SSD : Supervised Sarcasm Detection in Twitter

Swadhin Pradhan, Jian He
CS, The University of Texas at Austin

***Abstract*—Sarcasm is a nuanced form of language in which individuals state the opposite of what is implied. With this intentional ambiguity, sarcasm detection has always been a challenging task, even for humans. Recognition of sarcasm can benefit many sentiment analysis NLP applications, such as review summarization, dialogue systems and review ranking systems [15]. Even recently FBI was also interested to find sarcastic tweets [8] to improve their sentiment mining system. Moreover, this problem of sarcasm detection becomes very important in this age of mining public opinion from social media such as Twitter. Current approaches for automatic sarcasm detection in Twitter rely primarily on lexical cues of an independent tweets. However, the essence of sarcasm depends upon the context and world knowledge. This project aims to address the difficult task of sarcasm detection on Twitter by leveraging social features such as follower list or profile description, contextual feature such as sentiment expressed in the named entities in tweets etc. intrinsic to users expressing sarcasm with lexical and linguistic features. We evaluate our technique, *Supervised Sarcasm Detection* (SSD) on a dataset of 75,253 tweets consisting 38,112 sarcastic tweets having $\#sarcasm$ hashtag. We demonstrate efficiency of *SSD* in identifying sarcastic tweets by gaining F1-score of 97.46 which outperforms different state-of-the-art techniques [9], [19] .**

I. Introduction

Motivation : In recent years, social media sites such as Twitter have gained immense popularity and importance. These sites have evolved into large platforms where users express their ideas and opinions freely. Companies leverage this unique ecosystem to tap into public opinion on their products or services and to provide real-time customer assistance. With the high velocity and volume of social media data, companies rely on tools such as HootSuite[1], to analyze data and to provide customer service. These tools perform tasks such as content management, sentiment analysis, and extraction of relevant messages for the company's customer service representatives to respond to. However, these tools lack the sophistication to decipher more nuanced forms of language such as sarcasm or humor, in which the meaning of a message is not always obvious and explicit.

Our goal in this project is to tackle the difficult problem of sarcasm detection on Twitter. While sarcasm detection is inherently challenging, the style and nature of content on Twitter further complicate the process. Compared to other, more conventional sources such as news articles and novels, Twitter is (1) more informal in nature with an evolving vocabulary of slang words and abbreviations and (2) has a limit of 140characters per tweet which provides fewer word level cues and adds more ambiguity. However, it also different side channels to get some information about the context of a particular tweet like different hashtags, mentions, profile information, general sentiment of the named entities in tweets etc. To put the problem in a perspective, a few examples of sarcastic tweets from our experimental datasets are given below:

1) *Oh wow broken up 4 days and you've moved on already, thanks, don't feel like shit at all. #sarcasm.*
2) *No my roommate play out of tune Zeppelin songs right outside my door isnt annoying. Not at all #sarcasm #sigh.*
3) *Wow! @TWCable_NYC Thanks for the option of high speed internet at $5 a month or 6 months free to save $0.30 depending on the plan. #sarcasm.*
4) *Man, Robinson Cano could be the laziest MVP ever #sarcasm #bigtimesarcasm.*
5) *20 minutes of laundry at 1 am. Awesome #sarcasm.*
6) *Love walking through your cloud of cigarette smoke. Why buy my own pack when I can just inhale yours #sarcasm.*
7) *Is everyone as excited about the #GOP and #Democrat conventions as I am?!? #sarcasm #serioussarcasm #deepyawningmawofsarcaam.*

Our Approach : Current research on sarcasm detection on Twitter [21], [9], [19], [11] has primarily focused on obtaining information from the text of the tweets. These techniques treat sarcasm as a linguistic phenomenon, with limited emphasis on the socio-contextual aspects of sarcasm. However, sarcasm has been extensively studied in psychological and behavioral sciences and theories explaining when, why, and how sarcasm is expressed have been established. All these works point that sarcasm is bound with broader common knowledge (e.g., about news or celebrities), the context known only to the author or author's opinion.

Hence, to follow a systematic approach, we first use and extend different lexical and syntactic features used in [21], [9], [19], [11] to capture the literal form of sarcasm. We explicitly utilize the occurrence of punctuation, stop words, slang, and emoticons which are very closely related to sentiment representation of tweets. Although twitter users

[1]https://hootsuite.com/

think independently when updating their statuses, certain level of similarities in lexical structures are expected. Secondly, compared with normal tweets, more complicated syntactic structures may appear in sarcastic tweets so that implicit meaning of them can be understood by others. Syntactic complexity is an indicator of how easily the information of a tweet can be understood. Conventionally, sarcastic tweets convey messages implicitly, which possibly complicates syntactic structures. Finally and most importantly, we get different socio-contextual information from the information follower, followee, retweets, mentions etc. which represent the social context of the user and the tweet. So, we combine these features to train a supervised learning algorithm to detect sarcasm. We have collected sarcastic tweets with corresponding socio-contextual information using #sarcasm hashtag and also later consolidated the dataset upto 75,253 tweets consisting 38,112 sarcastic tweets by getting tweets from the publicly available tweet ids from the authors of riloff13,davidov10,tomas14. We used SVM and ensemble technique like Logitboost or bagging to get the best F1-score of 97.46 using this feature set in the above dataset.

We make the following contributions in this project:

- We have created a Twitter dataset of reasonable amount of tweets with relevant socio-contextual information.
- To the best of our knowledge, we have employed sophisticated syntactic and social features to capture the context to better detect sarcasm. This approach can help in building language independent sarcasm detector.
- We believe that some classes of sarcastic tweets actually aimed at entities (e.g. may be person or institution or event or show) and so we have extracted the sentiment to get good indication of sarcasm.

In Sec. II, we review related sarcasm detection research. In Sec. III, we formally define sarcasm detection on Twitter. In Sec. V, we discuss dataset collection procedure in detail and then we discuss the nature of different sarcastic tweets. In Sec. VI we discuss different feature set and in Sec. VII we describe the methodology employed in sarcasm detection. Next, in Sec. VIII, we present experimental setup and different evaluation results. Finally, in Sec. IX and Sec. X, we conclude with different possible future directions.

II. Related Work

Sarcasm has been widely studied by psychologists, behavioral scientists and linguists for many years. Theories explaining the cognitive processes behind sarcasm usage such as the echoic reminder theory [14], allusional pretense theory [24], and implicit display theory [26] have been extensively researched. However, automatic detection of sarcasm is a relatively unexplored research topic and a challenging problem [17]. While studies on automatic detection of sarcasm in speech [25] utilizes prosodic, spectral and contextual features, sarcasm detection in text has relied on identifying text patterns [9] and lexical features [13].

Experiments with semi-supervised sarcasm identification on a Twitter dataset were conducted in [9]. They used 5-fold cross validation on their kNN-like classifier using mainly lexical features obtained an F-measure of 0.55 on the Twitter dataset. In [9], authors use a semi-supervised sarcasm identification algorithm on a Twitter dataset and Amazon product reviews. In case of Twitter, authors mainly use 1500 tweets containing *#sarcasm* hashtag and 180 tweets tagged by 15 Amazon Mechanical Turkers [1] as golden test set or initial small labeled training set. The algorithm employs two modules: semi supervised pattern acquisition for identifying sarcastic patterns that serve as features for a classifier, and a classification stage that classifies each sentence to a sarcastic class. Reyes et al. [20] proposed features to capture properties of a figurative language such as ambiguity, polarity, unexpectedness and emotional scenarios. Their corpus consists of five categories (humor, irony, politics, technology and general), each containing 10,000 tweets. The best result in the classification of irony and general tweets was F-measure 0.65. Furthermore, Lukin and Walker [16] explored the potential of a bootstrapping method for sarcasm classification in social dialogue to learn lexical N-gram cues associated with sarcasm (e.g., oh really, I get it, no way, etc.) as well as lexico-syntactic patterns. The work of Riloff et al. [21] identifies one type of sarcasm : contrast between a positive sentiment and negative situation. They used a bootstrapping algorithm to acquire lists of positive sentiment phrases and negative situation phrases from sarcastic tweets. Their evaluation on a human-annotated dataset of 3000 tweets (23% sarcastic) was done using the SVM classifier with uni-grams and bigrams as features, achieving an F-measure of 0.48. [11] introduced a sarcasm detection technique using numerous lexical features (derived from LWIC [18] and Wordnet Affect [27]) and pragmatic features such as emoticons and replies. Tomas et. al. [19] also tried to employ different combinations of machine learning approaches using language independent specific feature set on Czech and English Twitter dataset (780,000 tweets) and achieved F-measure around 0.94.

III. Problem Definition

Sarcasm, while similar to irony, differs in that it is usually viewed as being caustic and derisive. Some researchers even consider it to be aggressive humor and a form of verbal aggression. While researchers in linguistics and psychology debate about what exactly constitutes sarcasm, for the sake of clarity, we use *the activity of saying or writing the opposite of what you mean, or of speaking in a way intended to make someone else feel stupid or show them that you are angry* (Macmillan English Dictionary (2007)). We formally define the sarcasm detection problem on Twitter as follows:

Definition 1. ***Sarcasm Detection on Twitter.*** *Given an unlabeled tweet t from user U along with a set of Us and t's social information S, a solution to sarcasm detection aims to automatically detect if t is sarcastic or not.*

So, our problem is different from past sarcasm detection research which use only text information from t and do not consider the user's and tweet's social information S that are available on Twitter. In SSD, we train a classifier system with a training tweet dataset using different features and test the prediction accuracy in the test dataset.

IV. Dataset Collection and Description

A. Crawler Development

To collect a large-scale dataset from Twitter, we developed a distributed crawling platform to parallelize the crawling process. The open-source tool Tweepy [5] was utilized to use APIs from Twitter application development framework. Twitter has limited each account to send 150 requests within a 15-min time window. To speed up the crawling process, we created 7 twitter accounts to send requests in parallel. Sarcastic tweet dataset consists of the recent $10k$ tweets containing hashtags *#sarcasm*. We collected comprehensible information for each tweet, including the tweeter, tweet text, post time, the count of favorites, the count of retweets, etc. From these tweets, we further collect information for each tweeter. For each Twitter, we collected its follower count, followee count, user ID, status count, the list of followers, the list of followees, etc. Moreover, we have also collected non-sarcastic tweets of around $10k$ from different accounts.

Furthermore, we have contacted different authors [9], [21], [19] for similar works for twitter dataset, but they were unable to provide contents of tweets due to new Twitter terms of services [7]. Instead, authors [9], [21], [19] have provided us twitter ids corresponding with their labeling (These labels are done either by Amazon Turkers or by independent human evaluators). Then, we used our crawlers to collect tweets from those tweet ids with their socio-contextual information (on an average 10% of the tweets were either deleted or inaccessible). Finally, we are able to collect 180 tweets annotated from [9], around 50k tweets from [19] and around 5k tweets from [21]. Thus, we consolidated twitter dataset of size $75,213$ tweets consisting $38,112$ sarcastic tweets.

B. Dataset Description

The whole dataset is divided into three parts: tweets information, user profiles and user social relationships.

Table I shows the format of a tweet information record. Note that, we only list the information we have used in our experiments. A tweet record in our dataset also includes other information, such as user device, geo-location, etc.

A user profile record consists of the user name, followers count, followees count and status count. Table II illustrates the format for a user profile record. Details of the user social relationship dataset can be seen in Table III.

Tweeter	Text	Favoriates Count	Retweets Count

TABLE I
FORMAT FOR TWEET INFORMATION DATASET.

User Name	Followers Count	Followees Count	Status Count

TABLE II
FORMAT FOR USER PROFILE DATASET.

User Name	Follower1, Follower2, $\cdots$	Followee1, Followee2, $\cdots$

TABLE III
FORMAT FOR USER SOCIAL RELATIONSHIP DATASET.

V. Discussion on Sarcastic Tweets

Based on our observations, sarcastic tweets reveal both independent and dependent features. Generally, independent features characterize lexical or syntactic structures of tweets no matter who posted them and what the content or topic of them is. While social network specific characteristics can reveal dependencies among tweets to some extent.

Lexical structures indicate similarities of using words or characters in raw text of tweets. The frequencies of punctuation, stopwords (e.g. and, is) and slang (e.g. booty call represent one aspect of lexical structures. We explicitly utilize the occurrence of emoticons which are very closely related , BFF, BRB) to sentiment representation of tweets. Moreover, hashtags and URLs are special lexicons in tweets, and they are possibly used to clarify the information conveyed in a sarcastic tweet. Some common phrases may be used to reverse the polarity of an utterance. Language models (such as Bigram, Trigram) are efficient ways to find out these phrases. Although twitter users think independently when updating their statues, certain level of similarities in lexical structures are expected to see due to their common intentions to represent sarcasms to their friends.

Compared with normal tweets, more complicated syntactic structures may appear in sarcastic tweets so that implicit meaning of them can be understood by others. The complexity of a syntactic tree can be represented by the number of nodes in the tree, the fraction of words appeared in the syntactic tree and the total words in tweet texts, etc. Syntactic complexity is an indicator of how easily the information of a tweet can be understood. Conventionally, sarcastic tweets convey messages implicitly, which possibly complicates syntactic structures.

Messages in social networks somewhat show some dependencies. On one hand, when people talk about a specific entity (such as a person, a movie, etc.), they tend to describe their opinions in similar ways. And we can always see common sentiments for some entities. For example, when someone wants to comment a movie, posting a sarcastic message can possibly invoke others' interests in following that message.

Named entities represent the target or topic of a tweet. After recognizing named entities appeared in tweets, we are able to classify tweets into different types. For example, tweets about general events (such as weather) and tweets targeting at

specific entities (like movies, actors, etc). Moreover, common sentiments towards specific entities can also possibly improve the accuracy of classification. The sentiment of the whole tweet is considered as the sentiment of named entities in that tweet. The number of named entities in a tweet also represents the intention that a user wants to convey complex messages. On the other hand, the social strength of a tweet can be defined as the number of favorites, retweets, the number of followers and followees the poster has. Generally, social strength can be considered as a factor which measures others' interests in a tweet. Posting sarcastic tweets is an efficient way to attract others' attention. Thus, tweets having higher social strength tend to be more likely sarcastic tweets. Exploiting dependencies can cluster tweets into several types having common properties.

In Table IV, we can see several representative sarcastic tweets which show dependencies we are going to exploit. All these tweets contain a common verb "love" which indicates positive context. Detecting contrary sentiment between the context and the target of the tweet for tweets with simple sentence structure is hard. For example, the target "growing up" is generally a negative thing. Language model is an efficient way to identify this kind of tweets since they only contain very simple phrases. We need more features such as syntactic features to classify general sarcastic tweets having complicated sentence structures. As for sarcastic tweets with named entities, they generally tend to have similar sentiment.

Sarcastic Tweet Types	Example
General Sarcastic Tweets with simple sentence structure	I love growing up #sarcasm
General Sarcastic Tweets with complicated sentence structure	I love when I can't find my resources for my lessons that I thought I had on my desk. #yeg #ualberta #sarcasm
Sarcastic Tweets with Named Entities	I just love Kalpana Bales voice and the way she sings. #sarcastic

TABLE IV
REPRESENTATIVE SARCASTIC TWEETS SHOWING DEPENDENCIES.

VI. FEATURES

We will use three different groups of features to help improve sarcasm detection accuracy, including lexical features, syntactic features, and social features. Table V shows the lexical features we used in our experiments. Lexical features mainly capture lexical patterns of tweets, such as common word usage, phrases, etc. Syntactic features illustrated in Table VI represent syntactic structures of tweets, such as the simple SVO pattern, etc. We exploit social features listed in Table VII to characterize dependencies across tweets. For example, tweets with similar named entities tend to have same sentiments.

VII. METHODOLOGY AND EXPERIMENT SETUP

In this section, we will discuss in detail about umderlying working priniciples of SSD and different evaluation setups.

Punctuation Frequencies (The ratio between the number of punctuation and the total number of tokens)
Stopwords and Slang
Emoticon Frequencies
Capitalization
Number of Hashtags and URLs
Length of Sentence (The number of tokens)
Language Models(Unigram, Bigram and Trigram)

TABLE V
LEXICAL FEATURES

Relative frequencies of different POS tags parsed by the tool TweetNLP[6] (The size of the tagset is 25)
Fraction of words having syntactic function (Syntactic dependency trees are generated by TweeboParser[12],some tokens do not have syntactic function, such as "RT", "@", hashtags, etc.)
Number of inner nodes in dependency trees (Phrases in a tweet can be characterized by subtrees, thus the number of inner nodes is a reliable indicator of the syntactic complexity of a tweet)
Sentiment of tweets (The sentiment of a tweet can be positive, negative or neutral. The online machine learning framework Datumbox[2] is used to analyze sentiment.)

TABLE VI
SYNTACTIC FEATURES.

A. Classifier Description

We have initally attempted to train the system with simple unigram (bag of words), bigram, and tigram language model trained with 70% of the tweets and testing with randomly selected 30% of the tweets. For this binary classification task, we habe used SVM [4] with linear kernel and different ensemble techniques of boosting and bagging. Since we have a wide variety of features, we experimented with various ensemble learning techniques and found that LogitBoost performed best empirically for boosting and bagging with SVM performed best for bagging technique. We used the Weka implementation of LogitBoost [10] and EnsembleSVM for bagging [3] to train a classifier using various combinations of features. We have used Decision Stumps as a base classifier in LogicBoost and ran boosting for 100 iterations. Furthermore, we have used SVM as a base classifier in bagging and ran bagging for 100 iterations. Training time of ensemble techniques were around 20 hours in 8GB quadcore 3.2GHZ Ubuntu machine compared to around 1 hour in SVM.

Named Entities(The number of Named Entities, the length of each named entity, the sentiment of each named entity)
Social strength(The number of favorites, the number of retweets, the follower count of the tweet handler, the followee count of the tweet handler)

TABLE VII
SOCIAL FEATURES

B. Experiment Setup

Fig. 1 shows the infrastructure of the whole system, which is divided into four parts: crawling module, parsing module, feature extracting module and sarcasm classifying module.

In the crawling module, we have developed a distributed crawling platform to collect a large-scale dataset from Twitter. The open-source tool Tweepy[5] was run on multiple machines in parallel, and all data will be centrally managed by the master node.

Raw tweets collected by the crawling module will be fed into the parsing module. The tool ark-twitter-nlp[6] was used to do tokenizing and POS tagging. After obtaining POS tagged tweets, TweeboParser[12] will further run syntactic parsing which will generate syntactic dependency tree for each tweet. Named entity recognition was done by the tool in developed by Ritter et al[22][23]. We applied sentiment analysis by utilizing the online machine learning framework Datumbox[2]. Due to the rate limit of this framework, we also created multiple parallel machine nodes to speed up sentiment analysis.

Parsed tweets are the input of the feature extracting module. We developed three feature extractors to extract corresponding features. After obtaining all tweet features, we are able to run the sarcasm classifying module, in which we used three classifiers, SVM with linear kernel, logitboosting with decision stump and bagging with SVM.

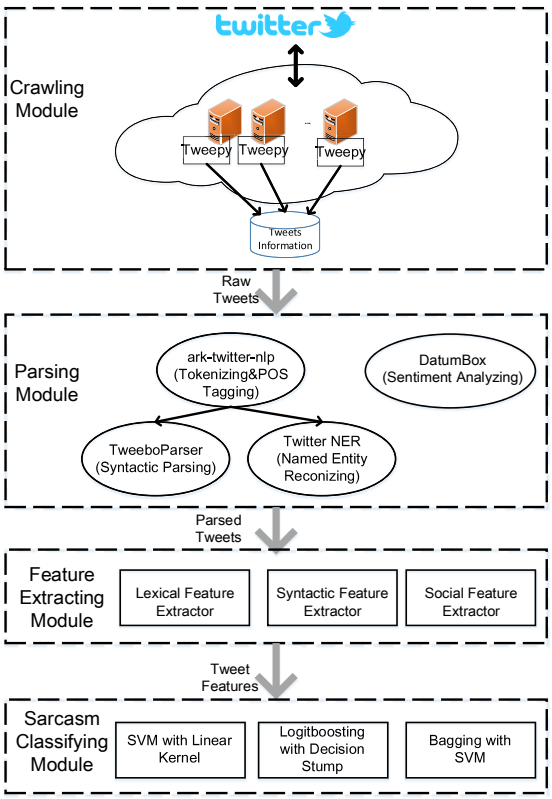

Fig. 1. The infrastructure of system setup.

VIII. Evaluation and Discussion

We conduct a set of experiments to evaluate our approach with different combinations of features and classification techniques. The goodness of classification results is measured by 5 common metrics, which are precision, recall, accuracy, F1-score and Area under Curve (AUC). All detailed results are shown in Table VIII. We have trained classifiers with 70% tweets from both sarcastic and normal datasets. Later, we test the classifier with separated dataset which contained the 30% of the dataset.

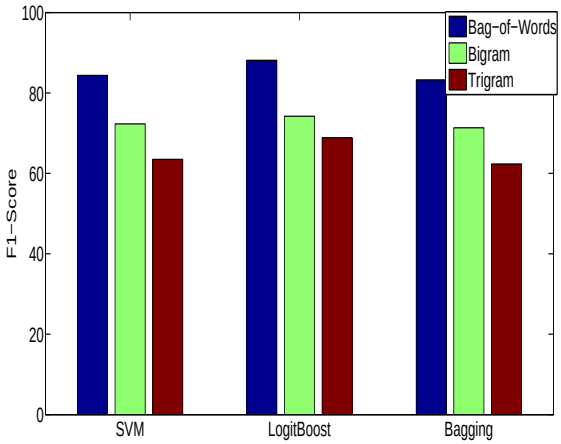

Fig. 2. Effect of different language model specific training in different SSD classifiers

Among various features, language models perform the best, and syntactic features result in serious over-fitting. Fig. 2 shows the results when using different language models. The bag-of-words which corresponds to the unigram language model achieves 10% and 19% higher accuracy compared with bigram and trigram, respectively. Due to highly flexible word usage in tweets, bigrams and trigrams are rarely common. However, we can always see some common bag-of-words since sarcastic tweets contain sentiment related words like “love”, “hate”, “favor”, etc. Frequent occurrence of these words can efficiently distinguish sarcastic tweets from normal tweets.

From the results, we can see that lexical features and syntactic features are easy to make classifiers overfitted. The reduction of accuracy is around 18% and 22% for lexical features and syntactic features compared with the bag-of-words feature. Unlike normal English sentences, tweets generally contain many punctuation, emoticons, etc. The results for lexical features demonstrate that only considering lexical features can not accurately identify sarcastic tweets. That is, lexical features are not in fact specific to sarcastic tweets. Syntactic features are even worse than lexical features. One possible reason is the words which do not have syntactic function will significantly confuse the construction of syntactic dependency trees. However, we have

observed that even though single type of features might fail, but combining them may result in better predictions.

Combining social features with all other possible features we can improve the accuracy to a value over 95%. The significant improvements demonstrate social features are more efficient to characterize a sarcastic tweet. From our dataset, we found that a large portion of sarcastic tweets have named entities embedded. As long as having recognized named entities, common sentiments towards them make sarcastic tweets more classifiable. Moreover, social strength also helps to identify the likelihood that tweets, fitting other sarcastic features, have similar probabilities to be sarcastic tweets if their social strength are at the same level. Therefore, social features which aim to capture dependencies among tweets are better than lexical features and syntactic features to characterize sarcastic tweets.

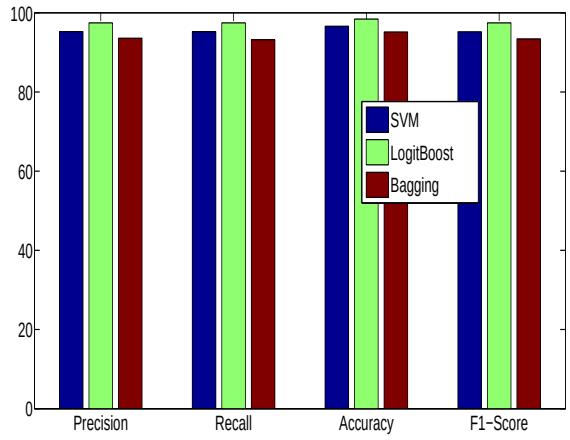

Fig. 3. Comparison of Precision, Recall, F1-Score and Accuracy of different SSD classifiers

Fig. 3 demonstrates the results when applying various classifiers with all features we can use. Initally, we have employed SVM classifier with linear kernel for classifying sarcastic tweets, and got reasonable F1-score with realtively less training time. However, later we have also tried with different ensemble techniques like boosting (using Weka implementation) and bagging (using EnsembleSVM implementation) as it is reported to perform better in literature. Although overall performance of LogitBoost with Decision Stump is the best, the gain is not significant.

We compare our results with other state-of-art approaches in Fig. 4 and Table IX. SASI [9] uses a semi-supervised approach with mainly pattern based lexical feature whereas authors of [19] used varied set of lexical and semantic features. The performance gain is around 34% compared to [9] mainly due to its pattern based approach which suffers from overfitting. Moreover, around 3% accuracy improvement has been achieved by our proposed approach SSD compared to

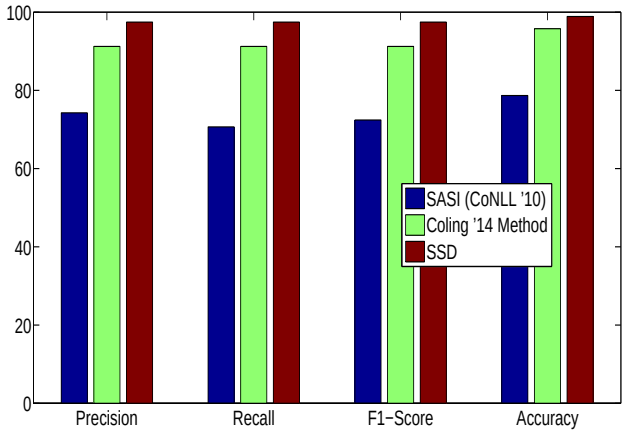

Fig. 4. Comparison of a few competing techniques with SSD

the method proposed in [19]. Since almost all other possible lexical features and syntactic features have been evaluated by our work and other related work, tweet dependencies characterized by social features are more important than those features to classify sarcastic tweets.

Technique	Precision	Recall	F1-Score	Accuracy
SASI [9]	74.27	70.67	72.43	78.68
Coling '14 [19]	91.23	91.23	91.23	95.77
SSD	97.46	97.46	97.46	98.89

TABLE IX
COMPARING DIFFERENT SUPERVISED TECHNIQUES OF SARCASM DETECTION IN TWITTER WITH SSD

IX. FUTURE WORKS

In this project, our goal was to capture the context of tweets in a holistic manner. To achieve this, one immediate future direction might be to get previous tweets of a user to get a better behavioral model which might help in prediction. We can also better model the persona or community of a twitter handle if we also can mine the interest from his follower list and previous retweets/mentions/favorites. Similarly, we can also get previous tweets belonging to particular topic or named entity mentioned in the tweet and get to know the general sentiment. If the general sentiment is negative, then any positive tweet, with high probability, might be sarcastic. Furthermore, in future, we will also attempt to get information about the topics or entities from outside sources such as search APIs. On the other hand, we can develop a hierarchical classifier for different types of sarcastic tweets which will initially predict the class of sarcastic tweet (e.g. general or entity specific) and then employ a class specific model to predict the sarcasm. We believe that this type of topical or socio-contextual model will help in predicting sarcastic tweets in a language independent way and can overcome the low F1-score reported in Czech dataset in [19]. Moreover, as #sarcasm hashtag is a bit noisy, in future, we will try to employ amazon mechanical turkers for more confidence in judgment. We also

Feature set	Method used	Precision	Recall	Accuracy	F1-Score	AUC
Bag-of-Words	SVM with linear kernel	84.0475	84.6621	83.6177	84.3674	86.5413
Word Bigram	SVM with linear kernel	72.0175	72.3411	73.5127	72.3132	77.1137
Word Trigram	SVM with linear kernel	63.8823	63.2914	64.7732	63.4913	67.3412
Only Lexical Feature	SVM with linear kernel	62.0813	62.2723	65.1184	62.3213	68.9023
Only Syntactic Feature	SVM with linear kernel	58.5623	59.4112	61.5234	58.8232	66.6784
All Features with Social Features	SVM with linear kernel	**95.2358**	**95.2218**	**96.6023**	**95.2288**	**97.7812**
Bag-of-Words	Logitboost	88.1239	88.1239	90.0324	88.1239	92.2013
Word Bigram	Logitboost with Decision Stump	73.3487	74.2375	76.5123	74.2234	79.7613
Word Trigram	Logitboost with Decision Stump	68.8812	68.8812	72.1276	68.8812	74.2314
Only Lexical Feature	Logitboost with Decision Stump	64.0823	64.0823	67.4098	64.0823	70.2349
Only Syntactic Feature	Logitboost with Decision Stump	60.6756	60.6756	63.5454	60.6756	64.0873
All Features with Social Features	Logitboost with Decision Stump	**97.4568**	**97.4568**	**98.4176**	**97.4568**	**98.8876**
Bag-of-Words	Bagging with SVM	83.0475	83.5611	84.5177	83.2674	85.3413
Word Bigram	Bagging with SVM	71.2713	71.4531	73.3427	71.3412	75.1745
Word Trigram	Bagging with SVM	62.8845	62.2914	68.7732	62.3457	70.3456
Only Lexical Feature	Bagging with SVM	58.0823	58.0127	61.2384	58.0432	63.4523
Only Syntactic Feature	Bagging with SVM	53.1256	53.1256	56.8674	53.1256	59.0234
All Features with Social Features	Bagging with SVM	**93.5812**	**93.1846**	**95.1456**	**93.3825**	**97.3216**

TABLE VIII
SUMMARY OF RESULTS FOR RUNNING SVM, BOOSTING, AND BAGGING WITH DIFFERENT FEATURE SETS ON DATASETS.

should have investigated the importance of different features in terms of information gain.

X. CONCLUSION

In this project, we have collected and consolidated a reasonable amount of sarcastic tweets for the evaluation purpose. We have also shown that if we include different socio-contextual feature with lexiacal and syntactic feature which basically represents world knowledge, we can achieve better accuracy in sarcasm prediction. There is also inherent variety of sarcastic tweets which needs to be tackled differently for better prediction. Using different supervised machine learning techniques, with these diverse set of features, we have shown to achieve better F1-score compared to different state-of-the-art techniques which mainly rely on syntactic and semantic features.

REFERENCES

[1] Amazon mechanical turk. https://www.mturk.com/mturk/welcome.

[2] Datumbox website. http://www.datumbox.com/.

[3] Ensemblesvm : Bagging with svm. http://homes.esat.kuleuven.be/~claesenm/ensemblesvm/.

[4] Libsvm – a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[5] Tweepy : An easy-to-use python library for accessing the twitter api. http://www.tweepy.org/.

[6] Tweet pos tagger tweetnlp. http://www.ark.cs.cmu.edu/TweetNLP/.

[7] Twitter terms of service 2015. https://twitter.com/tos?lang=en.

[8] Us secret service seeks twitter sarcasm detector. http://www.bbc.com/news/technology-27711109.

[9] D. Davidov, O. Tsur, and A. Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, 2010.

[10] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 1998.

[11] R. González-Ibáñez, S. Muresan, and N. Wacholder. Identifying sarcasm in twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, 2011.

[12] L. Kong, N. Schneider, S. Swayamdipta, A. Bhatia, C. Dyer, and N. A. Smith. A dependency parser for tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, to appear*, 2014.

[13] R. J. Kreuz and G. M. Caucci. Lexical influences on the perception of sarcasm. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, FigLanguages '07, 2007.

[14] S. Kreuz, Roger J.and Glucksberg. How to be sarcastic: The echoic reminder theory of verbal irony. In *Journal of Experimental Psychology: General*, 1989.

[15] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In C. C. Aggarwal and C. Zhai, editors, *Mining Text Data*, pages 415–463. Springer, 2012.

[16] S. Lukin and M. Walker. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue, 2013.

[17] B. Pang and L. Lee. Opinion mining and sentiment analysis. 2008.

[18] J. W. Pennebaker, M. E. Francis, and R. J. Booth. *Linguistic Inquiry and Word Count*. Lawerence Erlbaum Associates, Mahwah, NJ, 2001.

[19] T. Ptáček, I. Habernal, and J. Hong. Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014.

[20] A. Reyes, P. Rosso, and D. Buscaldi. From humor recognition to irony detection: The figurative language of social media. *Data Knowl. Eng.*, 2012.

[21] E. Riloff, A. Qadir, P. Surve, L. D. Silva, N. Gilbert, and R. Huang. Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*, pages 704–714. ACL, 2013.

[22] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, 2011.

[23] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *KDD*, 2012.

[24] S. G. S. Kumon-Nakamura and M. Brown. How about another piece of pie: The allusional pretense theory of discourse irony. In *Journal of Experimental Psychology: General*, 1995.

[25] J. Tepperman, D. R. Traum, and S. Narayanan. "yeah right": sarcasm recognition for spoken dialogue systems. In *INTERSPEECH*. ISCA, 2006.

[26] A. Utsumi. Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. In *Journal of Pragmatics*, 17771806.

[27] R. Valitutti. Wordnet-affect: an affective extension of wordnet. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004.