

Self-Deprecating Humor Detection: A Machine Learning Approach

Ashraf Kamal¹ and Muhammad Abulaish², *SMIEEE*

¹ Jamia Millia Islamia, (A Central University), New Delhi, India
ashrafkamal.mca@gmail.com

² South Asian University, New Delhi, India
abulaish@sau.ac.in

Abstract. Humor is one of the figurative language categories, and it is mainly used in human communication to express emotions and sentiments. Due to the complex structure of humorous texts, automatic humor detection is a challenging task. The detection becomes more challenging when we consider *self-deprecating humor*, which is a special category of humor in which users generally criticize and put themselves down. Interestingly, in recent years *self-deprecating humor* has been seen as a new business marketing trend, such as brand endorsement, advertisement, and content marketing. In this paper, we propose a novel *self-deprecating humor* detection approach using machine learning technique with an aim to enhance *self-deprecating humor* based marketing strategies. We have identified 16 new features related to three different feature categories – *self-deprecating pattern*, *exaggeration*, and *word-embedding*, and considered 11 humor-centric features from baseline works, and trained random forest classifier for detecting self-deprecating humor in Twitter. The proposed approach is evaluated over Twitter and two baseline datasets, and it performs significantly better in terms of standard information retrieval metrics.

Keywords: Social network analysis, Humor detection, Self-deprecating humor, Machine learning

1 Introduction

Humor is a special category of figurative language to express emotions and sentiments using laughter, jokes, and amusement in human communication. Its computational detection is an interesting, challenging and important task due to its complex characteristics and non-literal expression [1]. It can be useful for many applications, such as opinion mining and sentiment analysis, e-education, trends discovery, and e-advertisement [2]. *Self-deprecating humor* is a special category of humor in which users put themselves down. The online Urban dictionary³ (last accessed on 07-Sep-19) defines *self-deprecating humor* as a “*humor where you put yourself down. Sometimes funny, but sometimes overused fishing for*

³ <https://goo.gl/JoSaak>

compliments or a signal of low self-esteem". For example, consider a humorous text "out of all the things I lost *I miss my mind the most*", in which the phrase *I miss my mind the most* is used for self-deprecating.

Recently, *self-deprecating humor* is seen as a new marketing trend and many brands considered this category of humor for self-promotion through deprecating and disparaging themselves by accepting their own flaws in a humorous fashion without losing the brand value. Self-deprecating style of endorsement creates a sense of community by making humanized a relationship with the customer⁴. It is mainly used for brands⁵, content marketing⁶, and advertisement⁷ (last accessed on 07-Sep-19). Interestingly, it is also seen in celebrities interviews⁸ and politician speech⁹ (last accessed on 07-Sep-19).

In this paper, we propose a novel approach for *self-deprecating humor* detection with an aim to enhance *self-deprecating humor* based marketing strategies. The proposed approach follows a layered design in which first layer uses a semi-automated process to identify candidate *self-around* instances from the dataset, and second layer performs feature extraction and classifier learning for self-deprecating humor detection. The proposed approach identifies 16 new features based on *self-deprecating pattern*, *exaggeration*, and *word-embedding*, and uses 11 humor-centric features from baseline works to train random forest classifier for detecting self-deprecating humor in Twitter. It is evaluated over three datasets and performs significantly better in terms of standard information retrieval metrics.

The remainder of this paper is organized as follows. Section 2 presents a brief review of the state-of-the-art techniques and approaches for computational humor detection. It also highlights how our proposed approach differs from existing state-of-the-art techniques. Section 3 presents the functional details of our proposed approach. Section 4 explains the formulation of various newly identified and existing features. Section 5 presents the statistics of the datasets, including the data crawling and pre-processing processes. Section 6 presents our experimental setting and evaluation results. Finally, section 7 concludes the paper with future research directions.

2 Related Work

Automatic humor recognition is considered as a classification problem and the main task is to judge whether a textual message is humorous or non-humorous [1]. Mihalcea and Strapparava [4] classified humorous or non-humorous text in One-Liners English jokes, news sentences, BNC corpus, and proverb lists datasets. Reyes et al. [5] considered humor generation using supervised machine learning

⁴ <https://goo.gl/dfyXTr>

⁵ <https://goo.gl/2bMYqZ>

⁶ <https://goo.gl/H3W3u7>

⁷ <https://bit.ly/2Woblev>

⁸ <https://goo.gl/uDyvfw>

⁹ <https://goo.gl/KN1jHi>

techniques. Zhang and Liu [2] and Raz [6] applied humor detection in Twitter. Yang et al. [3] considered a random forest classifier to identify humor using 10-fold cross validation technique. They considered features, such as incongruity, ambiguity, interpersonal effect, and phonetic style. They also introduced humor anchors in the form of words or phrases that play a role in recognizing humor.

Zhang et al. [7] introduced features like contextual knowledge, affective polarity, and subjectivity for humor recognition. Liu et al. [1] proposed a sentiment association for humor recognition in discourse relations. Beukel and Aroyo [8] introduced homonym as an indicative feature for humor recognition. Chen and Soo [11] applied a deep learning approach for humor recognition using a convolutional neural network. Ortega-Bueno et al. [9] proposed an attention-based recurrent neural network for humor detection in Spanish language and Ermilov et al. [10] applied the supervised approach in the Russian language for humor recognition. Gultchin et al. [12] applied word embedding using the Euclidean vector for humor recognition.

It can be seen from the above discussion that none of the existing works aims to identify *self-deprecating humor*, in which users generally undervalue, disparage, or deprecate themselves using humorous words or phrases.

3 Proposed Approach

This section presents the functional details of the proposed self-deprecating humor detection technique. As stated earlier, the proposed approach follows a layered design in which first layer aims to identify candidate self-around instances from the dataset, and second layer focuses on feature extraction and classifier learning for self-deprecating humor detection. Further details about both layers are presented in the following sub-sections.

3.1 Layer-1

After an in-depth analysis of the datasets, it is noticed that in many instances (messages) generally users refer themselves. For example, consider the message “I love to stay at home on Sunday.” We consider all such instances as *self-around*. Moreover, many *self-around* instances are found to be *self-deprecating humor*, in which users deprecate, undervalue or disparage themselves using humorous words or phrases.

The first layer applies a semi-automated process to identify all candidate *self-around* instances from the datasets. Using this filtration process, only candidate *self-around* instances are retained for further processing and rest of the instances are discarded to enhance the efficacy of the *self-deprecating humor* detection process by the second layer.

Algorithm 1 presents the pseudo codes of the filtration process mentioned above. Initially, Spacy¹⁰ tagger is applied to generate tokens for each instances

¹⁰ <https://spacy.io/>

of the dataset. In an instance, if any of the token is first person singular (plural) personal pronoun, such as ‘*i*’, ‘*am*’, ‘*me*’, ‘*my*’, ‘*mine*’, ‘*myself*’, ‘*we*’, ‘*are*’, ‘*us*’, or ‘*our*’ then it is added to the *self-around explicit* file, otherwise it is added to the *self-around implicit* file. In algorithm 1, steps 3–9 show *self-around explicit* file creation process, whereas step 10 shows the *self-around implicit* file creation process.

The *self-around implicit* file is analyzed manually to identify candidate self-around instances. After manual analysis, both *self-around explicit* and *self-around implicit* files are merged and considered as the candidate *self-around* dataset, which is passed to the second layer for feature extraction and classification model learning.

Algorithm 1: Self-Around Instance Detection

Input : A file F containing pre-processed instances
Output: Files F_{exp} and F_{imp} containing self-around explicit and implicit instances, respectively

```

1 foreach instance in F do
2    $k \leftarrow \text{spacy}(\text{instance})$ ;           /* tokenization */
3    $n \leftarrow \text{no-of-token}(k)$ 
4   for  $i \leftarrow 1$  to  $n$  do
5     if  $k[i].\text{token} \in \{i, am, me, my, mine, myself, we, are, us, our\}$  then
6        $\text{append}(F_{exp}, k)$ ; /*  $k$  is self-around explicit instance */
7       goto step 1;
8     endif
9   endfor
10   $\text{append}(F_{imp}, k)$ ; /*  $k$  is self-around implicit instance */
11 endforeach

```

3.2 Layer-2

The second layer (i.e., layer-2) considers candidate self-around dataset as an input for feature extraction and classifier learning. A total number of 16 new features from three different feature categories are identified, out of which nine are based on *self-deprecating patterns*, three are *exaggeration*, and four are based on *word embeddings*. In addition, 11 features from one of the baseline works by Yang et al. [3] are used in this study. Further details about these features are presented in the following section. Finally, layer-2 learns one of the popular machine learning techniques, random forest classifier, for detecting *self-deprecating humor*.

4 Feature Extraction

This section presents the details about existing baseline features and newly identified features for detecting *self-deprecating humor*.

4.1 The Features Followed from Baseline

We have considered baseline features from Yang et al. [3]. A total number of 11 semantic structure associated humor-centric features are taken from 3 feature categories, such as *ambiguity*, *interpersonal effect*, and *phonetic style*. These semantic structures reflect important information about an instance to be a humorous.

4.2 Newly Proposed Features

Apart from baseline features, we have identified a total number of 16 features from three feature categories – *self-deprecating pattern*, *exaggeration*, and *word-embedding*. A detailed discussion about these feature categories is given in the following paragraphs.

Self-Deprecating Pattern (SDP):

The *self-deprecating pattern* features is inspired from Abualish and Kamal [14] work in which they proposed *self-deprecating sarcasm* detection task. Since *sarcasm* is considered as the aggressive form of *humor* [15], we have identified a similar but more reliable nine self-deprecating pattern-based features related to humor. These features mainly target self-deprecating patterns in an instance of the dataset. The patterns are either based on the relative order of the Part-of-Speech (POS) tags and tokens or frequency count of the tokens in an instance. All features in this category are binary in nature and they are briefly described in the following paragraphs.

\mathcal{F}_1 (interjection followed by ‘i’ or ‘we’) In an instance, if an interjection (UH-tagged word) is found and an immediate next token is either ‘i’ or ‘we’ (i.e., $UH \rightarrow i(we)$), then the value of this feature is considered as 1, otherwise 0. For example, “*Wow we love going to school after fever!*”, is a self-deprecating humor using this pattern.

\mathcal{F}_2 (common self-deprecating pattern) In an instance, if any of the following common self-deprecating pattern matches ($P1$ to $P5$) is found, then the value of this feature is set to 1, otherwise 0.

- $P1$: If the token ‘it’ is followed by a question word (WRB-tagged word), i.e., $it \rightarrow WRP$.
- $P2$: If a question word (WRP-tagged word) is followed by a personal pronoun (PRP-tagged word), i.e., $WRP \rightarrow PRP$.

- $P3$: If the token ‘*i*’ is followed by the word ‘love’, i.e., $i \rightarrow \text{love}$. During analysis, ‘love’ is the most frequently used word found in self-deprecating instances.
- $P4$: If an adjective (JJ-tagged word) is followed by an adverb (RB-tagged word), i.e., $JJ \rightarrow RB$.
- $P5$: If an adverb (RB-tagged word) is followed by an adjective (JJ-tagged word), i.e., $RB \rightarrow JJ$.

\mathcal{F}_3 (token ‘*i*’ or ‘*we*’ followed by negative modal verb) In an instance, if either a token ‘*i*’ or ‘*we*’ is found and an immediate next tag is a modal verb (MD) and then followed by token ‘*not*’ (i.e., $i(\text{we}) \rightarrow MD \rightarrow \text{not}$), then the value of this feature is set to 1, otherwise 0. For example, “*I can not* delay to go office anymore for more work.”, is a self-deprecating humor using this pattern.

\mathcal{F}_4 (token ‘*i*’ or ‘*we*’ followed by verb) In an instance, if either a token ‘*i*’ or ‘*we*’ is found and an immediate next tag is verb (non-3rd person singular present) (i.e., $i(\text{we}) \rightarrow VBP$), then the value of this feature is considered as 1, otherwise 0. For example, “*We love* deadlines at work, especially on Christmas eve.”, is a self-deprecating humor using this pattern.

\mathcal{F}_5 (token ‘*i*’ or ‘*we*’ followed by past tense verb) In an instance, if either a token ‘*i*’ or ‘*we*’ is found and an immediate next tag is past tense verb (i.e., $i(\text{we}) \rightarrow VBD$), then the value of this feature is considered as 1, otherwise 0. For example, “*I used* to be a doctor but *i* did not have the patients.”, is a self-deprecating humor using this pattern.

\mathcal{F}_6 (conjunction followed by token ‘*i*’ or ‘*we*’) In an instance, if a conjunction tag is found and an immediate next token is either ‘*i*’ or ‘*we*’ (i.e., $CC \rightarrow i(\text{we})$), then the value of this feature is considered as 1, otherwise 0. For example, “*We have* got some powdered water but *we* do not know what to add.”, is a self-deprecating humor using this pattern.

\mathcal{F}_7 (question word followed by token ‘*i*’ or ‘*we*’) In an instance, if a question word tag is found and an immediate next token is either ‘*i*’ or ‘*we*’ (i.e., $WRB \rightarrow i(\text{we})$), then the value of this feature is considered as 1, otherwise 0. For example, “*when i* am not in my right mind my left mind gets pretty crowded.”, is a self-deprecating humor using this pattern.

\mathcal{F}_8 (token ‘*i*’ count) In an instance, if the token ‘*i*’ occurs at least two, then the value of this feature is considered as 1, otherwise 0. For example, “*i* am not cheap but *i* am on special this week.”, is a self-deprecating humor using this pattern.

\mathcal{F}_9 (token ‘my’ count) In an instance, if the token ‘my’ occurs at least two, then the value of this feature is considered as 1, otherwise 0. For example, “my weight is perfect for my height which varies.”, is a self-deprecating humor using this pattern.

Exaggeration (EXA):

Exaggeration plays an important role to create over emphasis, over-statement in a humorous instance. The exaggeration contains an intensifier in the form of adverb, adjective, and interjection in an instance, and its frequency count is considered for self-deprecating humor detection. All features in this category are also *binary* binary and their values are either 0 or 1. A total number of three *exaggeration* features are identified to capture the frequency count of adverbs, adjectives, and interjections.

\mathcal{F}_{10} (interjection count) In an instance, if an interjection tag ‘UH’ occurs at least twice, then the value of this feature is assigned as 1, otherwise 0. For example, “oh wow! I am complete without holidays.”, is a self-deprecating humor using this pattern.

\mathcal{F}_{11} (adjective count) In an instance, if an adjective tag ‘JJ’ occurs at least twice, then the value of this feature is assigned as 1, otherwise 0. For example, “It is so great I am late as always !!1 perfect Monday.”, is a self-deprecating humor using this pattern.

\mathcal{F}_{12} (adverb count) In an instance, if an adverb tag ‘RB’ occurs at least twice, then the value of this feature is assigned as 1, otherwise 0. For example, “I am not a liar he just arranges the truth in his favor.”, is a self-deprecating humor using this pattern.

Word Embedding (WE):

Humor relies on a certain type of opposition or contradiction. These features are taken from Joshi et al. [13] in which authors considered semantic similarity or discordance works as a clue to handle context incongruity in an instance. A total number of four word embedding-based features are considered with an aim to capture those self-deprecating humorous instances in which sentiment-bearing words are absent, but semantic similarity or discordance between words are present. The word2vec¹¹ approach is implemented for the word embedding-based features. We trained the word2vec model over all three datasets and used equation 1 to calculate *Cosine* similarity between all word-pairs (W_m, W_n) in an instance. Finally, dissimilarity values between word-pairs are determined using equation 2.

$$\text{Cosine}(W_m, W_n) = \frac{W_m \cdot W_n}{|W_m| \cdot |W_n|} \quad (1)$$

¹¹ <https://code.google.com/archive/p/word2vec/>

$$Dissimilarity(W_m, W_n) = 1 - Cosine(W_m, W_n) \quad (2)$$

\mathcal{F}_{13} (maximum score of most similar word pair) The value of this feature is considered as the highest score of the most similar word-pair among all possible word-pairs of an instance.

\mathcal{F}_{14} (minimum score of most similar word pair) The value of this feature is considered as the lowest score of the most similar word-pairs among all possible word-pairs of an instance.

\mathcal{F}_{15} (maximum score of most dissimilar word pair) The value of this feature is considered as the highest score of the most dissimilar word-pair among all possible word-pairs of an instance.

\mathcal{F}_{16} (minimum score of most dissimilar word pair) The value of this feature is considered as the lowest score of the most dissimilar word-pair among all possible word-pairs of an instance.

5 Datasets

In this section, we discuss all datasets, including baselines, used for the empirical evaluation of the proposed self-deprecating humor detection technique.

- Baseline datasets: Baseline datasets are collected from two sources. First, *Pun of the Day* dataset from Yang et al. [3]. Pun belongs to a wordplay in which similar sounding words or words with multiple meanings indicate humorous effects. Second, 16000 *One-Liners* dataset is taken from Mihalcea and Strapparava [4].
- Twitter dataset: We have crawled the Twitter dataset using REST API in Python 2.7 during May 2019. We considered a hashtag-based approach to curate the Twitter dataset. Using hashtags, users self-label their Twitter instances (i.e., tweets), such as #humor and #not. Humorous instances are collected using #humor, #fun, and #love hashtags, whereas non-humorous instances are collected using the #not and #hate hashtags.

After crawling the Twitter dataset, we have applied a number of pre-processing tasks to obtain the fine-tuned dataset for the experimental evaluation of the proposed approach. The Twitter-specific pre-processing module consists of various data cleaning steps, such as removal of URLs, @mentions, retweets, and hashtags from the Tweets. In addition, other data cleaning steps, such as removal of numbers, duplicate instances, dots, ampersands, extra white spaces, hexa-characters, quotes, emoticons, and upper-case letters to lower case conversion are applied on both base-line and Twitter datasets.

Table 1 presents the statistics of the dataset. Table 2 presents the statistics of the retained instances after filtration through the semi-automated process of layer-1.

Table 1. Statistics of the datasets

Datasets	#Humor	#Non-humor	Total (#instances)
16000 One-Liners	16000	16002	32002
Pun of the Day	2423	2403	4826
Twitter	10000	10000	20000
Total (#instances)	28423	28405	56828

Table 2. Retained instances after filtration in layer-1

Datasets	#Humor	#Non-humor	Total (#instances)
16000 One-Liners	4520	1414	5934
Pun of the Day	618	728	1346
Twitter	2910	3405	6315
Total (#instances)	8048	5547	13595

6 Experiment Setup and Results

This section presents the experimental setup and evaluation results of our proposed approach. It also presents a comparative analysis of the proposed approach with one of the existing state-of-the-art approaches. All experimental tasks were evaluated on a machine with a configuration of 3.40 GHz Intel-Xeon processor, 16 GB RAM, and Windows 8.1 Pro (64-bit) operating system. Most of the functioning modules are implemented in `Python 2.7`. However, Random Forest (RF) classifier is implemented in `WEKA` tool kit 3.8 and evaluated through 10-fold cross validation using standard information retrieval metrics discussed in the following sub-section.

6.1 Evaluation Metrics

Standard information retrieval metrics such as *precision*, *recall*, and *f-score* are used for the experimental evaluation of the proposed approach. *Precision* measures the correctness, whereas *recall* measures the completeness of a classifier.

F-score is calculated as the harmonic mean of the *precision* and *recall* values. Furthermore, these metrics are calculated using True Positives (*TP*), False Positives (*FP*), and False Negatives (*FN*). *TP* is defined as the number of correctly retrieved *self-deprecating humor* instances, whereas *FP* is defined as the number of *non-self-deprecating humor* instances that are identified as *self-deprecating humor*. Finally, *FN* is defined as the number of *self-deprecating humor* instances that are missed. Formally, *precision*, *recall*, and *f-score* are defined in equations 3, 4, and 5 respectively.

$$\text{Precision } (P) = \frac{TP}{TP + FP} \quad (3)$$

$$\text{Recall } (R) = \frac{TP}{TP + FN} \quad (4)$$

$$\text{F-score } (F) = \frac{2 \times P \times R}{P + R} \quad (5)$$

6.2 Evaluation Results and Comparative Analysis

Table 3. Performance evaluation results using RF classifier with 10-fold cross validation on datasets presented in table 2

Features ↓	Datasets →	16000 One-Liners			Pun of the Day			Twitter		
		P	R	F	P	R	F	P	R	F
Yang et al. [3]	HCF	0.78	0.88	0.83	0.52	0.48	0.50	0.56	0.60	0.56
Newly proposed features	SDP	0.76	0.98	0.85	0.72	0.50	0.60	0.53	0.63	0.58
	EXA	0.76	0.97	0.86	0.65	0.62	0.61	0.52	0.74	0.62
	WE	0.78	0.88	0.83	0.61	0.62	0.62	0.55	0.54	0.55
Yang et al. [3] + Newly proposed features	Hybrid (HCF + SDP + EXA + WE)	0.79	0.95	0.87	0.71	0.72	0.72	0.64	0.58	0.61

In this section, we discuss evaluation results and comparative analysis of our proposed self-deprecating humor detection approach. Table 3 presents the performance evaluation results using RF classifier with 10-fold cross validation on the datasets presented in table 2. Similarly, figures 1, 2, and 3 present the visualization of the comparative analysis of the newly proposed features and baseline features in terms of *precision*, *recall*, and *f-score* on *16000 One-Liners*, *Pun of the Day*, and *Twitter* datasets, respectively.

To the best of our knowledge, *self-deprecating humor* detection is a new problem and no study has been reported yet in the existing literatures. Therefore, comparative evaluation of our proposed approach is not possible. However, Yang et al. [3] have considered humor-centric-features (HCF) for humor

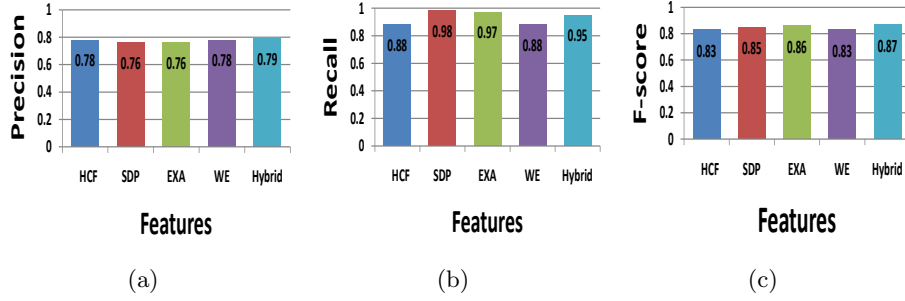


Fig. 1. Comparative analysis of the HCF (Yang et al. [3]) and newly proposed features over 16000 One-Liners dataset (a) Precision (b) Recall (c) F-score.

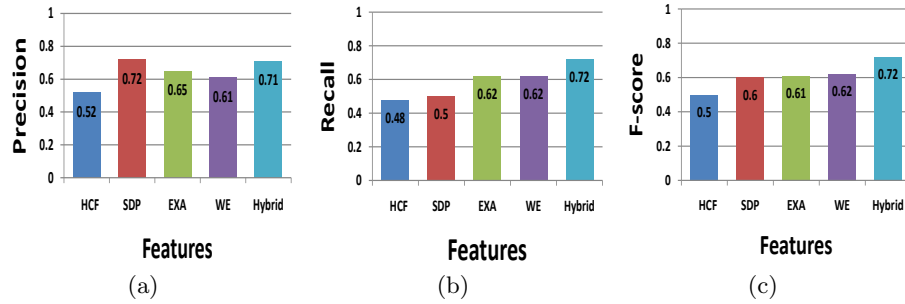


Fig. 2. Comparative analysis of the HCF (Yang et al. [3]) and newly proposed features over Pun of the Day dataset (a) Precision (b) Recall (c) F-score.

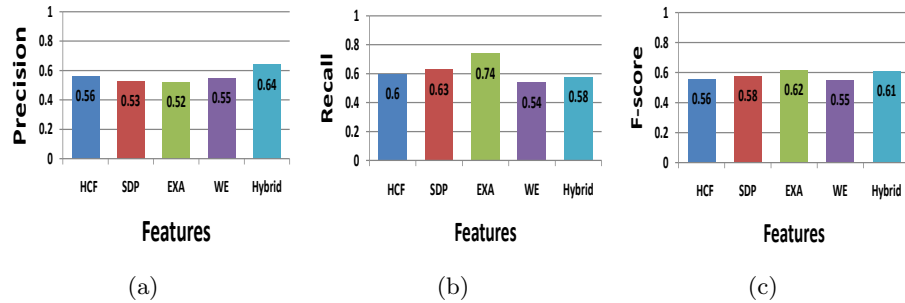


Fig. 3. Comparative analysis of the HCF (Yang et al. [3]) and newly proposed features over Twitter dataset (a) Precision (b) Recall (c) F-score.

detection. Therefore, we have analyzed the effectiveness of their features for self-deprecating humor detection. We have also analyzed the combined effect of Yang et al. features and our newly proposed features towards self-deprecating humor detection.

Table 3 presents evaluation results using the HCF, newly proposed features (i.e., SDP, EXA, and WE), and their combination. It can be observed from this table that our newly proposed features show significantly better classification results. The highest *precision* of 0.78, 0.72, 0.55 are obtained over *One Liner*, *Pun of the Day*, and *Twitter* datasets using newly proposed features. Similarly, it can be observed that combining HCF and newly proposed features resulted in improved f-score over two datasets.

7 Conclusion and Future Work

In this paper, we have considered a new problem of *self-deprecating humor* detection and modeled it as a binary classification task. We have identified a list of 16 new features that are clubbed with existing humor-centric features to learn classification models. We have adopted a layered-design approach in which the first layer filters out all those instances that are not self-around. The main intent behind this work is to enhance those systems that target self-deprecating humor-based marketing strategies. Detecting new *self-deprecating* discriminating features and experimental evaluation of the proposed approach over large datasets seems one of the future directions of research.

Acknowledgment

This publication is an outcome of the R&D work undertaken project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation.

References

1. Liu, L., Zhang, D., Song, W.: Modeling sentiment association in discourse for humor recognition. In: Proceedings of the 56th Association for Computational Linguistics (Short Papers), pp. 586–591. ACL, Melbourne, Australia (2018)
2. Zhang, R., Liu, N.: Recognizing humor on twitter. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 889–898. ACM, Shanghai, China (2014)
3. Yang, D., Lavie, A., Dyer, C., Hovy, E.: Humor recognition and humor anchor extraction. In: Proceedings of the International Conference on Empirical Methods in Natural Language Processing, pp. 2367–2376. ACL, Lisbon, Portugal (2015)
4. Mihalcea, R., Strapparava, C.: Making computers laugh: investigations in automatic humor recognition. In: Proceedings of the International Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 531–538. ACL, Vancouver, Canada (2005)

5. Reyes, A., Rosso, P., Buscaldi, D.: From humor recognition to irony detection: the figurative Language of Social Media. *Data & Knowledge Engineering*. **74**, 1–12 (2012)
6. Yishay, R.: Automatic humor classification on twitter. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Student Research Workshop*, pp. 66–70. ACL, Montréal, Canada (2012)
7. Zhang, D., Song, W., Liu, L., Du, C., Zhao, X.: Investigations in automatic humor recognition. In: *10th International Symposium on Computational Intelligence and Design*, pp. 272–275. IEEE, Hangzhou, China (2017)
8. Van D.B.S., Aroyo, L.: Homonym detection for humor recognition in short text. In: *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 286–291. ACL, Brussels, Belgium (2018)
9. Ortega-Bueno, R., Muniz-Cuza, C.E., Pagola, J.E.M., Rosso, P.: UO UPV: deep linguistic humor detection in spanish social media. In:
10. Ermilov, A., Murashkina, N., Goryacheva, V., Braslavski, P.: Stierlitz Meets SVM: humor detection in russian. In: *Proceedings of the 7th Conference on Artificial Intelligence and Natural Language*, pp. 178–184. Springer, St. Petersburg, Russia (2018)
11. Chen, P.-Y., Soo, V.W.: Humor recognition using deep learning. In: *Proceedings of the 16th Conference on the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 113–117. ACL, New Orleans, Louisiana (2018)
12. Gultchin, L., Patterson, G., Baym, N., Swinger, N., Kalai, A.T.: Humor in word embeddings: cockamamie gobbledegook for nincompoops.
13. Joshi, A., Tripathi, V., Patel, K., Bhattacharyya., Carman, M.: Are word embedding-based features useful for sarcasm detection?. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1006–1011. ACL, Austin, Texas (2016)
14. Abulaish, M., Kamal, A.: Self-deprecating sarcasm detection: an amalgamation of rule-based and machine learning approach. In: *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 574–579. IEEE, Santiago, Chile (2018)
15. Stieger, S., Formann, A.K., Burger, C.: Humor styles and their relationship to explicit and implicit self-esteem. *Personality and Individual Differences*. **50**(5), 747–750 (2011)