	0.837	0.844	0.859	0.869	0.869	0.854	0.861	0.877	0.886	0.887
LP	0.872	0.882	0.902	0.906	0.908	0.837	0.847	0.867	0.87	0.872	0.854	0.864	0.884	0.888	0.889
PP	0.875	0.888	0.904	0.914	0.912	0.84	0.853	0.868	0.878	0.876	0.857	0.87	0.886	0.895	0.893
PC	0.875	0.897	0.904	0.915	0.917	0.841	0.861	0.868	0.879	0.881	0.858	0.879	0.886	0.896	0.899
SC	0.878	0.898	0.905	0.917	0.919	0.844	0.862	0.869	0.88	0.882	0.861	0.88	0.887	0.898	0.9
PU	0.88	0.898	0.906	0.917	0.92	0.845	0.862	0.87	0.881	0.883	0.863	0.88	0.887	0.899	0.901
LP+PP	0.881	0.899	0.908	0.918	0.921	0.846	0.864	0.872	0.882	0.885	0.863	0.881	0.889	0.899	0.902
LP+PC	0.882	0.901	0.908	0.92	0.924	0.847	0.865	0.872	0.883	0.887	0.864	0.882	0.89	0.901	0.905
LP+SC	0.884	0.901	0.911	0.92	0.926	0.849	0.866	0.875	0.883	0.89	0.866	0.883	0.893	0.901	0.908
LP+PU	0.884	0.902	0.913	0.92	0.927	0.849	0.866	0.877	0.884	0.89	0.866	0.884	0.895	0.901	0.908
PP+PC	0.884	0.904	0.915	0.921	0.928	0.849	0.868	0.879	0.884	0.891	0.866	0.886	0.897	0.902	0.909
PP+SC	0.888	0.905	0.916	0.922	0.933	0.853	0.869	0.88	0.886	0.896	0.87	0.886	0.897	0.903	0.914
PP+PU	0.888	0.905	0.916	0.922	0.933	0.853	0.869	0.88	0.886	0.896	0.87	0.886	0.898	0.904	0.914
PC+SC	0.888	0.906	0.917	0.923	0.937	0.853	0.87	0.881	0.886	0.9	0.87	0.887	0.898	0.904	0.918
PC+PU	0.888	0.906	0.917	0.923	0.937	0.853	0.87	0.881	0.887	0.9	0.87	0.887	0.899	0.904	0.918
SC+PU	0.891	0.906	0.918	0.926	0.937	0.856	0.87	0.881	0.889	0.9	0.873	0.888	0.899	0.907	0.918
LP+PP+PC	0.892	0.907	0.922	0.927	0.939	0.857	0.871	0.885	0.89	0.902	0.874	0.889	0.903	0.908	0.92
LP+PP+SC	0.899	0.908	0.922	0.928	0.939	0.863	0.872	0.886	0.891	0.902	0.881	0.89	0.904	0.909	0.92
LP+PP+PU	0.899	0.909	0.922	0.928	0.939	0.863	0.873	0.886	0.891	0.902	0.881	0.891	0.904	0.909	0.92
LP+PC+SC	0.901	0.909	0.924	0.929	0.94	0.865	0.873	0.888	0.893	0.902	0.883	0.891	0.906	0.911	0.921
LP+PC+PU	0.901	0.91	0.924	0.931	0.94	0.865	0.874	0.888	0.894	0.903	0.883	0.892	0.906	0.913	0.921
LP+SC+PU	0.903	0.912	0.925	0.931	0.94	0.867	0.875	0.888	0.895	0.903	0.885	0.893	0.906	0.913	0.921
PP+PC+SC	0.903	0.914	0.928	0.932	0.941	0.868	0.878	0.891	0.895	0.904	0.885	0.896	0.909	0.913	0.922
PP+PC+PU	0.906	0.915	0.929	0.934	0.941	0.87	0.879	0.892	0.897	0.904	0.888	0.896	0.91	0.916	0.922
PP+SC+PU	0.907	0.916	0.929	0.937	0.944	0.871	0.88	0.893	0.899	0.906	0.889	0.898	0.911	0.918	0.925
PC+SC+PU	0.909	0.917	0.93	0.937	0.945	0.873	0.881	0.893	0.9	0.908	0.891	0.898	0.911	0.918	0.926
LP+PP+PC+SC	0.911	0.918	0.931	0.937	0.947	0.875	0.881	0.894	0.9	0.909	0.893	0.899	0.913	0.918	0.928
LP+PP+PC+PU	0.913	0.921	0.933	0.94	0.947	0.877	0.884	0.896	0.902	0.91	0.894	0.902	0.914	0.921	0.928
LP+PP+SC+PU	0.914	0.921	0.933	0.94	0.947	0.878	0.885	0.897	0.902	0.91	0.896	0.902	0.915	0.921	0.928
LP+PC+SC+PU	0.922	0.922	0.934	0.944	0.95	0.885	0.885	0.897	0.906	0.912	0.903	0.903	0.915	0.925	0.93
PP+PC+SC+PU	0.923	0.929	0.944	0.946	0.954	0.886	0.892	0.907	0.909	0.917	0.904	0.91	0.925	0.927	0.935
LP+PP+PC+SC+PU	0.924	0.936	0.948	0.955	0.961	0.887	0.899	0.911	0.918	0.923	0.905	0.917	0.929	0.936	0.941

acquired by standard classification algorithms have usually been improved with the use of ensemble learning techniques.

Regarding the results of distinct ensemble learning techniques, the random subspace algorithm usually accomplished the best predictive performance. The AdaBoost algorithm usually had the second best predictive performance, and the bagging algorithm usually was third. The greatest predictive achievement among all

Table 7. Classification accuracy values of ensemble learning algorithms.

Feature set	B-KNN	B-LR	B-SVM	B-NB	B-RF	A-KNN	A-LR	A-SVM	A-NB	A-RF	RS-KNN	RS-LR	RS-SVM	RS-NB	RS-RF
TF	84.82	85.27	85.76	88.03	88.11	86.65	84.79	86.51	87.54	88.68	88.75	87.2	88.49	88.75	90.72
LP	84.82	85.5	86.06	88.27	88.42	86.88	85.66	86.56	87.65	88.86	88.98	87.28	88.58	89.35	91.1
PP	84.87	85.69	86.67	89.06	88.88	87.1	86.31	86.94	87.99	89.25	89.06	87.81	89	89.92	91.56
PC	85.35	85.93	86.84	89.33	89.55	87.13	86.37	86.98	88.2	89.53	89.16	88.26	89.03	90.05	91.84
SC	85.58	86.42	87.27	89.45	89.67	87.15	86.48	87.19	88.42	89.83	89.22	88.6	89.43	90.16	92.13
PU	86.41	86.83	87.4	89.34	89.59	87.18	86.49	87.27	88.54	90.1	89.24	88.68	89.52	90.11	92.28
LP+PP	87.56	87.31	87.75	89.51	89.81	87.38	86.62	87.6	88.7	90.57	89.83	88.78	89.8	90.37	92.61
LP+PC	87.6	87.36	88.13	89.74	90.15	87.57	86.71	87.62	88.84	90.73	89.87	89.06	89.97	90.53	92.82
LP+SC	87.61	87.54	88.26	89.59	90.23	87.75	86.77	87.74	89.03	90.8	89.88	89.16	90.11	90.62	92.92
LP+PU	87.73	87.94	88.41	89.68	90.32	88.13	86.95	87.9	89.09	90.97	89.93	89.2	90.23	90.57	93.12
PP+PC	87.77	88.11	88.51	89.86	90.56	88.18	87.09	87.97	89.16	91.08	90.03	89.51	90	90.66	93.27
PP+SC	88	88.74	88.54	89.92	90.96	88.3	87.35	88.13	89.49	91.34	90.14	89.64	90.5	90.95	93.59
PP+PU	88.04	88.8	88.66	90.06	91.1	88.38	87.52	88.54	89.64	91.44	90.21	89.7	90.79	90.76	93.7
PC+SC	88.18	88.88	88.69	89.86	91.24	88.5	87.59	88.55	89.77	91.48	90.27	89.86	90.64	90.73	93.78
PC+PU	88.24	88.94	88.91	89.94	91.28	88.51	87.64	88.6	89.79	91.56	90.46	89.87	90.64	91.04	93.87
SC+PU	88.27	89.08	89.05	90.39	91.46	88.56	87.78	88.71	89.92	91.74	90.48	89.92	90.68	91.31	94.02
LP+PP+PC	88.29	89.23	89.14	90.37	91.54	88.69	88.07	88.8	90.06	91.82	90.6	89.92	90.93	91.25	94.16
LP+PP+SC	88.38	89.33	89.15	90.43	91.48	88.74	88.32	88.82	90.27	91.91	90.71	90.01	91.06	91.21	94.24
LP+PP+PU	88.53	89.47	89.22	90.58	91.65	88.76	88.35	88.86	90.3	92.02	90.76	90.03	90.91	91.65	94.34
LP+PC+SC	88.74	89.75	89.23	90.59	91.57	88.84	88.38	88.89	90.39	92.14	90.82	90.21	91.2	91.38	94.42
LP+PC+PU	89.1	90.05	89.26	90.79	91.61	88.9	88.75	89.03	90.46	92.33	91.1	90.27	91.19	91.66	94.6
LP+SC+PU	89.24	90.06	89.65	90.72	91.54	89.06	88.94	89.06	90.53	92.44	91.2	90.28	91.32	91.69	94.73
PP+PC+SC	89.31	90.1	89.72	90.98	91.9	89.14	88.97	89.1	90.59	92.54	91.53	90.29	91.41	91.98	94.85
PP+PC+PU	89.8	90.27	90.03	91.32	91.97	89.17	88.98	89.41	90.77	92.84	91.57	90.33	91.5	92.2	95.02
PP+SC+PU	89.97	90.42	90.35	91.43	92.13	89.29	89	89.43	90.9	93.01	91.78	90.46	91.63	92.33	95.19
PC+SC+PU	90.05	90.61	90.42	91.31	92.14	89.41	89.15	89.63	91.1	93.1	91.81	90.49	91.75	91.88	95.28
LP+PP+PC+SC	90.09	90.77	90.63	91.45	92.37	89.49	89.38	89.66	91.22	93.23	91.84	91.09	91.79	92.48	95.47
LP+PP+PC+PU	90.48	91.04	91.23	91.51	92.25	89.52	89.41	89.8	91.32	93.52	91.89	91.45	91.82	92.38	95.69
LP+PP+SC+PU	90.54	91.09	91.87	91.52	92.26	89.77	89.42	89.86	91.4	93.68	91.89	91.91	92.07	92.44	95.87
LP+PC+SC+PU	90.98	91.52	92.26	92.2	92.76	90.06	89.43	89.92	91.74	94.14	92.75	92.03	92.1	93.02	96.22
PP+PC+SC+PU	91.21	91.56	92.33	92.4	93.21	90.08	89.47	90.34	92.06	94.31	93.34	92.8	92.64	93.39	96.37
LP+PP+PC+SC+PU	93.99	91.71	93.05	93.06	93.6	90.28	90.04	90.42	92.33	95.23	93.49	94.11	92.57	93.72	96.92

comparative settings was achieved through the ensemble feature set that integrates five LIWC-based feature sets, namely language procedures, psychological procedures, private issues, spoken categories, and punctuation, in combination with the random subspace ensemble of random forest. A classification accuracy of 96.92 percent was achieved for this setup. The findings of predictive performance mentioned in Tables 8–11 show comparable trends to the outcomes of performance shown in Table 7.

Table 8. F-measure values of ensemble learning algorithms.

Feature set	B-KNN	B-LR	B-SVM	B-NB	B-RF	A-KNN	A-LR	A-SVM	A-NB	A-RF	RS-KNN	RS-LR	RS-SVM	RS-NB	RS-RF
TF	0.857	0.861	0.866	0.889	0.89	0.875	0.856	0.874	0.884	0.896	0.897	0.881	0.894	0.896	0.916
LP	0.857	0.864	0.869	0.892	0.893	0.878	0.865	0.874	0.885	0.898	0.899	0.882	0.895	0.903	0.92
PP	0.857	0.866	0.875	0.9	0.898	0.88	0.872	0.878	0.889	0.902	0.9	0.887	0.899	0.908	0.925
PC	0.862	0.868	0.877	0.902	0.905	0.88	0.872	0.879	0.891	0.904	0.901	0.891	0.899	0.91	0.928
SC	0.864	0.873	0.882	0.904	0.906	0.88	0.874	0.881	0.893	0.907	0.901	0.895	0.903	0.911	0.931
PU	0.873	0.877	0.883	0.902	0.905	0.881	0.874	0.881	0.894	0.91	0.901	0.896	0.904	0.91	0.932
LP+PP	0.884	0.882	0.886	0.904	0.907	0.883	0.875	0.885	0.896	0.915	0.907	0.897	0.907	0.913	0.935
LP+PC	0.885	0.882	0.89	0.906	0.911	0.885	0.876	0.885	0.897	0.917	0.908	0.9	0.909	0.914	0.938
LP+SC	0.885	0.884	0.892	0.905	0.911	0.886	0.876	0.886	0.899	0.917	0.908	0.901	0.91	0.915	0.939
LP+PU	0.886	0.888	0.893	0.906	0.912	0.89	0.878	0.888	0.9	0.919	0.908	0.901	0.911	0.915	0.941
PP+PC	0.887	0.89	0.894	0.908	0.915	0.891	0.88	0.889	0.901	0.92	0.909	0.904	0.909	0.916	0.942
PP+SC	0.889	0.896	0.894	0.908	0.919	0.892	0.882	0.89	0.904	0.923	0.911	0.905	0.914	0.919	0.945
PP+PU	0.889	0.897	0.896	0.91	0.92	0.893	0.884	0.894	0.905	0.924	0.911	0.906	0.917	0.917	0.947
PC+SC	0.891	0.898	0.896	0.908	0.922	0.894	0.885	0.894	0.907	0.924	0.912	0.908	0.916	0.916	0.947
PC+PU	0.891	0.898	0.898	0.908	0.922	0.894	0.885	0.895	0.907	0.925	0.914	0.908	0.916	0.92	0.948
SC+PU	0.892	0.9	0.9	0.913	0.924	0.895	0.887	0.896	0.908	0.927	0.914	0.908	0.916	0.922	0.95
LP+PP+PC	0.892	0.901	0.9	0.913	0.925	0.896	0.89	0.897	0.91	0.927	0.915	0.908	0.918	0.922	0.951
LP+PP+SC	0.893	0.902	0.901	0.913	0.924	0.896	0.892	0.897	0.912	0.928	0.916	0.909	0.92	0.921	0.952
LP+PP+PU	0.894	0.904	0.901	0.915	0.926	0.897	0.892	0.898	0.912	0.929	0.917	0.909	0.918	0.926	0.953
LP+PC+SC	0.896	0.907	0.901	0.915	0.925	0.897	0.893	0.898	0.913	0.931	0.917	0.911	0.921	0.923	0.954
LP+PC+PU	0.9	0.91	0.902	0.917	0.925	0.898	0.897	0.899	0.914	0.933	0.92	0.912	0.921	0.926	0.956
LP+SC+PU	0.901	0.91	0.906	0.916	0.925	0.9	0.898	0.9	0.914	0.934	0.921	0.912	0.922	0.926	0.957
PP+PC+SC	0.902	0.91	0.906	0.919	0.928	0.9	0.899	0.9	0.915	0.935	0.924	0.912	0.923	0.929	0.958
PP+PC+PU	0.907	0.912	0.909	0.922	0.929	0.901	0.899	0.903	0.917	0.938	0.925	0.912	0.924	0.931	0.96
PP+SC+PU	0.909	0.913	0.913	0.924	0.931	0.902	0.899	0.903	0.918	0.94	0.927	0.914	0.926	0.933	0.961
PC+SC+PU	0.91	0.915	0.913	0.922	0.931	0.903	0.9	0.905	0.92	0.94	0.927	0.914	0.927	0.928	0.962
LP+PP+PC+SC	0.91	0.917	0.915	0.924	0.933	0.904	0.903	0.906	0.921	0.942	0.928	0.92	0.927	0.934	0.964
LP+PP+PC+PU	0.914	0.92	0.921	0.924	0.932	0.904	0.903	0.907	0.922	0.945	0.928	0.924	0.927	0.933	0.967
LP+PP+SC+PU	0.915	0.92	0.928	0.924	0.932	0.907	0.903	0.908	0.923	0.946	0.928	0.928	0.93	0.934	0.968
LP+PC+SC+PU	0.919	0.924	0.932	0.931	0.937	0.91	0.903	0.908	0.927	0.951	0.937	0.93	0.93	0.94	0.972
PP+PC+SC+PU	0.921	0.925	0.933	0.933	0.941	0.91	0.904	0.913	0.93	0.953	0.943	0.937	0.936	0.943	0.973
LP+PP+PC+SC+PU	0.949	0.926	0.94	0.94	0.945	0.912	0.909	0.913	0.933	0.962	0.944	0.951	0.935	0.947	0.979

To summarize the primary results of the empirical analysis, we describe in Figure 3 and Figure 4, respectively, the main effect plots for precision and main effect plots for F-measure values.

In addition to the predictive performance of conventional classifiers, ensemble learners, and feature sets, we have also considered the predictive performance of deep learning architectures on satire identification in Turkish. In Table 12, classification accuracies and AUC values obtained by five deep learning architectures

Table 9. Precision values of ensemble learning algorithms.

Feature set	B-KNN	B-LR	B-SVM	B-NB	B-RF	A-KNN	A-LR	A-SVM	A-NB	A-RF	RS-KNN	RS-LR	RS-SVM	RS-NB	RS-RF
TF	0.874	0.879	0.884	0.908	0.908	0.893	0.874	0.892	0.902	0.914	0.915	0.899	0.912	0.915	0.935
LP	0.874	0.881	0.887	0.91	0.912	0.896	0.883	0.892	0.904	0.916	0.917	0.9	0.913	0.921	0.939
PP	0.875	0.883	0.893	0.918	0.916	0.898	0.89	0.896	0.907	0.92	0.918	0.905	0.918	0.927	0.944
PC	0.88	0.886	0.895	0.921	0.923	0.898	0.89	0.897	0.909	0.923	0.919	0.91	0.918	0.928	0.947
SC	0.882	0.891	0.9	0.922	0.924	0.898	0.892	0.899	0.912	0.926	0.92	0.913	0.922	0.93	0.95
PU	0.891	0.895	0.901	0.921	0.924	0.899	0.892	0.9	0.913	0.929	0.92	0.914	0.923	0.929	0.951
LP+PP	0.903	0.9	0.905	0.923	0.926	0.901	0.893	0.903	0.914	0.934	0.926	0.915	0.926	0.932	0.955
LP+PC	0.903	0.901	0.909	0.925	0.929	0.903	0.894	0.903	0.916	0.935	0.926	0.918	0.928	0.933	0.957
LP+SC	0.903	0.902	0.91	0.924	0.93	0.905	0.894	0.905	0.918	0.936	0.927	0.919	0.929	0.934	0.958
LP+PU	0.904	0.907	0.911	0.925	0.931	0.909	0.896	0.906	0.918	0.938	0.927	0.92	0.93	0.934	0.96
PP+PC	0.905	0.908	0.913	0.926	0.934	0.909	0.898	0.907	0.919	0.939	0.928	0.923	0.928	0.935	0.962
PP+SC	0.907	0.915	0.913	0.927	0.938	0.91	0.901	0.909	0.923	0.942	0.929	0.924	0.933	0.938	0.965
PP+PU	0.908	0.916	0.914	0.928	0.939	0.911	0.902	0.913	0.924	0.943	0.93	0.925	0.936	0.936	0.966
PC+SC	0.909	0.916	0.914	0.926	0.941	0.912	0.903	0.913	0.925	0.943	0.931	0.926	0.934	0.935	0.967
PC+PU	0.91	0.917	0.917	0.927	0.941	0.912	0.904	0.913	0.926	0.944	0.933	0.927	0.934	0.939	0.968
SC+PU	0.91	0.918	0.918	0.932	0.943	0.913	0.905	0.915	0.927	0.946	0.933	0.927	0.935	0.941	0.969
LP+PP+PC	0.91	0.92	0.919	0.932	0.944	0.914	0.908	0.915	0.928	0.947	0.934	0.927	0.937	0.941	0.971
LP+PP+SC	0.911	0.921	0.919	0.932	0.943	0.915	0.91	0.916	0.931	0.948	0.935	0.928	0.939	0.94	0.972
LP+PP+PU	0.913	0.922	0.92	0.934	0.945	0.915	0.911	0.916	0.931	0.949	0.936	0.928	0.937	0.945	0.973
LP+PC+SC	0.915	0.925	0.92	0.934	0.944	0.916	0.911	0.916	0.932	0.95	0.936	0.93	0.94	0.942	0.973
LP+PC+PU	0.919	0.928	0.92	0.936	0.944	0.916	0.915	0.918	0.933	0.952	0.939	0.931	0.94	0.945	0.975
LP+SC+PU	0.92	0.928	0.924	0.935	0.944	0.918	0.917	0.918	0.933	0.953	0.94	0.931	0.941	0.945	0.977
PP+PC+SC	0.921	0.929	0.925	0.938	0.947	0.919	0.917	0.919	0.934	0.954	0.944	0.931	0.942	0.948	0.978
PP+PC+PU	0.926	0.931	0.928	0.941	0.948	0.919	0.917	0.922	0.936	0.957	0.944	0.931	0.943	0.951	0.98
PP+SC+PU	0.927	0.932	0.931	0.943	0.95	0.921	0.918	0.922	0.937	0.959	0.946	0.933	0.945	0.952	0.981
PC+SC+PU	0.928	0.934	0.932	0.941	0.95	0.922	0.919	0.924	0.939	0.96	0.946	0.933	0.946	0.947	0.982
LP+PP+PC +SC	0.929	0.936	0.934	0.943	0.952	0.923	0.921	0.924	0.94	0.961	0.947	0.939	0.946	0.953	0.984
LP+PP+PC +PU	0.933	0.939	0.94	0.943	0.951	0.923	0.922	0.926	0.941	0.964	0.947	0.943	0.947	0.952	0.986
LP+PP+SC +PU	0.933	0.939	0.947	0.944	0.951	0.925	0.922	0.926	0.942	0.966	0.947	0.947	0.949	0.953	0.988
LP+PC+SC +PU	0.938	0.944	0.951	0.951	0.956	0.928	0.922	0.927	0.946	0.971	0.956	0.949	0.95	0.959	0.992
PP+PC+SC +PU	0.94	0.944	0.952	0.953	0.961	0.929	0.922	0.931	0.949	0.972	0.962	0.957	0.955	0.963	0.994
LP+PP+PC +SC+PU	0.969	0.945	0.959	0.959	0.965	0.931	0.928	0.932	0.952	0.982	0.964	0.97	0.954	0.966	0.999

(convolutional neural network, recurrent neural network, long short-term memory, gated recurrent unit, and recurrent neural network with attention mechanism) on three word-embedding schemes have been presented.

The predictive performance results listed in Tables 12 and 13 indicate that the GloVe word embedding scheme yields higher predictive performance compared to the other word embedding schemes. The second highest predictive performances have been obtained by the fastText CBOW model, which is followed by the

Table 10. Recall values of ensemble learning algorithms.

Feature set	B-KNN	B-LR	B-SVM	B-NB	B-RF	A-KNN	A-LR	A-SVM	A-NB	A-RF	RS-KNN	RS-LR	RS-SVM	RS-NB	RS-RF
TF	0.84	0.844	0.849	0.872	0.872	0.858	0.839	0.857	0.867	0.878	0.879	0.863	0.876	0.879	0.898
LP	0.84	0.846	0.852	0.874	0.875	0.86	0.848	0.857	0.868	0.88	0.881	0.864	0.877	0.885	0.902
PP	0.84	0.848	0.858	0.882	0.88	0.862	0.855	0.861	0.871	0.884	0.882	0.869	0.881	0.89	0.907
PC	0.845	0.851	0.86	0.884	0.887	0.863	0.855	0.861	0.873	0.886	0.883	0.874	0.881	0.892	0.909
SC	0.847	0.856	0.864	0.886	0.888	0.863	0.856	0.863	0.875	0.889	0.883	0.877	0.885	0.893	0.912
PU	0.856	0.86	0.865	0.885	0.887	0.863	0.856	0.864	0.877	0.892	0.884	0.878	0.886	0.892	0.914
LP+PP	0.867	0.864	0.869	0.886	0.889	0.865	0.858	0.867	0.878	0.897	0.889	0.879	0.889	0.895	0.917
LP+PC	0.867	0.865	0.873	0.889	0.893	0.867	0.859	0.868	0.88	0.898	0.89	0.882	0.891	0.896	0.919
LP+SC	0.867	0.867	0.874	0.887	0.893	0.869	0.859	0.869	0.881	0.899	0.89	0.883	0.892	0.897	0.92
LP+PU	0.869	0.871	0.875	0.888	0.894	0.873	0.861	0.87	0.882	0.901	0.89	0.883	0.893	0.897	0.922
PP+PC	0.869	0.872	0.876	0.89	0.897	0.873	0.862	0.871	0.883	0.902	0.891	0.886	0.891	0.898	0.923
PP+SC	0.871	0.879	0.877	0.89	0.901	0.874	0.865	0.873	0.886	0.904	0.892	0.888	0.896	0.9	0.927
PP+PU	0.872	0.879	0.878	0.892	0.902	0.875	0.866	0.877	0.887	0.905	0.893	0.888	0.899	0.899	0.928
PC+SC	0.873	0.88	0.878	0.89	0.903	0.876	0.867	0.877	0.889	0.906	0.894	0.89	0.897	0.898	0.928
PC+PU	0.874	0.881	0.88	0.89	0.904	0.876	0.868	0.877	0.889	0.907	0.896	0.89	0.897	0.901	0.929
SC+PU	0.874	0.882	0.882	0.895	0.906	0.877	0.869	0.878	0.89	0.908	0.896	0.89	0.898	0.904	0.931
LP+PP+PC	0.874	0.884	0.883	0.895	0.906	0.878	0.872	0.879	0.892	0.909	0.897	0.89	0.9	0.903	0.932
LP+PP+SC	0.875	0.884	0.883	0.895	0.906	0.879	0.874	0.879	0.894	0.91	0.898	0.891	0.902	0.903	0.933
LP+PP+PU	0.877	0.886	0.883	0.897	0.907	0.879	0.875	0.88	0.894	0.911	0.899	0.891	0.9	0.907	0.934
LP+PC+SC	0.879	0.889	0.883	0.897	0.907	0.88	0.875	0.88	0.895	0.912	0.899	0.893	0.903	0.905	0.935
LP+PC+PU	0.882	0.892	0.884	0.899	0.907	0.88	0.879	0.882	0.896	0.914	0.902	0.894	0.903	0.908	0.937
LP+SC+PU	0.884	0.892	0.888	0.898	0.906	0.882	0.881	0.882	0.896	0.915	0.903	0.894	0.904	0.908	0.938
PP+PC+SC	0.884	0.892	0.888	0.901	0.91	0.883	0.881	0.882	0.897	0.916	0.906	0.894	0.905	0.911	0.939
PP+PC+PU	0.889	0.894	0.891	0.904	0.911	0.883	0.881	0.885	0.899	0.919	0.907	0.894	0.906	0.913	0.941
PP+SC+PU	0.891	0.895	0.895	0.905	0.912	0.884	0.881	0.885	0.9	0.921	0.909	0.896	0.907	0.914	0.942
PC+SC+PU	0.892	0.897	0.895	0.904	0.912	0.885	0.883	0.887	0.902	0.922	0.909	0.896	0.908	0.91	0.943
LP+PP+PC +SC	0.892	0.899	0.897	0.905	0.915	0.886	0.885	0.888	0.903	0.923	0.909	0.902	0.909	0.916	0.945
LP+PP+PC +PU	0.896	0.901	0.903	0.906	0.913	0.886	0.885	0.889	0.904	0.926	0.91	0.905	0.909	0.915	0.947
LP+PP+SC +PU	0.896	0.902	0.91	0.906	0.913	0.889	0.885	0.89	0.905	0.928	0.91	0.91	0.912	0.915	0.949
LP+PC+SC +PU	0.901	0.906	0.913	0.913	0.918	0.892	0.885	0.89	0.908	0.932	0.918	0.911	0.912	0.921	0.953
PP+PC+SC +PU	0.903	0.906	0.914	0.915	0.923	0.892	0.886	0.894	0.912	0.934	0.924	0.919	0.917	0.925	0.954
LP+PP+PC +SC+PU	0.931	0.908	0.921	0.921	0.927	0.894	0.891	0.895	0.914	0.943	0.926	0.932	0.917	0.928	0.96

fastText skip-gram model. The lowest predictive performances in terms of classification accuracies have been obtained by word2vec skip-gram model. For the different vector sizes and dimensions of projection layers considered in the empirical analysis, vector size of 300 and dimension projection layer of 300 yield higher predictive performance.

Regarding the predictive performances of deep-learning-based architectures for sentiment analysis, the highest predictive performances have been obtained by the recurrent neural network with attention mechanism

Table 11. AUC values of ensemble learning algorithms.

Feature set	B-KNN	B-LR	B-SVM	B-NB	B-RF	A-KNN	A-LR	A-SVM	A-NB	A-RF	RS-KNN	RS-LR	RS-SVM	RS-NB	RS-RF
TF	0.853	0.858	0.863	0.886	0.887	0.872	0.853	0.87	0.881	0.893	0.893	0.877	0.891	0.893	0.914
LP	0.853	0.86	0.866	0.888	0.89	0.874	0.862	0.871	0.882	0.895	0.896	0.878	0.892	0.9	0.918
PP	0.853	0.862	0.872	0.897	0.895	0.876	0.868	0.875	0.886	0.899	0.897	0.884	0.896	0.905	0.922
PC	0.858	0.864	0.874	0.899	0.902	0.877	0.869	0.875	0.888	0.901	0.898	0.888	0.896	0.907	0.925
SC	0.861	0.869	0.878	0.901	0.903	0.877	0.87	0.877	0.89	0.904	0.898	0.892	0.9	0.908	0.928
PU	0.869	0.873	0.879	0.899	0.902	0.877	0.87	0.878	0.891	0.907	0.898	0.893	0.901	0.907	0.93
LP+PP	0.881	0.878	0.883	0.901	0.904	0.879	0.871	0.881	0.893	0.912	0.904	0.894	0.904	0.91	0.933
LP+PC	0.881	0.879	0.887	0.903	0.908	0.881	0.872	0.882	0.894	0.914	0.905	0.896	0.906	0.912	0.935
LP+SC	0.881	0.881	0.888	0.902	0.908	0.883	0.873	0.883	0.896	0.914	0.905	0.897	0.907	0.913	0.936
LP+PU	0.883	0.885	0.89	0.903	0.909	0.887	0.875	0.884	0.897	0.916	0.905	0.898	0.908	0.912	0.938
PP+PC	0.883	0.887	0.891	0.905	0.912	0.887	0.876	0.885	0.897	0.917	0.906	0.901	0.906	0.913	0.94
PP+SC	0.885	0.893	0.891	0.905	0.916	0.889	0.879	0.887	0.901	0.92	0.908	0.902	0.911	0.916	0.943
PP+PU	0.886	0.894	0.892	0.907	0.917	0.889	0.88	0.891	0.902	0.921	0.908	0.903	0.914	0.914	0.944
PC+SC	0.887	0.894	0.893	0.905	0.919	0.891	0.881	0.891	0.904	0.921	0.909	0.905	0.913	0.914	0.945
PC+PU	0.888	0.895	0.895	0.905	0.919	0.891	0.882	0.892	0.904	0.922	0.911	0.905	0.913	0.917	0.946
SC+PU	0.888	0.896	0.896	0.91	0.921	0.891	0.883	0.893	0.905	0.924	0.911	0.905	0.913	0.919	0.947
LP+PP+PC	0.888	0.898	0.897	0.91	0.922	0.892	0.886	0.894	0.907	0.925	0.912	0.905	0.915	0.919	0.949
LP+PP+SC	0.889	0.899	0.897	0.91	0.921	0.893	0.889	0.894	0.909	0.926	0.913	0.906	0.917	0.918	0.95
LP+PP+PU	0.891	0.9	0.898	0.912	0.923	0.893	0.889	0.894	0.909	0.927	0.914	0.906	0.915	0.923	0.951
LP+PC+SC	0.893	0.903	0.898	0.912	0.922	0.894	0.889	0.894	0.91	0.928	0.914	0.908	0.918	0.92	0.951
LP+PC+PU	0.897	0.906	0.898	0.914	0.923	0.895	0.893	0.896	0.911	0.93	0.917	0.909	0.918	0.923	0.953
LP+SC+PU	0.898	0.907	0.902	0.913	0.922	0.896	0.895	0.896	0.911	0.931	0.918	0.909	0.919	0.923	0.955
PP+PC+SC	0.899	0.907	0.903	0.916	0.925	0.897	0.895	0.897	0.912	0.932	0.922	0.909	0.92	0.926	0.956
PP+PC+PU	0.904	0.909	0.906	0.92	0.926	0.897	0.895	0.9	0.914	0.935	0.922	0.909	0.921	0.929	0.958
PP+SC+PU	0.906	0.91	0.909	0.921	0.928	0.899	0.896	0.9	0.915	0.937	0.924	0.911	0.923	0.93	0.959
PC+SC+PU	0.906	0.912	0.91	0.919	0.928	0.9	0.897	0.902	0.917	0.938	0.924	0.911	0.924	0.925	0.96
LP+PP+PC +SC	0.907	0.914	0.912	0.921	0.93	0.901	0.899	0.902	0.918	0.939	0.925	0.917	0.924	0.931	0.962
LP+PP+PC +PU	0.911	0.916	0.918	0.921	0.929	0.901	0.9	0.904	0.919	0.942	0.925	0.921	0.924	0.93	0.964
LP+PP+SC +PU	0.911	0.917	0.925	0.921	0.929	0.903	0.9	0.904	0.92	0.944	0.925	0.925	0.927	0.931	0.966
LP+PC+SC +PU	0.916	0.921	0.929	0.928	0.934	0.906	0.9	0.905	0.924	0.948	0.934	0.927	0.927	0.937	0.97
PP+PC+SC +PU	0.918	0.922	0.93	0.93	0.939	0.906	0.9	0.909	0.927	0.95	0.94	0.934	0.933	0.941	0.971
LP+PP+PC +SC+PU	0.947	0.923	0.937	0.937	0.943	0.908	0.906	0.91	0.929	0.959	0.941	0.948	0.932	0.944	0.977

(RNN-AM), the second highest predictive performances have been achieved by gated recurrent units (GRU), and the third highest predictive performances have been obtained by the long short-term memory networks (LSTM). For all the compared configurations, the highest predictive performance has been achieved by RNN-AM in conjunction with the Glove word embedding scheme with a classification accuracy of 97.72%.

To further evaluate the empirical results, we have conducted a two-way ANOVA test on the Minitab statistical toolkit. The findings of the ANOVA test are provided in Table 14. In this table, respectively,

Table 12. Classification accuracy and AUC values of deep learning architectures.

			Classification accuracy values					AUC value				
Word Embedding	Vector size	Dimension of projection layer	CNN	RNN	LSTM	GRU	RNN-AM	CNN	RNN	LSTM	GRU	RNN-AM
word2vec (Skip-gram)	200	100	84.22	86.7	88.98	91.35	93.9	0.848	0.874	0.897	0.922	0.948
word2vec (Skip-gram)	200	200	84.33	86.83	89.43	91.42	94.06	0.849	0.875	0.902	0.922	0.95
word2vec (Skip-gram)	300	100	84.57	87.27	89.46	91.44	94.12	0.852	0.88	0.902	0.923	0.95
word2vec (Skip-gram)	300	300	84.58	87.39	89.5	91.8	94.23	0.852	0.881	0.903	0.926	0.951
word2vec (CBOW)	200	100	84.84	87.51	89.67	91.9	94.24	0.855	0.882	0.904	0.927	0.952
word2vec (CBOW)	200	200	85.02	87.55	89.84	92.11	94.28	0.856	0.883	0.906	0.93	0.952
word2vec (CBOW)	300	100	85.11	87.7	89.9	92.43	94.32	0.857	0.884	0.907	0.933	0.952
word2vec (CBOW)	300	300	85.29	88	90.19	92.45	94.53	0.859	0.887	0.91	0.933	0.955
fastText (Skip-gram)	200	100	85.3	88.01	90.3	92.45	94.85	0.859	0.887	0.911	0.933	0.958
fastText (Skip-gram)	200	200	85.54	88.28	90.34	92.77	95.05	0.862	0.89	0.911	0.936	0.96
fastText (Skip-gram)	300	100	85.72	88.39	90.35	92.9	95.24	0.864	0.891	0.911	0.938	0.962
fastText (Skip-gram)	300	300	85.9	88.43	90.52	93.19	95.83	0.866	0.892	0.913	0.941	0.968
fastText (CBOW)	200	100	85.94	88.5	90.62	93.38	96.12	0.866	0.892	0.914	0.943	0.971
fastText (CBOW)	200	200	86.01	88.56	90.67	93.43	96.28	0.867	0.893	0.915	0.943	0.973
fastText (CBOW)	300	100	86.26	88.61	90.7	93.47	96.55	0.869	0.894	0.915	0.944	0.975
fastText (CBOW)	300	300	86.31	88.64	90.77	93.53	96.59	0.87	0.894	0.916	0.944	0.976
GloVe	200	100	86.32	88.67	90.83	93.57	96.67	0.87	0.894	0.916	0.945	0.977
GloVe	200	200	86.45	88.78	90.88	93.61	96.96	0.871	0.895	0.917	0.945	0.98
GloVe	300	100	86.49	88.92	90.88	93.72	97.3	0.872	0.897	0.917	0.946	0.983
GloVe	300	300	86.54	88.97	91.03	93.77	97.72	0.872	0.897	0.918	0.947	0.987

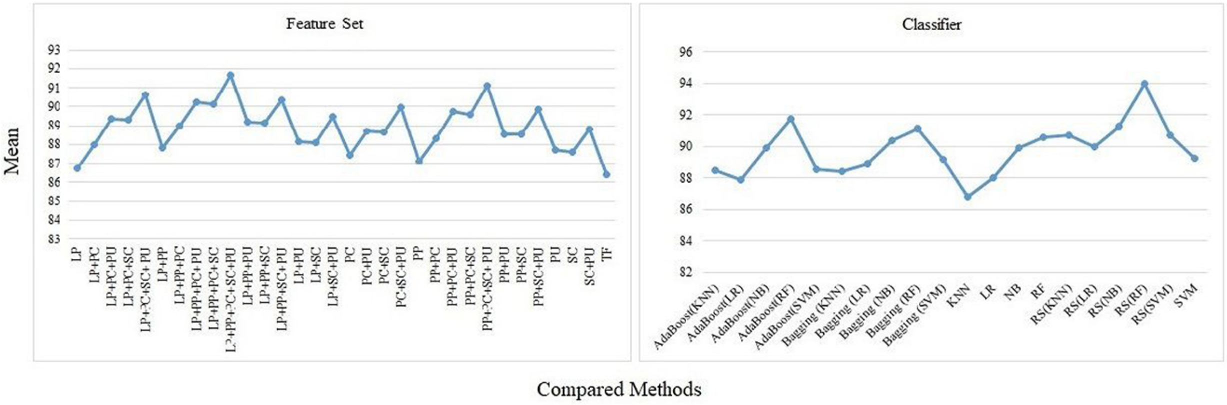

Figure 3. The main effects plot for classification accuracy values.

DF, SS, MS, F, and P denote degrees of freedom, adjusted square amount, adjusted mean square value, f-value, and probability value. There are statistically significant variations between the predictive performance of comparative feature sets and classification algorithms (P <0.0001) according to two-way ANOVA test outcomes.

Table 13. Precision and recall values of deep learning architectures.

			Precision					Recall				
Word Embedding	Vector size	Dimension of projection layer	CNN	RNN	LSTM	GRU	RNN-AM	CNN	RNN	LSTM	GRU	RNN-AM
word2vec (Skip-gram)	200	100	0.868	0.894	0.917	0.942	0.968	0.834	0.858	0.881	0.904	0.93
word2vec (Skip-gram)	200	200	0.869	0.895	0.922	0.942	0.97	0.835	0.86	0.885	0.905	0.931
word2vec (Skip-gram)	300	100	0.872	0.9	0.922	0.943	0.97	0.837	0.864	0.886	0.905	0.932
word2vec (Skip-gram)	300	300	0.872	0.901	0.923	0.946	0.971	0.837	0.865	0.886	0.909	0.933
word2vec (CBOW)	200	100	0.875	0.902	0.924	0.947	0.972	0.84	0.866	0.888	0.91	0.933
word2vec (CBOW)	200	200	0.876	0.903	0.926	0.95	0.972	0.842	0.867	0.89	0.912	0.933
word2vec (CBOW)	300	100	0.877	0.904	0.927	0.953	0.972	0.843	0.868	0.89	0.915	0.934
word2vec (CBOW)	300	300	0.879	0.907	0.93	0.953	0.975	0.844	0.871	0.893	0.915	0.936
fastText (Skip-gram)	200	100	0.879	0.907	0.931	0.953	0.978	0.845	0.871	0.894	0.915	0.939
fastText (Skip-gram)	200	200	0.882	0.91	0.931	0.956	0.98	0.847	0.874	0.894	0.919	0.941
fastText (Skip-gram)	300	100	0.884	0.911	0.931	0.958	0.982	0.849	0.875	0.895	0.92	0.943
fastText (Skip-gram)	300	300	0.886	0.912	0.933	0.961	0.988	0.85	0.876	0.896	0.923	0.949
fastText (CBOW)	200	100	0.886	0.912	0.934	0.963	0.981	0.851	0.876	0.897	0.925	0.961
fastText (CBOW)	200	200	0.887	0.913	0.935	0.963	0.983	0.852	0.877	0.898	0.925	0.963
fastText (CBOW)	300	100	0.889	0.914	0.935	0.964	0.985	0.854	0.877	0.898	0.925	0.965
fastText (CBOW)	300	300	0.89	0.914	0.936	0.964	0.986	0.855	0.878	0.899	0.926	0.966
GloVe	200	100	0.89	0.914	0.936	0.965	0.987	0.855	0.878	0.899	0.926	0.967
GloVe	200	200	0.891	0.915	0.937	0.965	0.99	0.856	0.879	0.9	0.927	0.969
GloVe	300	100	0.892	0.917	0.937	0.966	0.983	0.856	0.88	0.9	0.928	0.983
GloVe	300	300	0.892	0.917	0.938	0.967	0.987	0.857	0.881	0.901	0.928	0.987

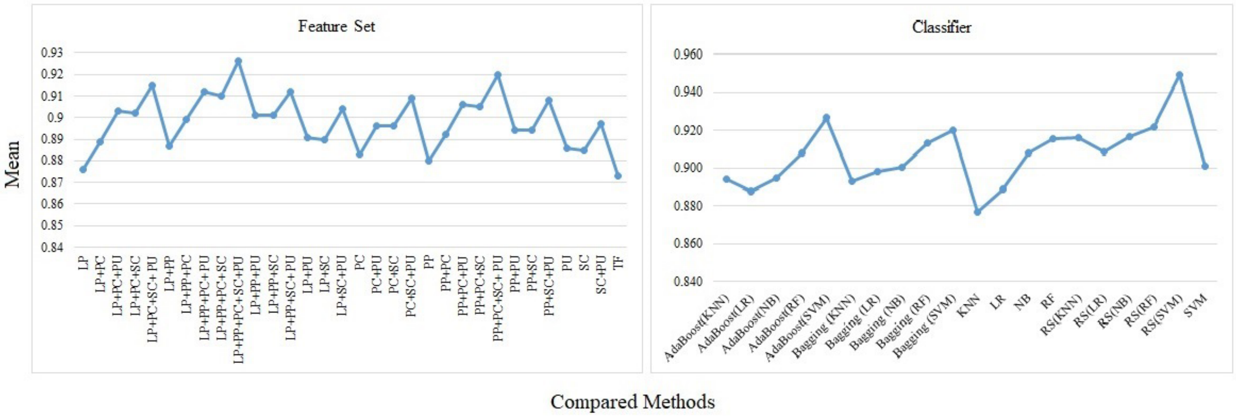

Figure 4. The main effects plot for F-measure values.

5. Conclusion

Figurative language, which contains elements such as metaphor, analogy, ambiguity, irony, and satire, can be frequently encountered on social media and microblogging platforms. The automatic identification of figurative language can be regarded as a challenging task in natural language processing. In this paper, we present a machine learning-based approach for satirical text identification in Turkish. The contributions of this paper are

Table 14. Two-way ANOVA test results for classification accuracy.

Source	DF	SS	MS	F-value	P-value
Feature set	36	1206.22	33.5061	8.15	0
Lack-of-fit error	628	2887.00	4.5971	95.01	0
Total	739	4996.85			

threefold. First, we have collected a corpus of news articles in Turkish for satire identification. Second, we have performed an extensive empirical analysis of different psychological and linguistic feature set dimensions. Third, we have experimented on the use of ensemble learning methods in conjunction with ensemble feature subsets and classifiers. The comprehensive analysis on different feature sets, classifiers, and ensemble methods indicated that the utilization of psychological and linguistic feature sets in conjunction with ensemble learners can yield encouraging predictive results for satire detection in Turkish. We have achieved a classification accuracy of 96.92% with the use of ensemble feature subsets and the random subspace ensemble of the random forest algorithm.

In addition to conventional classifiers and feature sets, deep-learning architectures have been also evaluated for satire identification. The experimental results indicate that deep-learning-based architectures can yield promising results on satire detection. We have obtained a classification accuracy of 97.72% with the recurrent neural network architecture with attention mechanism with the use of the GloVe-based word embedding scheme.

It should be beneficial to extend this study in several dimensions. First, we consider extending the feature subsets by taking other linguistic feature subsets into account. The experimental analysis indicates that deep-learning methods can yield higher predictive performances on satire identification. Hence, it should be beneficial to extend this study by proposing a novel deep-learning-based architecture. In addition, the predictive performance of the empirical schemes can be adopted to other languages in order to draw conclusions regarding different and/or similar patterns in different languages.

References

[1] Cambria E. Affective computing and sentiment analysis. IEEE Intelligent Systems 2016; 31: 102-107.

[2] Onan A. Deep learning based sentiment analysis on product reviews on Twitter. In: The International Conference on Big Data Innovations and Applications; Istanbul, Turkey; 2019. pp. 80-91.

[3] Toçoğlu MA, Onan A. Satire detection in Turkish news articles: a machine learning approach. In: The International Conference on Big Data Innovations and Applications; Istanbul, Turkey; 2019. pp. 107-117.

[4] Poria S, Cambria E, Hazarika D, Vij P. A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv preprint arXiv:1610.08815, 2016.

[5] Davidov D, Tsur O, Rappoport A. Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. In: The 14th Conference on Computational Natural Language Learning; Uppsala, Sweden; 2010. pp. 107-116.

[6] González-Ibánez R, Muresan S, Wacholder N. Identifying sarcasm in Twitter: a closer look. In: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers; Portland, OR, USA; 2011. pp. 581-586.

[7] Filatova E. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In: The 8th International Conference on Language Resources and Evaluation (LREC); Istanbul, Turkey; 2012. pp. 392-398.

[8] Del Pilar Salas-Zárate M, Paredes-Valverde MA, Rodriguez-Garcia MA, Valencia-García R, Alor-Hernández G. Automatic detection of satire in Twitter: A psycholinguistic-based approach. Knowledge-Based Systems 2017; 128: 20-33.

[9] Francis ME, Pennebaker JW. LIWC: Linguistic Inquiry and Word Count. Dallas, TX, USA: Southern Methodist University, 1993.

[10] Brubaker JR, Kivran-Swaine F, Taber L, Hayes GR. Grief-Stricken in a crowd: The language of bereavement and distress in social media. In: 6th International AAAI Conference on Weblogs and Social Media; Dublin, Ireland; 2012. pp. 42-49.

[11] Rubin VL, Conroy NJ, Chen Y, Cornwell S. Fake news or truth? Using satirical cues to detect potentially misleading news. In: The Workshop on Computational Approaches to Deception Detection at the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-CADD2016); San Diego, CA, USA; 2016. pp. 7–17.

[12] Justo R, Corcoran T, Lukin SM, Walker M, Torres MI. Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web. Knowledge-Based Systems 2014; 69: 124–133.

[13] Ahmad T, Akhtar H, Chopra A, Akhtar MW. Satire detection from web documents using machine learning methods. In: The International Conference on Soft Computing and Machine Intelligence; New Delhi, India; 2014. pp. 102-105.

[14] Barbieri F, Ronzano F, Saggion H. Is this Tweet satirical? A computational approach for satire detection in Spanish. Procesamiento del Lenguaje Natural 2015; 55: 135-142.

[15] Barbieri F, Ronzano F, Saggion H. Do we criticise (and laugh) in the same way? Automatic detection of multi-lingual satirical news in Twitter. In: The 24th International Joint Conference on Artificial Intelligence; Buenos Aires, Argentina; 2015. pp. 1215-1221.

[16] Perez-Rosas V, Kleinberg B, Lefevre A, Mihalcea R. Automatic detection of fake news. arXiv preprint arXiv:1708.07104, 2017.

[17] Ravi K, Ravi V. A novel automatic satire and irony detection using ensembled feature selection and data mining. Knowledge-Based Systems 2017; 120: 15-33.

[18] Ahmed H, Traore I, Saad S. Detection of online fake news using n-gram analysis and machine learning techniques. In: International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments; Vancouver, BC, Canada; 2017. pp. 127-138.

[19] Yang F, Mukherjee A, Dragut E. Satirical news detection and analysis using attention mechanism and linguistic features. arXiv preprint arXiv:1709.01189, 2017.

[20] Ravi K, Ravi V. Irony detection using neural network language model, psycholinguistic features and text mining. In: The 17th International Conference on Cognitive Informatics and Cognitive Computing; Berkeley, CA, USA; 2018. pp. 254-260.

[21] Mihalcea R, Pulman S. Linguistic ethnography: Identifying dominant word classes in text. In: 10th International Conference on Intelligent Text Processing and Computational Linguistics; Mexico City, Mexico; 2009. pp. 594–602.

[22] Pennebaker JW, Boyd RL, Jordan K, Blackburn K. The development and psychometric properties of LIWC 2015, Austin, TX, USA: University of Texas at Austin, 2015.

[23] Onan A. Sentiment analysis on Twitter based on ensemble of psychological and linguistic feature sets. Balkan Journal of Electrical and Computer Engineering 2018; 6: 69-77.

[24] Onan A. Classifier and feature set ensembles for web page classification. Journal of Information Science 2016; 42: 150-165.

[25] Onan A, Korukoğlu S, Bulut H. Ensemble of keyword extraction methods and classifiers in text classification. Expert Systems with Applications 2016; 57: 232-247.

[26] Kantardzic M. Data Mining: Concepts, Models, Methods, and Algorithms. New York, NY, USA: John Wiley & Sons, 2011.

[27] Cortes C, Vapnik V. Support-vector networks. Machine Learning 1995; 20: 273-297.

[28] Breiman L. Random forests. Machine Learning 2001; 45: 5-32.

[29] Zhou ZH. Ensemble Methods: Found Algorithm. Boca Raton, FL, USA: CRC Press, 2012.

[30] Breiman L. Bagging predictors. Machine Learning 1996; 24: 123-140.

[31] Ho TK. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 1998; 20: 832-844.