IDENTIFYING UNDESIRABLE BEHAVIOR IN SOCIAL MEDIA: TOWARDS AUTOMATED FACT-CHECKING AND YOUTUBE META-DATA SPAM DETECTION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Ayush Patwari

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2017

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Saurabh Bagchi, Chair

School of Electrical and Computer Engineering

Dr. Dan Goldwasser

Department of Computer Science

Dr. He Wang

Department of Computer Science

Dr. Bruno Ribeiro

Department of Computer Science

Approved by:

Dr. Voicu Popescu by Dr. William Gorman

Head of the School Graduate Program

This is dedicated to the most important people in my life, Khushboo Ladia, Saroj Patwari and Ashok Patwari.

ACKNOWLEDGMENTS

Firstly, I would like to express my gratitude to Professor Saurabh Bagchi, my advisor, for giving me opportunity to explore research problems on machine learning applied to identifying undesirable behavior in social media. His belief, constant support and motivation helped me get through difficult the phases – which are part and parcel of any academic endeavor – and eventually meet my deadlines. I am also indebted to my co-adviser Professor Dan Goldwasser who agreed to guide me through an interesting yet challenging problem in computational journalism. I have been very fortunate to have been interacting with them in the past two years which has helped me immensely in developing the right attitude and zeal towards research. I would also like to thank Prof. Pawan Goyal, IIT Kharagpur, our collaborator on the project on detecing spam in YouTube, who supported us consistently throughout the project. I also thank Prof. Alex Quinn, ECE, Purdue University, who initially mentored me for the project on fake Amazon reviews. Through discussions with him, I was able to gauge how crowd-powered systems work and its applicability in HCI.

I would like to thank Sudhanshu Bahety, UCSD who was instrumental to our collaboration with IIT KGP and my colleague, Bilal Siddiqui, Purdue Univerisity for his invaluable contribution in carrying Sudhanshu's work forward and help in paper writing. I am also very grateful to two undergraduate students at Purdue University, Naman Patwari and Kush Rustagi, for their continued support in making the fact-checking project happen, including building the visualization tool, exploring crawling techniques to collect data, collecting annotations and at times volunteering to do annotations themselves.

I am also obliged to my Dependable Computing Systems Laboratory (DCSL) colleagues for sitting through my presentations and provide interesting comments and suggestions.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

ABSTRACT

Ayush Patwari Master of Science, Purdue University, August 2017. Identifying Undesirable Behavior in Social Media: Towards Automated Fact-Checking and YouTube Meta-Data Spam Detection. Major Professor: Saurabh Bagchi.

Automated scrutiny and filtering of undesirable data is of paramount importance in the modern digital world driven by plentiful yet unrefined data. We study two related problems in the social media domain: automated fact-checking for computational journalism and fine-grained meta-data spam detection for YouTube, the world's largest video sharing platform.

Fact-checking political discussions has become an essential clog in computational journalism. This task encompasses an important sub-task – identifying the set of statements with 'check-worthy' claims. Previous work has treated this as a simple text classification problem discounting the nuances involved in determining what makes statements check-worthy. We introduce a dataset of political debates from the 2016 US Presidential election campaign annotated using *all* major fact-checking media outlets. We study the characteristics of check-worthy statements and show that there is a need to model conversation context, debate dynamics and implicit world knowledge. We design a multi-classifier system TATHYA, that models latent groupings in data and improves state-of-the-art systems by 19.5% in F1-score on a held-out test set, gaining primarily in recall.

YouTube is plagued with spam campaigns that include spreading malicious links through video description or comments, disseminating adult or illegal content and generating artificial traffic through click baits. We tackle the problem of detecting misleading videos – those having description and title unrelated to the posted content. We show several characteristics of misleading (spam) behavior modeled through tex-

tual and temporal analysis of comments and the uploader. We develop NIRMALYA – a supervised learning framework to detect spam videos that can help prune search recommendations to contain only the legitimate videos. We evaluate our system on a novel manually annotated data set curated from a large corpus of 500K videos. It achieves mean F1-score of 0.82 in detecting spam videos with a recall of 0.83.

1 Tathya

1.1 Introduction

Social media has become a crucial communication medium for politicians. They use it to promote their message and often, bias public opinion in their favor on important issues; sometimes leading to *fake news*. This was evident in the last US presidential election and resulted in an industry wide effort. Claims made during presidential debates were actively scrutinized by multiple fact-checking organizations in real-time. Within the technology industry there has been a thrust towards fact-checking with several big companies introducing preliminary efforts to flag fake news, prominent examples being Google and Facebook.

Table 1.1.
An excerpt from our dataset showing nuances in fact-checking. Statements fact-checked (2,3,8) are *italicized*. Implied information is shown in blue. Pronouns that need to be resolved are marked red and corresponding resolution entities are marked green.

Excerpt: Carly Fiorina, 5th Republican Primary
1. Let me tell you a story.
2. *Soon after 9/11, I got a phone call from the NSA.*
3. *I stopped a truckload of equipment.(for NSA)*
4. It was escorted by the NSA into headquarters.
5. We need the private sector's help, because government is not innovating.
6. Technology is running ahead by leaps and bound.
7. The private sector will help, just as I helped after 9/11.
8. *But they must be engaged (with NSA), and they must be asked.*

Current fact-checking efforts are manual and hence lack in extensive coverage due to information overload. Automation of fact-checking is a difficult problem. Prior work [1] has suggested a natural, structured pipeline of sub-tasks: (1) Extract check-worthy statements (2) Construct structured queries (3) Obtain answers from knowledge bases, e.g., Freebase, Wikipedia, etc. (4) Synthesize the evidence from these multiple sources and pronounce a verdict about the original statement. Each of these have their own footing in the NLP domain and associated challenges. Past research has addressed the problem of detecting check-worthiness [2], perturbing structured queries of check-worthy claims [3] and verification of simple numerical claims [4, 5]. In all such work there is an inherent assumption that properties of the statement itself is sufficient for performing those tasks. In this paper we care about studying check-worthiness of statements which affects the performance of the entire pipeline and is defined below:

Check-worthy statement: *A statement which has a checkable claim, has topical relevance in the current political scenario and is not an opinion on a future event. Such statements if checked, will increase the quality of fact-checking and increase political awareness.*

To better understand the concept of check-worthiness and complexities of this task we discuss a short excerpt by Carly Fiorina in Table 1.1. The statements that were checked (by *nytimes.com*) are italicized and we consider these as 'check-worthy'. At a first glance, we can see that almost all statements have an associated claim that is checkable (i.e. has an associated checkable fact) e.g., in statement (5) the claim could be *the U.S. government is not innovating*, whereas in (8) it could be derived from the implication *The private sector is currently not being engaged with the NSA, therefore it must be engaged* . Only statements (2), (3) and (8) were fact-checked suggesting that 'check-worthy' is a subset of 'checkable'; detecting what is check-worthy becomes even harder and is not the same as 'Separating fact from opinions' [6]. What makes humans able to tell if something is check-worthy is beyond just the ability to read. Along with natural language understanding they have the debate context and world

knowledge, which enables them for example, to connect the implications (show in blue) or link entities being discussed e.g., *they* from (8) to *the private sector* in (7). Then, they can reason if a statement discussing *the private sector* and *NSA* is worth checking. Studies in political journalism [7] have suggested that the rise in fact-checking is motivated more by "professional motives within journalism", such as prestige for the publishing house, than audience satisfaction. Thus, the human checker carries not only her own bias but commercial and economic imperatives of it's organization as well (such as, the first to market with the entire debate fact checked). An automatic detector that merely looks at the current statement loses out on the ability to tap these intuitions.

With these challenges in mind, we address the problem of detecting check-worthy statements in a political discussion as an objective task. After manual inspection we found that there is not a consistent granularity used by the organization for fact-checking. There are instances of both paragraph level and sentence level checks. For our study we a sentence as the unit of fact-checking. This allows use to study check-worthiness at a finer granularity and prevent training a coarse model which would not be able to tackle finer instances. We gather data from the online transcripts of 20 presidential debates and one political speech. We mark a statement as checked if it has been fact-checked by any one of the 9 reputed US fact-checking organizations (Table 1.2). We create the dataset using labels from fact-checking organizations as opposed to in-house annotators to reduce opinion bias and train our model to imitate actual fact-checker decisions. Instead of focusing only on a bag-of-words model, we adopt a more principled approach in language understanding and model factors that affect fact-checking – debate context, important topics of discussion and the nature of claims. These are mapped to features in the NLP domain. We normalize the text in a statement to incorporate entity information from previous statements and then use the normalized text for the experiments. We evaluate our system on two different tasks: 1. classifying sentences are check-worthy or not 2. ranking sentences in order of their check-worthiness.

TATHYA outperforms the current state-of-the-art ClaimBuster [1] [2, 8] by 16.8% on a held-out-set of 4 presidential debates in the classification task. While ranking sentences as check-worthy, our system has a 37% better *ndcg* (normalized discounted cumulative gain) score for the 100 top ranked check-worthy statements. To aid further efforts in automation of fact-checking, we perform a rich error analysis to document the cases in which our model fails and explore directions which could help improve our system.

1.2 Related Work

Computational journalism [9, 10] is one of the main challenges emerging from the explosion in the number of available media and news outlets. Meeting this challenge requires algorithms that can access multiple information sources and evaluate the novelty and validity of content generated by these sources.

In recent years, several works addressed related issues, such as automating fact checking [1, 2], which is so far limited to very specific domains that can leverage existing knowledge bases and numerical statements [4, 5, 11], or existing knowledge by the user [12].

In this work we focus on one aspect of this challenge, identifying what type of content should be checked. This task was previously addressed by [2], and treated as a classification problem. In this paper we argue that the scope of the problem goes beyond a straight-forward classification problem and analyze the difficulties in computationally formulating this task.

In addition to the inherent bias in deciding what should be checked, there are substantial linguistic challenges in analyzing such statements successfully. Some of these challenges bear resemblance to existing work. For example, identifying the arguments and how they relate to one another [13, 14], the discussion strategies used by the speakers [15].

[1] http://idir-server2.uta.edu/claimbuster/

Table 1.2.
Data. R and D denote statements by Republicans and Democrats respectively.

	R	D	Total
Primary Debates			
All	8781	6454	15235
Check-worthy	290	318	608
Check-worthy: Organization Wise			
Washington Post	67	152	219
factcheck.org	63	113	176
Politifact	72	37	109
PBS	35	47	82
CNN	29	33	62
NY Times	29	19	48
Fox News	13	16	29
USA Today	14	9	23
Presidential Debates [2]			
All	2956	2270	5226
Check-worthy (NPR)	300	164	464

Identifying check-worthy claims could also be considered as distantly related to the deception detection task [16,17], however current work on deception detection builds on general representations of deception and bias, expressed through word choice and syntactic patterns [18, 19], and do not address the challenges of fact checking, such as pragmatic inferences and world knowledge representation.

1.3 Dataset

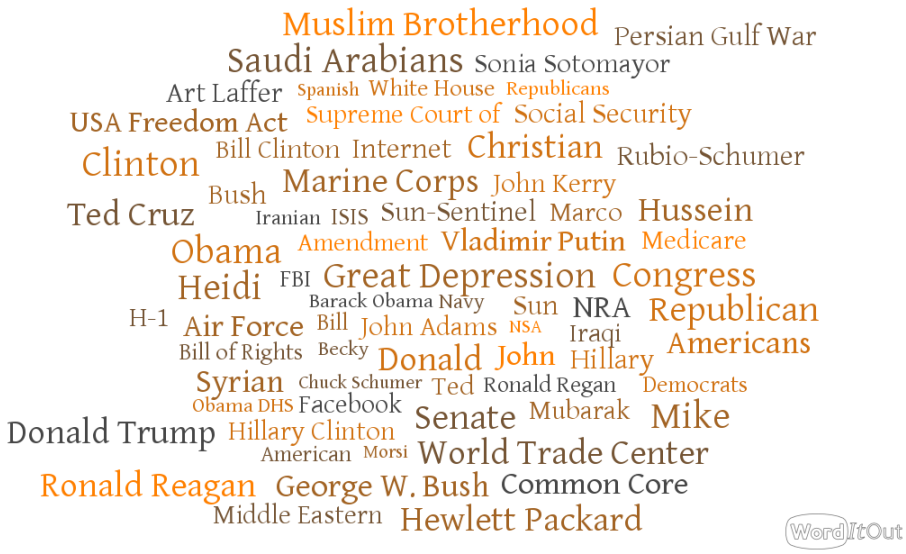
[2]Statements from moderators are also included. 13 statements out of 1239 were fact-checked.

We create our dataset by gathering transcripts from primary debates (7 Republican and 8 Democratic) and presidential debates (3 presidential and one vice-presidential) which form our train and held-out test set respectively. We also include Donald Trump's Presidential Announcement Speech to our development set to analyze a discourse by only one speaker. For each of these transcripts, we split at granularity of a sentence, which forms the unit of checking similar to previous work [2,5]. A statement is labeled as check-worthy if any of the fact-checking organizations listed in Table 1.2 checked it [3]. A total of 20,461 statements were collected with 1,085 of them marked check-worthy. Since some statements were very short, we removed those with less than 2 tokens (tokens are extracted after removing frequently occurring words and stop-words) from our dataset. After this, the corpus had 15,735 statements, out of which 967 are marked check-worthy (6.1% of the corpus – 550 in train set and 417 in test set). All our analysis is based only on the development set and we use the test set only for final evaluation. We avoid using in-house annotators because first, it would require careful consideration of political affiliation of the annotators to reduce bias. Second, this would not give us an insight into how professional fact-checkers perform this task.

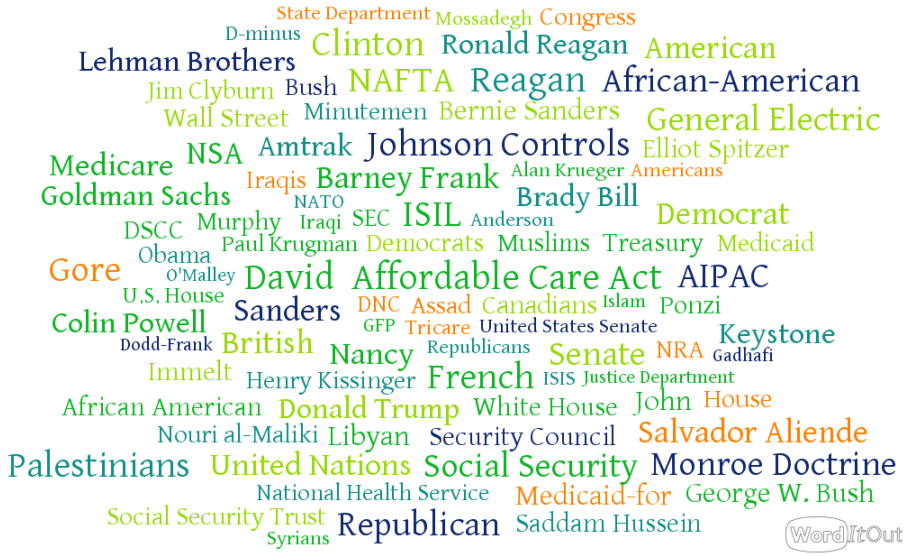
1.4 Challenges

Fact-Checking Subjectivity: The practice of fact-checking is guided by several subjective factors including journalistic reputation, perceived urgency of putting out the results of the fact checking process, likely political bias, and economic constraints of the organization. All of these factors lead to inconsistencies amongst different organizations, e.g., *Washington Post* and *NY Times* checked 16 and 29 statements in the 5th Republican Primary Debate respectively, but their overlap is only 6 sentences. The dataset distribution is shown in Table 1.2. Within the purview of our dataset, it is clear that *Washington Post* and *factcheck.org* have checked more statements by

[3]For presidential debates we collected labels from only NPR to ensure no overlap of organizations between development and test set



(a) Republican



(b) Democrat

Figure 1.1. Named Entities in check-worthy statements. There are a few common entities across the two parties: *Americans*, *ISIS*, *Bush*, *Clinton*, *Donald Trump*, *Obama*, *White House*, *Social Security*, *NSA*, but the majority of the statements are specific to the party (70.4% for Republicans and 77.7% for Democrats).

Table 1.3.
Results from two annotation rounds for 1st presidential debate

	Check-Worthy	Not Check-worthy
Gold	112	896
Annotation1	103	905
Annotation2	64	944
Intersection	31	977
Union	136	872

Democrats, whereas *Politifact* and *NPR* seem to have focused more on Republicans. However, without an exhaustive profile on fact-checking done by each organization, it is hard to quantify or validate this bias. By taking a union of the statements fact checked by any of the organizations, we attempt to mitigate these inconsistencies and avoid training a model with a clear bias for one party.

Party-Wise Differences: In Figure 1.1 we show a word cloud for the frequent Named Entities in check-worthy statements for each party. Quite naturally, candidates very often refer to their opponents and past Presidents, from either party. It is interesting to observe the topic differences between the two parties. Out of the 71 and 94 named entities mentioned by Republicans and Democrats respectively, there are only 21 in common. This gives an impression that a global entity-centric model may not be very useful, rather some context of the previous statements will need to be factored in. Through a study of our dataset, we see that there are other, more involved considerations that are also indicative of whether a statement will be fact checked. For example, a statement on gun rights by Bernie Sanders is considered check-worthy while one by Ted Cruz is not.

Human Evaluation: Our dataset is highly unbalanced with 6.1% statements marked check-worthy according to fact-checking organizations. To understand how non-expert humans would perform in identifying check-worthy statements, we give a task

to two annotators – a graduate and an under-graduate student – who are well versed in the U.S. political scenario. We ask them to read the transcript of the 1st Presidential debate and mark statements which they deem check-worthy. We give them the same definition of check-worthiness as defined in the previous section and no other information regarding the gold labels. The results are shown in Table 1.3. The *gold* labels are taken using fact-checks by NPR. We find that the annotators have low agreement, which has been shown to be case for similar subjective tasks [17]. The agreement between human annotators and professional fact-checkers is presented in a later experiment (Table 1.12). We find that non-expert humans have low agreement with fact-checking organizations. This shows that a dataset annotated by non-expert humans would not truly reflect the intuitions used in professional fact-checking and since there is wide variance between the opinion of human annotators, it would make the dataset inconsistent.

1.5 Detecting Check-worthy Statements

The approach that we adopt to detect check-worthy statements arises from the following observations that we make about the dynamics of the political debates. First, the debates progress in an undulating manner, where each wave corresponds to a certain topic under discussion by multiple candidates. For example, in Table 1.1, Carly Fiorina was discussing the importance of private sector in aiding the government, promptly after which the moderator changes topic to terrorism (marking the end of the previous wave) and Chris Christie responds. Second, within a continuous set of statements made by a speaker, they are likely to introduce entities (antecedents), and then keep referring to them using pronouns (anaphora). Third, the candidates participate with the intention to present themselves in better light or to point holes in past actions and future policies of their fellow candidates. These claims have a dependency structure of the form $(subj, verb, obj)$ where $subj$ may be self or opponent reference. In our human annotator process discussed in Sec 1.3, out of 206 check-

worthy statements, 75 statements were found to be referring to self (or own party) and 33 statements as referring to opponent (or opponent party).

We follow these observations and describe the different feature classes that we use in our model.

Topics of Discussion: Claims in certain topics, like foreign policy, health-care, gun control etc. are more likely to be checked by fact-checkers as compared to those on personal life. Capturing the changes in topic under discussion along a debate is important, for which, we follow a two-pronged approach. First, we train an LDA topic model [20] on transcripts from all previous presidential debates[4] (from 1976 to 2012), all statements made by Hillary Clinton and Donald Trump on public forums before elections and our current dataset. We tune the number of topics to 20, beyond which the top representative words overlap significantly. Using older debates we hope to have a larger corpus and also catch persistent topics which might not have been seen yet in the current cycle of presidential debates. We split a debate into chunks – all statements in the same paragraph – and predict a topic distribution using the trained model for each chunk. The same distribution is assumed for all statements within a chunk. This is to use the surrounding context in determining the topic of current statement. Then we define a context size (say c_1), and for $c\_1/2$ previous chunks and $c\_1/2$ following chunks we compute the cosine similarity with the topic distribution of the current statement. The topic distribution and the cosine similarities form the set of features abbreviated as t_1.

Second, we keep a set of tuples: $(named\_entity, entity\_tag, speaker)$ for named entities that are encountered in previous c_2 statements (forming entity history). For each *named_entity* seen in the current statement, if it has already been seen before there are two cases; it was (a) spoken by the same speaker (b) spoken by a counterpart. We define a binary feature for each repeating entity: $(entity\_tag, repeat\_type)$, where *repeat_type* equals 'support' in case (a) and 'response' in case (b). We only use entities of type *person*, *org* and *misc* (e.g., Republicans, Muslims etc.) in the entity

[4]http://www.presidency.ucsb.edu/debates.php

Table 1.4.
Usefulness of normalization. o_1 and o_2 are classifier outputs using ClaimBuster and TATHYA

	Orignal Text	o_1	Normalized Text	o_2
Clinton	*Yes look, I have made it clear based on Senator Sanders' own record that he has voted with the NRA, with the gun lobby numerous times.*	✓	...	✓
Clinton	*He voted for what we call, the Charleston Loophole.*	✗	Sanders voted ...	✓
Clinton	*He voted to let guns go onto the Amtrak, guns go into National Parks.*	✗	Sanders voted ...	✓

history. These set of features are abbreviated as t_2. Note that, c_2 could be larger than c_1 to capture the important discussion points for a longer history since speakers may rebut a previous point much later in the debate. We also keep the count of each *entity_tag* type for each statement as features.

Text Normalization: It is common to refer to entities using second and third person pronouns in a discussion; this information is lost when analyzing a statement 'out-of-context'. We perform text normalization by propagating chained named-entities along a discussion e.g., in Table 1.4 *Sanders* is a named-entity that is propagated to the subsequent statements. We again restrict propagation to entities of types *person*, *org* and *misc*. We also impose a rule to restrict entities of type *person* being mapped to *it*. For normalization, we also include the speaker information in text, so that references to *you, your* in subsequent turns can be resolved to the previous speaker as discussed in [21]. We exclude resolution of *we*, since it is particularly confusing for an automated system and increases error in normalization. A sample of normalization process is shown in Table 1.4

Part-of-Speech tuples: Claims often have a dependency structure (*subj*,*verb*,*obj*). We want to target *subj* and the *verb* and capture the sense (+ve or −ve) of self and opponent references. To achieve this we define POS target tuples:

- *(noun_tag, verb_tag)* e.g., 'Sanders has'
- *(noun_tag, verb_tag, neg)* e.g., 'She did not'
- *(noun_tag, neg, verb_tag)* e.g., 'I never told'

where $noun\_tag \in \{$*'NN', 'NNS', 'NNP', 'NNPS', 'PRP', 'PRP', 'WP', 'WP'*$\}$, $verb\_tag \in \{$*'VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ'*$\}$ and $neg \in \{$*'neither', 'never', 'no', 'not', 'none'*$\}$. These features are abbreviated as *pos-T*. For each statement the count of each POS-tag is also used as a feature.

Bag-Of-Words: We use bag-of-unigrams using tf-idf weighting as a baseline model. Very frequently occurring n-grams (phrases) are also used, e.g., 'Affordable Care Act'. Stop words are removed and tokens appearing in more than 20% of the sentences are removed. We also include sentiment class ($+ve$ or $-ve$) and number of tokens for each statement as features.

Experimental Setup: We aggregate the above features to build a check-worthiness detector. We also attempted to use stylometric features using constituency parse trees which was shown to be useful in deception detection [19]. However, we found that this was not useful in our task, possibly because politicians are likely to make claims using both simple and complex sentences i.e. they are naturally deceptive, as opposed to fake reviewers on product sites.

In our experiments, we use support vector machines (SVM) for classification and ranking models. We also tried ensemble classifiers Random Forest and ADA-Boost for evaluation, however, SVM gave us the best performance. We first train a check-worthy classifier on a training set of primary debates using cross-validation and perform an ablation study. The held-out test set of the presidential debates is used to compare our model Tathya, with the current state-of-the-art ClaimBuster [2]. In all our evaluation we use $Precision(P)$, $Recall(R)$ and $F1score(F)$ for the check-worthy

class as our metrics of comparison. where $P = \frac{\#correct}{\#predicted}$, $P = \frac{\#correct}{\#gold}$ and $F = 2\frac{P \times R}{P+R}$. The probability scores from the SVM model using Platt's scaling technique [22] are used to rank statements as check-worthy in the ranking model. We use the standard information retrieval metrics – precision@k and ndcg@k – to compare the relevance scores for the two systems.

Finally, we perform a manual error analysis to categorize potential sources of error that effect the false-negative and false-positive rate of our model. We used Stanford CoreNLP[5] and NLTK[6] for tokenization, POS-tagging, NER-tagging and Coreference-Resolution. We train LDA topic model using Gibbs sampling[7]. We use scikit-learn [8] for training classifier and ranking SVM models.

1.6 Multi-Classifier System

Multi-classifier systems have been shown to improve performance in cases where a single classifier system lacks expressiveness for the task at hand [23]. We essentially want to learn a latent grouping of our dataset that best describes the target output function, in our case, given a statement, whether it is check-worthy or not. Such latent representations have been shown to improve performance in the past [24]. To achieve this we design a classifier system as shown in algorithms 1 and 2. In the training step 1, we first cluster the training data into k groups which we use as initialization seeds for the algorithm. In steps 2-7 we learn the best groupings of our data $g_1, ...g_k$ which allows us decrease ambiguity of classification and improve training performance by learning separate classifiers $C_1, ...C_k$. Prediction is done simply using the most-confident classifier for each sample.

Algorithm 1: Multi-classifier System Prediction

Input : input samples X, classifiers $C_1, ...C_k$

Output: output labels Y

for *each* $x \in X$ **do**

  Predict y_i using C_i for $i \in [1, k]$

  Using C^* having highest confidence, predict label y^*

end

Table 1.5.
Accuracy of detecting check-worthy claims for cross-validation. *bow* is bag-of-words, *pos* is POS-tag counts, *ent* is NER-tag counts, *pos-T* is POS-tuples, t_1 is topic agreement, t_2 is entity history

	Without Normalized Text			Using Normalized Text		
	P	R	F	P	R	F
Baseline	0.048	1.000	0.091	–	–	–
bow	**0.191**	0.313	0.231	**0.194**	0.337	0.241
bow, *pos*	0.176	0.381	0.237	0.181	0.399	0.245
bow, *pos*, *ent*	0.171	0.405	0.237	0.185	0.411	0.251
bow, *pos*, *ent*, *pos-T*, t_1, t_2	0.186	**0.414**	**0.252**	0.193	**0.435**	**0.263**

Table 1.6.
Comparison of TATHYA and ClaimBuster on primary and presidential debates.

	Primary Debates			Presidential Debates		
	P	R	F	P	R	F
ClaimBuster	**0.194**	0.32	0.241	0.226	0.148	0.179
TATHYA	0.193	**0.435**	**0.263**	**0.227**	**0.194**	**0.209**
% improvement	-0.500	35.9	9.1	0.4	31.1	16.8

Table 1.7.
Performance of TATHYA for party-wise tests. All values are F1-score for the respective test.

Train / Test	Republican	Democrat
Republican	0.192	0.184
Democrat	0.196	0.236

1.7 Evaluation

1.7.1 Ablation Study

We describe here the model performance for the combination of different feature sets described in the last section. We divide our training set into 16 folds – each fold

[5]https://stanfordnlp.github.io/CoreNLP/
[6]http://www.nltk.org/
[7]https://pypi.python.org/pypi/lda
[8]http://scikit-learn.org/stable/modules/svm.html

Table 1.8.
Most Important Features for TATHYA in different feature classes described in Section 1.5. Named Entity features are prefixed with ENT.

unigrams	topics of discussion	part-of-speech
percent	ENT-PERCENT	PRP-never-VBD
$	ENT-MONEY	NNS-VBP-not
voted	ENT-ORDINAL	NNP-not-VB
lost	ENT-TIME	WP-VBZ-never
supported	topic29 (insurance)	NN-VBN
year	topic14 (budget_spending)	NN-VBP
million	topic16 (labor)	NNS-VBD-not
size	topic15 (terrorism)	NNP-VBD-not
nra	topic17 (nuclear)	WP$

contains statements from one debate/speech – and perform 16-fold cross-validation by training on all-but-one and testing on the remaining debate. We train a linear SVM classifier and select *top-k* features using ANOVA score. We tune hyper-parameters of the system using grid-search on cross-validation. For SVM we keep penalty parameter $C = 0.1$ and $class\_weight$ proportional to half of ratio of non-check-worthy and check-worthy statements. We select top k=850 features, with c_1=6 and c_2=100 for best performance.

We present the results of the study in Table 1.5. The baseline results were computed by always predicting the minority (check-worthy) class. We can see that the bag-of-words model using unigrams and phrases performs significantly better than this baseline. Adding POS-tags and NER-tags improves the model by very little (2.5%). Adding POS-tuples and topic based features improves the F1score to 0.252, which is our best result without using normalization. When normalized text is used as described in the previous section, all the features are re-computed and the models trained again. It is very interesting to see that in all cases, the F1score increases when trained on the normalized text. Our best result reaches an F1score of 0.263, which has primary gain in recall over other models.

The most important features for the classifier are shown in Table 1.8. *unigram* features are generally ranked higher than others. Within features that model topics of discussion, we find that entities representing numbers i.e. percent, money, ordinal and time are important. The high ranking topics are also intuitive and cover several big talking points in the presidential debates. [9]. We also find that part-of-speech tuple features are ranked higher than part-of-speech tags depicting the usefulness of these simple yet intuitive features.

[9]It is to be noted that the topic names are assigned by manual inspection of important words of each topic.

1.7.2 Intra and Inter Party Tests

We evaluate our model performance on inter and intra-party splits of our dataset. For inter party tests we train on all debates of one party and test on debates of the other party. For intra-party tests we perform a leave-one-out cross-validation where each fold is one debate. The results are presented in Table 1.7. The intra-party scores are slightly higher than inter-party scores which is quite intuitive due to the use of unigrams. We also see that overall the model performance is much lower than our cross-validation model performance presented in the ablation study. This could be attributed to the significant decrease in the number of check-worthy samples in the training datasets that happens when splitting party-wise.

1.7.3 Comparison with ClaimBuster

The ClaimBuster [25] system is trained on the statements made by presidential candidates in the debates for elections from 1960-2012. They use only sentences that are more than five words in length. The labels are crowd-sourced through voluntary participants and validated through a set of statements with gold labels agreed upon by three experts. For a fair comparison with ClaimBuster we evaluate on the test set comprising of only the presidential and vice-presidential debates. This set is not used for training of our model.

Classification: To compute the classifier output for ClaimBuster we use their web-api which provides a score in $[0, 1]$ for any given piece of text. Using that score, we classify a statement as check-worthy if it is ≥ 0.5 as used by the ClaimBuster team [25]. We also compute the ClaimBuster results on our training set for completeness. The results are tabulated in Table 1.9. Our model out-performs ClaimBuster on both our training and test sets by 9.1% and 16.8% respectively. It is interesting to observe that although the precision is roughly similar, we show majority of the improvement in recall. This could be attributed to our features that model debate dynamics better and take into account statement context. ClaimBuster treats each

statement separately and hence misses on necessary semantic and contextual information. A clear case of usefulness of our model is well explained in Table 1.4. We see that post-normalization each sentence has a better representation of the associated entities.

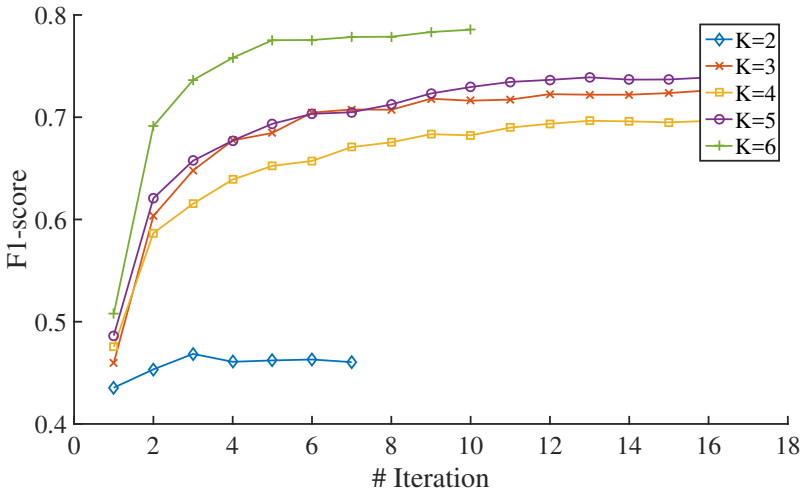
Multi-Classifier Performance We evaluate our multi-classifier system by first training and predicting on the training set for various values of $K \in [2, 6]$. Beyond 7, some of the initial clusters had no $+ve$ samples. We follow the algorithm described in Algorithm 1 and compute training F1-score after each iteration. The results are shown in Fig. 1.2 (a).

The training accuracy increases with the iterations and after a point it saturates. We find that generally with higher K training score saturates faster. After 8 iterations the differences F1-score are negligible in all cases (¡ 0.05). The performance on the test set is shown in Fig. 1.2 (b). We find that only for $K = 3, 5$ there is consistent performance. We conclude that for $K = 3$ the latent groupings are optimal for our classification task, in the sense that they best describe our final output function. We find the best F1-score of 0.214 for $K = 3$, an improvement of 2.4% over Tathya-SVM along with a 28.8% improvement in recall. We call this system Tathya-MULT.

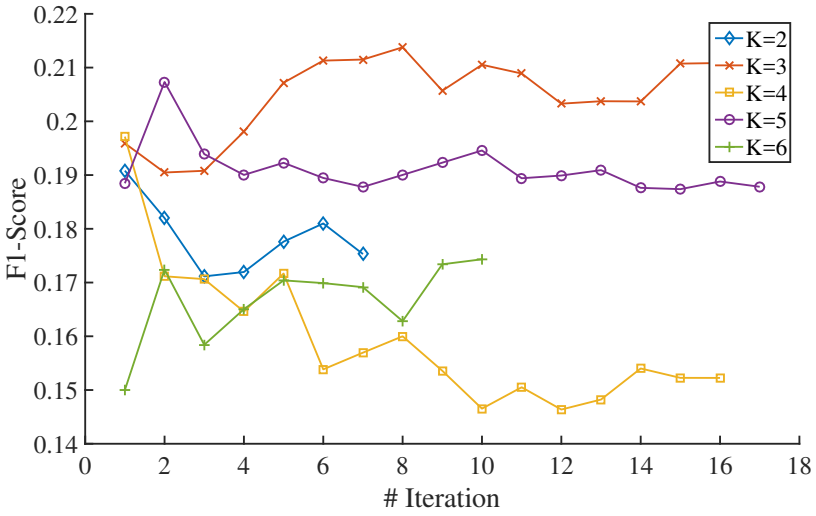
Table 1.9.
Performance comparison on held-out test set of presidential debates from US Presidential Elections 2016.

	P	R	F
ClaimBuster	0.226	0.148	0.179
Tathya-SVM	**0.227**	0.194	0.209
Tathya-MULT	0.188	**0.248**	**0.214**

Ranking: To compute the ranked set if statements in order of their check-worthiness we use the probability scores from our SVM model using Platt's scaling. ClaimBuster provides it's scores using the same technique and we use the scores from their web-api as is for this experiment. The results from the two systems for the top-K ranked



(a) Train (Baseline with Single SVM = 0.41)



(b) Test (Baseline with Single SVM = 0.209)

Figure 1.2. Performance of the multi-classifier system on training (a) and test (b) set respectively.

statements is shown in Table 1.11. The probability scores for being check-worthy are used as relevance scores to compute the cumulative gain. A statement is relevant if it's check-worthy in our gold annotation from fact-checkers.

We see that for most k ClaimBuster has slightly higher precision i.e. out of the top-k check-worthy statements more statements were checked by fact-checkers. However, TATHYA has considerable improvement in ndcg@k scores for all values of k. This implies that according to TATHYA, the high ranked relevant(in our case – check-worthy) statements have very high relevance scores and contribute more towards the cumulative gain. ndcg@k is a metric that is gaining wide acceptance for ranking based on classification models [10] since it captures "non-binary notions of relevance". For fact-checking is time-constrained, a ranking model should give highly relevant results in the top-k as opposed to giving many, but less relevant results.

Human Annotators We compare TATHYA and ClaimBuster against non-expert human annotators in flagging check-worthy statements (described in Section 1.4. The results for the 1st Presidential Debate between Donald Trump and Hillary Clinton are tabulated in Table 1.12. We find the intersection of human annotations perform the worst, showing the poor agreement between them and the fact-checkers. Taking the union of the two human annotations gives a much better F1-score for classifying check-worthy statements. Although ClaimBuster has higher precision, it has dismally low recall as compared to TATHYA. Both the automated systems performs poorly compared to the union of annotations of the untrained humans. This is a little surprising, however, it can be attributed to the fact that even these annotators were well versed in the U.S political scene and would have a mental model of what would be important and check-worthy.

The ClaimBuster team hires annotators to acquire labels as opposed to using those from fact-checkers. These annotators have no economic constraints, and intuitively they would mark many more statements as check-worthy compared to professional fact-checkers. In other words, their model trained on a less constrained dataset should have less false-positives and hence higher precision which we observe in our comparison. Higher recall and ndcg scores for TATHYA suggests we are able to correctly classify many statements that requires contextual information and are highly rele-

[10] https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html

vant according to fact-checker labels, which we consider as a more reliable source of ground-truth.

Table 1.10.
Taxonomy for sources of errors for false-negatives in TATHYA

Category	Explanation	%	Examples
not checkable	these claims are either reference to a future event or an opinion that is not clearly checkable	21.1	1. I don't think General Douglas MacArthur would like that too much. 2. NAFTA is the worst trade deal maybe ever signed anywhere, but certainly ever signed in this country.
sentence context essential	statements like these lack enough context information for our system and would require implicit world knowledge of references mentioned	11.1	1. It got us into the mess we were in, in 2008 and 2009. 2. Saudi Arabia is not accepting one. ('one' refers to refugee from Syria)
unit of checking subjective	certain sentences are marked checked only because appear within context of the main claim, raising a question on whether entire paragraphs be used as units of the study	8.1	1. She talks about solar panels. We invested in a solar company, our country. 2. But stop-and- frisk had a tremendous impact on the safety of New York City. Tremendous beyond belief.
ill-formed sentence	broken text or sentences which are grammatically improper are difficult to analyze linguistically	5.6	1. I don't mind releasing – I'm under a routine audit. 2. And believe me, this country thinks it's – really thinks it's disgraceful, also.
reason unclear	sentences which do not clearly contain a claim or are too vague to check	4.3	1. I'm going to have a special prosecutor. 2. And a lot of really smart, wealthy people know that.
similar claims;different labels	some claims are checked due to their importance in current debate context, however, others that are very similar in syntax and semantics are not	3.1	1. "Senator Sanders voted against the Brady Bill", O'Malley and "I have been for the Brady bill, I have been against assault weapons", Clinton 2. Statement on personal experience e.g., "I had numerous conversations with Sean Hannity at Fox." are sometimes checked.
implication checked	an implication of the sentence is the claim that is checked	2.5	1. Let's be sure we have affordable child care and debt-free college. 2. But you take the gun away from criminals that shouldn't be having it.

1.7.4 Error Analysis

In this section we manually analyze the sources of error due to which our automated detector is likely to make mistakes. We do this for the 1st Presidential debate and study both false-negatives and false-positives. There were 112 checked statements out of which we only predicted 17 to be check-worthy. We categorize the remaining

Table 1.11.
Comparsion of ranking results for the two systems.

	Tathya		ClaimBuster	
k	precision@k	ndcg@k	precision@k	ndcg@k
10	0.300	0.503	0.400	0.416
20	0.250	0.456	0.350	0.388
50	0.300	0.489	0.320	0.372
100	0.290	0.493	0.290	0.360
200	0.270	0.497	0.270	0.357
250	0.264	0.498	0.280	0.372
300	0.263	0.502	0.280	0.381
400	0.252	0.506	0.270	0.388
500	0.258	0.522	0.272	0.420
800	0.232	0.574	0.233	0.499
1000	0.221	0.603	0.219	0.541

Table 1.12.
Comparison on 1st presidential debate: Tathya, ClaimBuster and human annotators

	P	R	F
Humans-Intersection	0.22	0.062	0.097
Humans-Union	0.235	0.285	0.258
ClaimBuster	0.312	0.079	0.126
Tathya	0.252	0.147	0.186

statements into different reasons for why our model could not correctly label them. In cases where it is clear our model is wrong we mark them as such and this accounts for 44% of the cases. The different categories and their frequency of occurrences in percentage are shown in Table 1.10.

For false-positives, on manual inspection we could only determine roughly a third of the statements that do not seem check-worthy e.g., *The 90-minute debate is divided into six segments, each 15 minutes long.*. The remaining samples fall into the check-worthy category according to our definition e.g., *Nine million people – nine million people lost their jobs. Five million people lost their homes. And $13 trillion in family wealth was wiped out.* This helps to emphasize the need to automate fact-checking for meeting real-time constraints and provide exhaustive coverage of check-worthy claims.

1.7.5 Threats to Validity

Tathya is build with an aim to detect check-worthy statements from political discussions and aid in fact-checking. We make certain assumptions on the nature of check-worthiness – a concept which is inherently subjective in nature and define it as tightly possible. Through our error analysis we find that fact-checkers sometimes check statements that do no conform to our definition for check-worthiness e.g., they might check a statement that claims something in the future, since they are able to use extraneous knowledge to predict what the outcome of the event in the claim might be. Another point of concern while training on a dataset aggregated from fact-checking organizations is that it lacks in coverage and entails a risk of under-training our model, which we believe is the case and a reason for our low F1-score. However, with sufficient data samples and possibly information from industry experts, our model can be significantly improved.

1.8 Conclusion

In this chapter, we tackle the problem of detecting whether statements made by politicians are check-worthy or not. Such a determination is the first step in automating fact-checking. We find that this problem is made difficult by a confluence of factors – fact-checkers' subjectivity, understanding dynamics of discussion and incorporating information from world knowledge. We provide a data-driven taxonomy on the hardness of this problem, which exposes the challenges in learning to automate fact-checking and assign a quantitative score to a qualitative notion of check-worthiness.

Acknowledging these difficulties, we focus on exploiting the semantic context and debate dynamics. We design a classifier system that uses features to model these factors and also attempts to learn latent groupings of data. Comparing our system Tathya to the current state-of-the-art, ClaimBuster, on the presidential debates, we find an improvement of 19.5% in F1-score and 67% in recall. Our classifier uses a set of different classes of features—bag-of-words, topic agreement, entity history and targeted part-of-speech tuples. Importantly, we find that normalization by doing anaphora-reference resolution significantly improves the accuracy of the prediction. In future work, we will attempt to learn better latent representations that would enable to increase the expressiveness of the classifier and further improve performance. In essence, using our system, fact-checkers and the general public will be aided with a wider coverage of interesting claims and improve quality of political fact-checking.

2 Nirmalya

2.1 Introduction

Since the inception of YouTube in 2005, it has seen an exponential growth in terms of video content as well as user engagement in a variety of fields including entertainment, advertisement, publicity, news, education, etc. Video-based information sharing is gaining leverage over text-based dissemination and has deep social impact[1]. This fact is corroborated by Alexa's web traffic statistics that lists YouTube as the second most visited page globally.

The growing popularity and economic opportunities for content providers has triggered the creation and promotion of spam campaigns on these platforms. There are various dimensions of this act: posting advertising links, increasing the view count of a video which in turn is monetized by the incentive schemes common in the video sharing sites, sharing illegal content or copyrighted content, disseminating adult content with misleading description, and promoting campaign that may be spurious and meant for phishing. In particular, let us consider this last motivation behind spam videos. One example is fake computer support companies posting videos purporting to be from reputed anti-virus/anti-malware companies and encouraging users to download some software for scanning and disinfecting their computing devices[2]. Another threat vector is users watching a YouTube video falling victim to drive-by-download attacks whereby malware such as Trojans are being downloaded to the user's device[3].

[1] `https://en.wikipedia.org/wiki/Social_impact_of_YouTube`

[2] https://blog.malwarebytes.org/threat-analysis/2013/12/tech-support-scammers-spam-youtube-with-robot-like-warnings/

[3] http://www.spamfighter.com/News-18855-Malware-Attacks-Users-PCs-While-Enjoying-YouTube-Videos-Bromium.htm

YouTube, itself, has classified spam videos into many different categories based on their policy[4]. These include video and comment spam, artificial traffic spam, misleading metadata, misleading or racy thumbnails, scams, and blackmail or extortion. According to YouTube: *"Metadata refers to any and all additional information provided on a video. This includes the title, description, tags, annotations, and thumbnail."* In this paper we focus on detecting spam related to subset of misleading metadata i.e. non coherence between video content and its title and description. *The unsolved problem till now is how to automatically and accurately classify spam videos with sufficient accuracy and robustness.*

This form of spam video classification is important because it is a predominant mode of launching the sample attacks discussed above that aims to trick naïve users into clicking unsafe links. Detecting spam videos is a challenging task as compared to detecting scam or artificial traffic spam. This is because these videos are created with an intent to deceive both the consumers as well as the anti-spam algorithms of YouTube. It is not surprising to see that there are several such videos still at large on the platform and have not yet been flagged even after a period of more than six months, e.g., see Figure 2.1(a) which from its metadata seems to indicate that it is related to the season finale of the highly popular television series "Game of Thrones" but the actual video has only content from previous seasons. In fact, our dataset contains spam videos from the period of 2013 to 2014 that are still not removed.

Most of the work in this domain has been related to detecting spammers or promoters of spam content [26–29] or detecting if response (also video content) to a video is a spam [28]. However, to the best of our knowledge, we are the first to tackle the problem of identifying spam videos with misleading description and title. This covers two of the five categories of spam activity specified by YouTube on its website. We approach the problem by first characterizing spam and legitimate videos through human annotation and creating a rich dataset, which we release publicly to

[4]https://support.google.com/youtube/answer/2801973?hl=en

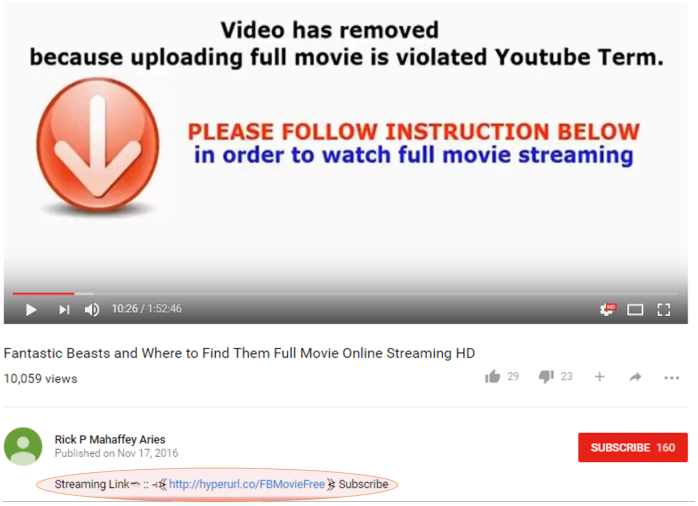
spur further research in this field [5]. Then, we extract a rich set of features for the videos which we categorize into three categories—comments, characteristics of the video, and characteristics of the channel of the posted video—and utilize the set of features in a supervised learning approach for detection. We find that video level and channel level information are good indicators of spam characteristics in many cases, however, they are not sufficient by themselves. The comment activity on a video provides very interesting insights as it captures human feedback on the video veracity, relevance and popularity. For this paper, we exclude features related to the actual video content. Our supervised detection system NIRMALYA is trained and validated on a manually annotated dataset of spam and legitimate videos extracted by crawling YouTube. The high level design of NIRMALYA is shown in the Figure 2.2. A corpus of videos is collected and then filtered to increase the proportion of spam videos. This is then given to human annotators to flag as spam or legitimate. The final labeled set is used to train an ensemble classifier, which is then used at runtime to predict if new videos are spam or legitimate. In summary, the main contributions of this work are:

- To the best of our knowledge, we are the first to tackle the problem of identifying spam content on video sharing platforms with misleading description and title. We do this by creating a novel dataset dataset of 1690 videos containing 161 spam videos curated from a large corpus of 500k.

- We characterize spam activity in terms of comment activity (temporal patterns and textual information) and statistical features describing the video and the uploader's channel. On a balanced set, NIRMALYA achieves the mean F-score of 0.82 with a recall of 0.83 using 4-fold cross-validation compared with three baselines: (1) 24% higher than a classifier that always predicts the legitimate class with F-score of 0.66 (2) 52% higher than multi-variate Gaussian estimation

[5]https://tinyurl.com/n65ew78

(a)



(b)

Figure 2.1. A spam trailer for a popular TV series with almost 1M views. It has misleading tags which will cause it to appear in search results for the actual trailer (which appeared at a later date).

with an F-score of 0.54. (3) 64% higher than random predictor with F-score of 0.5.

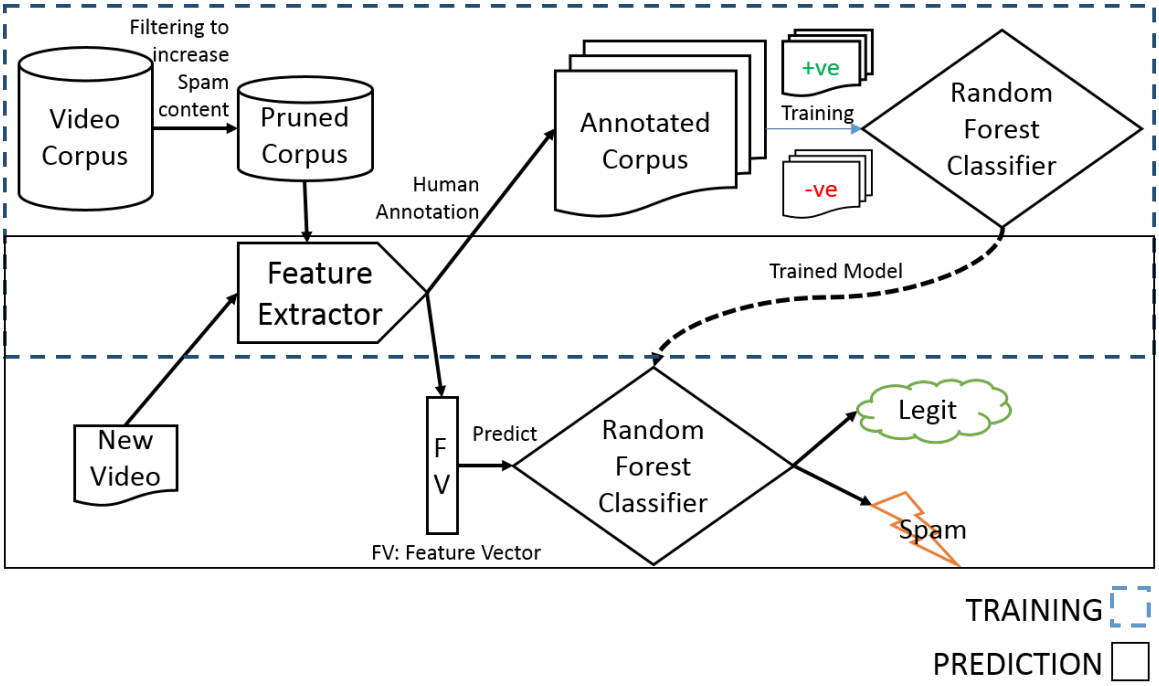


Figure 2.2. NIRMALYA: Workflow showing the training phase and the prediction phase. The training phase, done in batch, generates the Random Forest model, which is then used in the prediction phase, on each newly posted video to determine if it is legitimate.

- Finally, we test the robustness of NIRMALYA by training and testing on non-overlapping datasets, calculate precision and recall on spam and legitimate classes and do error analysis of our technique.

2.2 Building a Reference Dataset

Since there is no publicly available dataset which contains spam videos suiting our problem definition of videos with misleading meta-data, we have created a manually annotated data-set for this task. This section describes the data collection and the annotation process.

Table 2.1.
Dataset description

Video Description
view count, comment count, video duration, video licensed content dislike count, like count, type of thumbnails, publish date category, relevant topics, video channel, Up-loader's Google+ account URL
Channel Description
subscriber count, video count, view count, comment count Up-loader's Google+ account URL
Comment Description
Text, Comment Likes, Commenter's Google+ account URL, Upload Date Modified Date, Reply Count, Video Id, Can Rate

2.2.1 Crawling

Crawling was done using the YouTube REST data API v3. Videos uploaded between September 2013 and October 2016 were crawled. The annotation and the subsequent manual inspection clearly revealed that there were spam videos still present on the site, e.g., a video with the title "PROOF Obama is a Member of the Muslim Brotherhood"[6] was posted on 23rd Jan, 2014. It has $130k$ views, 800 likes and 520 dislikes and a manual inspection of the video reveals that the narrator has no actual proof to prove the point made in the description and the video only serves to attract anxious viewers with the controversial title.

[6]https://www.youtube.com/watch?v=BhDVqrjxwUE

YouTube API segregates the videos into several broad categories out of which we chose a few which we thought would be interesting to study [7]. The region code selected for crawling purpose was set to "US", i.e. all the videos were available in USA. A total of 503,824 videos were crawled, which had 60,962,704 comments on them. The parameters collected for each video as well as the comments are shown in Table 2.1. Apart from video characteristics, the channel information of the video was also crawled and the information available for the channel are also shown in Table 2.1. The information contained in the comments crawled for each video is shown in Table 2.1. It is important to mention here that the API does not allow us to crawl all comments for a particular video. Instead it provides an option to crawl comments based on their relevance to the video or in chronological order and provides a sampling based on this criteria. There is a limit to the number of comments available (of the order a thousand) however we did not find it to be fixed across videos. To overcome this sampling limit, we developed a comment crawler in CSS and python to download all the comments associated with a video to ensure integrity of the features studied. The results of the classifier with all comments proved to be better than the results obtained by using only the comments provided by the YouTube API.

2.2.2 Data Curation and Dataset Creation

The desired properties of the dataset we want to use for our experiments are: 1. Have a representative set of videos which capture the demographics of YouTube i.e. represent videos that are very popular as well as ones that are moderately and mildly popular. However, bias should be towards popular content because they have more potential to cause harm. 2. Since the percentage of misleading content on YouTube is expected to be much lower in comparison to the legitimate content, our dataset should be appropriately biased to have sufficient number of spam videos. Below we

[7]Categories: Film & Animation, Autos & Vehicles, Music, Shows, Pets & Animals, Sports, Travel & Events, Gaming, People & Blogs, Comedy, Entertainment, News & Politics, Howto & Style, Education, Science & Technology, Nonprofits & Activism

describe our data curation process to create a final dataset that we will use for further processing.

With around 500K videos being crawled and possibly very low percentage of spam content, manually annotating each and every video and searching for spam videos was practically impossible. Random sampling from this set is not guaranteed to capture spam videos as number of spam videos will be much lower on such a highly managed platform. Also, a high percentage of videos would fall in the mildly popular category due to the sheer number of videos uploaded everyday on YouTube. Furthermore, randomly collected sample does not necessarily follow the class distribution in the dataset to achieve classification [30]. Hence, we decided to use some heuristics to narrow down the dataset keeping in mind the desired properties described above. First, we decided to create two non-overlapping datasets with respect to date of upload of videos with the motive to train our classifier on one dataset and test it on another. For our training dataset, we chose the videos uploaded between September 2013 to October 2014. The reason for choosing these dates were two fold: (1) we wanted to capture spam videos that were still at large and not detected by YouTube anti-spam algorithms. (2) we wanted our classifier to be trained on data that capture sufficient statistics from metrics described in Table 2.1. For test dataset, we chose the date ranges from September 2015 to October 2016. After categorizing the training and testing datasets in two non-overlapping date ranges, we applied further heuristics on training and the test datasets. For training dataset, all the videos having less than 10,000 views, which is the average views per video for our dataset, were eliminated. This was done to train our classifier on videos having higher visibility and see the effect of testing it on videos with higher and lower visibility. Secondly, videos which had less than 120 comments, which is the average comments per video in our dataset, were eliminated.

On the remaining videos, a smart heuristic was applied as shown in Figure 2.4 to ensure that we remove videos which are more likely to be legitimate. In this method, we first manually identified a handful of spam videos. We then went through

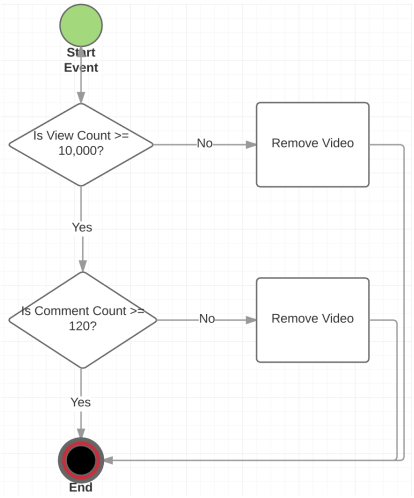


Figure 2.3. Narrowing the data set: Eliminating videos with less views and comments

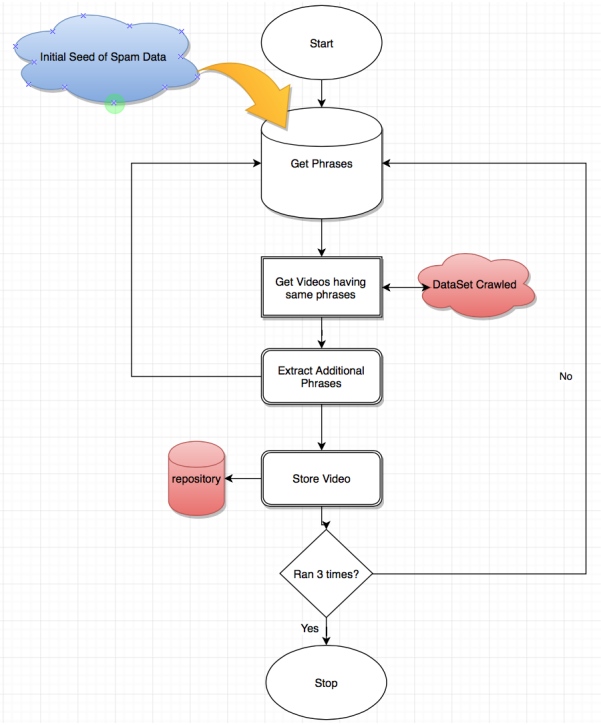


Figure 2.4. Narrowing the data set: Bootstrapping phase to find videos with similar comments

comments posted by different users on these videos and collected a few phrases that were commonly appearing in these comments. Few of these comments were "complete bullshit","fake fake fake" etc. Using these as the initial seed set, we looked for videos in our data set that contained at least 2 comments having phrases from the seed set. Using these new videos, we added commonly occurring phrases in our seed set and continued the process. After 3 iterations, we got a set of 4,284 videos which could potentially have nature similar to the spam content we want to identify. The intuition behind adopting this method is that videos which mislead users by posting spam content are more likely to have phrases which vent anger in the form of swear words and phrases. A similar method was adopted for clustering tweets belonging to a particular rumor chain on twitter in [31] with good effect. After this, another heuristic was used — ratio of dislike count:like count of the video for further filtering. The hypothesis to use such a heuristic was that for spam videos, the number of dislikes would be significant compared to the number of likes. Sorting the videos based on the ratio in non-ascending order and taking videos having ratio greater than 0.3 gave us a final set with 650 videos. For test dataset, we used similar methodology with some relaxed conditions: ratio of dislike count:like count was set to be greater than 0.1, average comments on the video were set to be greater than 50, comments had at least one comment from the seed set and number of views were set to be greater than 3k to capture videos of low popularity. Motivation behind relaxing these condition was to have a more realistic dataset that would closely resemble a random sample of YouTube. After applying these conditions, we got 1040 videos. This set now contains videos that had not caught the eye of large population as well as popular videos. It is important to note that after careful curation process, we have made our classification process harder because the filtering has removed many clearly legitimate videos.

2.2.3 Annotating the Data

For the training dataset with 650 videos, we created an online annotation task where a user was given the link to a video and was asked to mark it as "spam", "legitimate". We also instructed to mark a video as "not sure" if some ambiguity is present in classifying as spam to minimize bias towards incorrect marking. We provided our annotators with the following instruction to identify a spam video: a video which has title or description not relevant to the content of the video is to be considered "spam". We made 33 separate surveys having 20 videos per survey(one having only 10). and gave it for annotation to 20 volunteering participants at our institutions. This task was repeated for a second round of annotation with the same set of annotators. During the experiments, precaution were taken so that no annotator marked the same set of 20 videos in the two rounds. The results of the two rounds of annotation can be seen in Table 2.2. On analyzing the results, we identified that there was lack of unanimous decision on several videos as seen in Table 2.3 which describes the (dis-)agreement between annotators in the two rounds of annotation. Each cell of Table 2.3 can be used to understand the annotator agreement for the different labels. For example, the first cell denotes that 70 videos which were marked spam in round one of annotation were also marked spam in round two of the annotation. We see that inter-annotator agreement was not perfect, an issue which has been reported repeatedly in prior works for annotation tasks in social media [17, 32]. We believe that this was due to fact that our dataset contains videos with at least 10k views and 120 comments. As a result we are dealing with several videos which at the outset may appear to be legitimate due to several reasons, e.g., there is a credible conversation thread on the video, the content looks legitimate at first glance but is actually morphed, or it is uploaded by a channel which looks reputable. Thus, the decision as to whether these videos are spam or legitimate, was subjective and challenging in nature. The discrepancies in the annotations were then resolved by another tie-breaker round. A graduate student volunteer then went through all the

Table 2.2.
Statistics from the two rounds of annotation

	Round1	Round2	Final Annotation
Spam	158	130	123
Legitimate	400	422	423
Not Sure	92	98	104

video content and annotations and if any ambiguity lied in characterizing the video as spam, the video was marked as "not sure". The statistics from the two rounds as well as the final annotation are shown in Table 2.2. The distribution of spam/legitimate in different categories of the pruned dataset is shown in Table 2.4.

For test dataset, we distributed 1040 videos among 52 volunteers with each volunteer getting to annotate 20 videos. After first round of annotations, we got 809 legitimate, 111 spam and 120 not-sure labels. Similar to the previous approach, we did a second round of annotation and used labels only for which both annotators agree. In the end of round 2, we ended up with 38 spam, 856 legitimate and 146 not sure videos. Presence of only 4% spam in our testing dataset down from 20% spam in our training set justifies our pruning process. If we relax the filtering conditions, we are likely to get very low percentage of spam in the dataset and it is extremely difficult to annotate larger datasets, in search of spam videos. From our datasets, there are some categories which were crawled but were completely eliminated in the pruning process because the crawled videos from those categories did not contain enough suspects of spam.

2.3 Features for Distinguishing Spam

Any spam detection technique relies on features which can help distinguishing spam and legitimate by modeling their behavior. In this section we propose three

Table 2.3.
Annotator (dis-)agreement from rounds 1 & 2

	Spam	Legitimate	Not Sure
Spam	70	62	26
Legitimate	54	308	38
Not Sure	6	27	59

Table 2.4.
Category wise distribution of spam

	Spam	**Legitimate**
Entertainment	66	262
News & Politics	41	95
Film & Animation	12	31
Music	4	22
Shows	0	13

classes of features, which we believe can serve this purpose and help to build a robust classifier model.

2.3.1 Channel Level Indicators

These features capture the information about the channel through which a video was uploaded. The hypothesis is that channels which upload more legitimate videos would have much more positive likes, comments and subscribers etc.

- **Comment Count Ratio:** Ratio of total number of comments in that channel to the total number of videos, i.e., the average number of comments per video.
- **View Count Ratio:** Ratio of total number of view of that channel to the total number of videos, i.e., average number of views per video.
- **Subscriber Count Ratio (Video):** Ratio of total number of subscribers of that channel to the total number of videos.
- **Subscriber Count Ratio (View):** Ratio of total number of subscribers on that channel to the total number of views on videos of that channel.The last three features will all measure the average popularity of the videos posted on the channel.

2.3.2 Video Level Indicators

This category of features covers information about the video through the use of the YouTube API. Features picked for the study are following:

- **Video definition:** It can take only two values - HD (High Definition) or SD (Standard Definition). We hypothesize that legitimate videos would have a higher quality definition on average than spam videos.

- **Licensed Content:** It can take only two values - $true$ or $false$. We hypothesize that legitimate videos would be more likely to have licensed content than spam videos.

- **Comment Count Ratio:** Ratio of total number of comments to the total number of views. This feature would measure the level of user interaction with the video.

- **Like Count Ratio:** Ratio of total number of likes for the video to the total number of views. This feature would measure the average likeness rating of the video per view.

- **Dislike Count Ratio:** Ratio of total number of dislikes for the video to the total number of views. This feature would measure the average dis-likeness rating of the video per view.

- **Dislike to Like Ratio:** Ratio of total number of dislikes in the video to the total number of likes in the video. This feature can be an important indicator of average approval rating of the video by its viewers. We hypothesize that spam videos are more likely to have a higher dislike to like ratio.

2.3.3 Comment Level Indicators

Comments allow users to express themselves more freely compared to expressing a binary opinion (like or dislike). Hence we believe that mining comments can give us several indicators to model spam behavior. We obtain all the comments using web scraping and also use the subset of comments available through the YouTube API. Here we describe two kinds of indicators based on textual or temporal patterns.

Linguistic Indicators

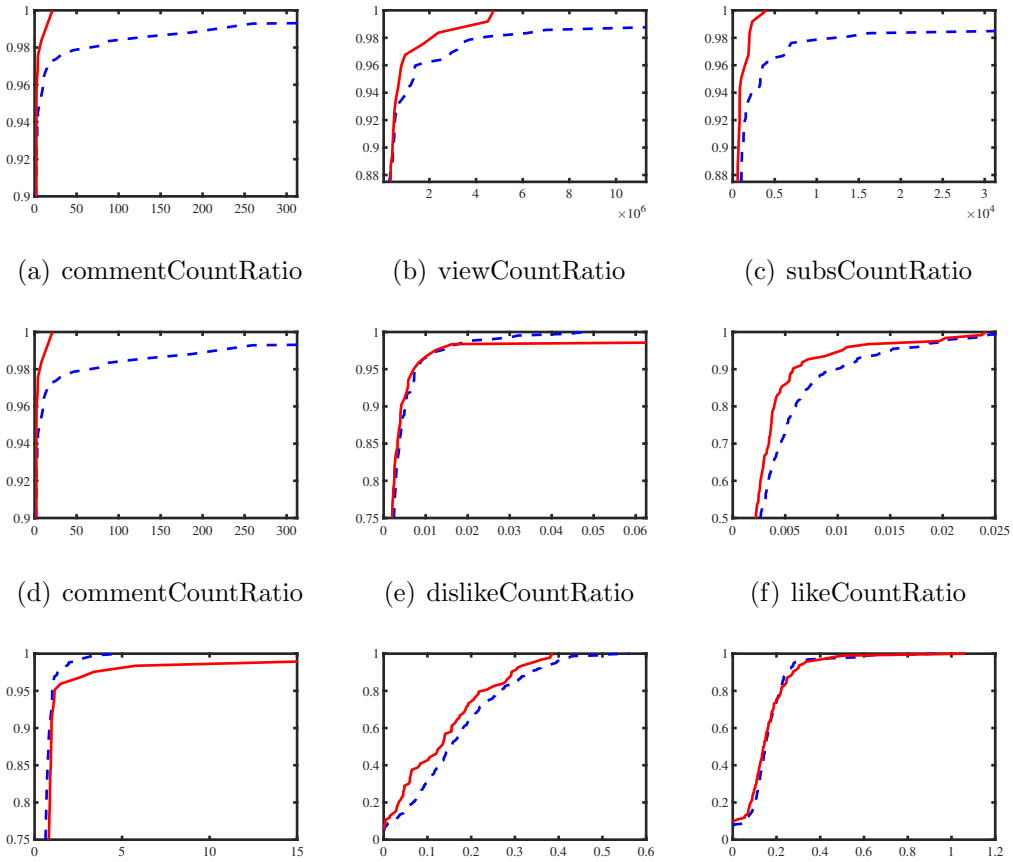
- **Inappropriateness score:** This score is used to capture the extent of inappropriateness in the comments, i.e., number of swear or cuss words used in the comments. To achieve this, we created a set of words using LIWC[8] swear words and their synonyms. We then count the number of uni-grams and bi-grams (concatenated as a single word) present in the set of swear words. The score is normalized by a factor of $2n-1$ where n is the number of words in the comment. This is because there are n uni-grams and $n-1$ bi-grams in a comment of n words.

- **Directness:** This feature captures how directed a comment is toward the video as opposed to directed toward some user (up-loader or another user on the comment thread). Our hypothesis is that comments which were made as a response to some user and not directed toward the video have very little impact in helping to distinguish the legitimacy of the video itself. If the comment contains any user mention in the form <user-name>, then we assign a score of 0 signifying non-directness else a score of 1 signifying directness. Finally, we use what fraction of total comments are directed as the score for this feature.

- **Conversation Ratio:** It is the ratio of number of conversation comments to the total number of comments. A comment is defined to be a conversational if it has at least one reply to it or is a reply itself i.e. it is part of a conversation. We want to analyze whether conversation ratio feature proves to be helpful in categorizing a video as spam or not.

- **Similarity score:** We create two bag-of-words models – for the video description and the comment in consideration. Video description contains the words used in video title, description and tags. We then compute the Jaccard similarity between these two sets of words and report it as the similarity score. We

[8]http://liwc.wpengine.com/



(a) commentCountRatio (b) viewCountRatio (c) subsCountRatio

(d) commentCountRatio (e) dislikeCountRatio (f) likeCountRatio

(g) dislikelikeRatio (h) directness (i) conversationRatio

Figure 2.5. Cumulative % of spam (solid red) and legitimate (dashed blue) groups vs. feature value. (a), (b), (c) are Channel level features. (d), (e), (f), (g) are Video level features. (h) and (i) are Comment level features. It can be observed that in all features denoting video popularity, legitimate group have higher feature values for same cumulative % as compared to spam group(and vice-versa for other features which denote non-popularity)

believe that this can also capture the relevance of a particular comment towards the video.

Temporal Indicators

Arrival rate of comments on videos can be a good indicator for spam and legitimate videos. The arrival rate behavior could be starkly different at different phases in the lifetime of a video based on whether it is spam or legitimate. For example, spam videos with racy content are more likely to have a lot of hits during the initial phase followed shortly by comments which refute the video content. On the other hand, the legitimate videos may also have hits in the initial phase but not a similar comment pattern. Legitimate videos are also expected to have a steadier decline in comment rate as compared to spam videos. This is because spam videos will likely have accumulated a high number of dislikes very quickly and hence people are less likely to visit and comment on them. This kind of behavior has also been observed and used in prior art to create models for detection of anomalous entities on social networks [33]. We created a feature vector of size 365 representing 365 days since the date of upload. The value for each feature is set to the comments posted on that day normalized by total number of comments on that video. Through this we aim to capture the growth of comments on a video over a period of one year.

Binned Features : For 'Inappropriateness' and 'Similarity' scores, we further wanted to capture the distribution of scores in the video of a particular class by creating bins. Each bin's index signifies a fraction of the range of the score and the value in that bin is the number of comments with its score in that range. We tested our classifier with different bin sizes ranging from 1 to 10 and optimal number of bins was found to be 3. This may be attributed to the fact that when we use more bins, the distribution across bins becomes more sparse and there is less distinction between the legitimate and the spam distributions across the bins.

Comment Sets The comments of a video are further segregated based on several parameters that we believed would be useful to gauge the importance of comments in classifying the spam and legitimate videos:

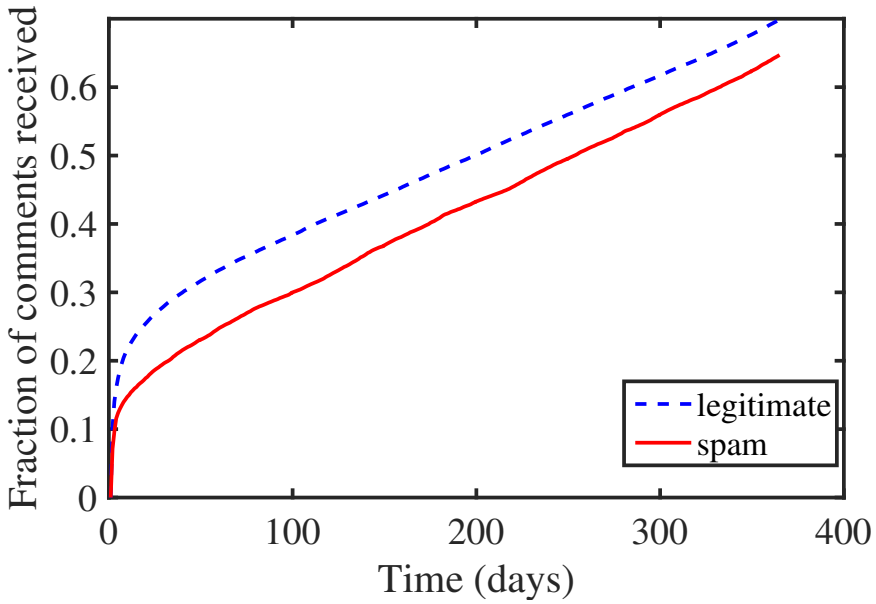


Figure 2.6. Growth of comments with time for Legitimate and Spam videos

- **Relevant comments:** This set contains only the comments when crawled using the *relevance* parameter specified in the API. This parameter is used by YouTube to rank comments by relevance. It should be noted here that this parameter is not available when all comments are crawled using web scraping. But all comments crawled does contain the relevant comments as its subset and yield better results as compared to only relevant comments.

- **Liked Comments:** This set contains comments that had at least 1 like. We think that liked comments carry a sense of approval with them since they have been liked by peers.

- **Liked, Relevant Comments:** Comments that had at least 1 like and were also relevant as per the API.

- **All Comments:** The set of all comments crawled for a video.

2.4 Experimental Results

In this section, we describe our experimental framework in detail and present the classifier results on our training set for each class of features and finally with all the features combined. We then train our classifier using training dataset and evaluate its performance on our test dataset in section 2.4.6. We now present the results for training dataset in subsequent sections. For all our experiments, we created a balanced dataset by randomly selecting 123 videos from 422 videos marked legitimate and 123 spam videos that were annotated as such. We experimented with several learning methods using the scikit-learn library including SVM, Decision Trees, AdaBoost, Gradient Boosting and Random Forest (RF). The RF algorithm consistently performed better than the rest. All our experiments were subsequently performed using RF as the classification tool. Further, all the experimental results are presented using 4-fold cross validation. In each run, original set is partitioned into 4 sets with 3 used for training and 1 used for testing. In 4-fold validation, this is repeated four times with testing on each subset exactly once and then averaging the results. We repeated 4-fold validation experiments for 50 times with random seed for shuffling each time and then averaging the results. We have presented the best and the mean results for classification.

Table 2.5.
Contrast of mean values of features that measure video quality for spam and legitimate videos

Features	Spam	Legitimate
% HD definition	57%	73%
% Licensed content	65%	77%

2.4.1 Evaluation Metrics and Baselines

For the training and test datasets, we use precision, recall, and F-score as the evaluation metrics for classification which are used according to their standard definitions. We will also look at the confusion matrix resulting from these metrics for test dataset. Since there was no prior baseline for comparison, we have created a synthetic baseline comparison algorithm. For this, we assumed that both spam and legitimate video are generated from different multivariate normal distributions. For training, we use the Gaussian Kernel Density Estimation method and learn the density functions from which these two classes may potentially arise. For testing we calculated the probability of each data point arising from spam kernel density function and legitimate kernel density function and assigned it to the class with the higher probability. After 4-fold cross validation we got Precision and Recall for spam class to be 0.5 and 0.58 with a F-Score of 0.54. In addition to this baseline, we also used the classifier that always predicts the majority class, which gives an F-score of 0.66. We also used the random predictor that gives a baseline F-score of 0.5.

2.4.2 Macro Evaluation and Comparison with Baselines

We experimented by using the different categories of features explained in Section 2.3 and finally using all the features together to form the macro benchmark evaluation. The results are reported in Table 2.6. For the mean F-score reported, standard deviation was within 2% for all the models except for the Comment Liked + Relevant feature for which it was 4%. From the table we see that all variants of Nirmalya have mean F-scores better than the random baseline. This includes using Channel features, Video features, and Comment features individually.

In the comment features, we use various classes of comments, such as 'Liked', 'Relevant' etc. We see that using all comments, which were not available from YouTube API but rather, only through our web scraping technique, yields better results than only using the Liked + Relevant comments from the YouTube API. This confirms

Table 2.6.
Classification results for different models. 'F', 'P' and 'R' refers to F-score, Precision and Recall respectively. For comments, 'L' means Liked subset and 'R' means Relevant subset.

Model	**F**	**P**	**R**
Random Prediction Baseline	0.5	0.5	0.5
Majority Class Prediction Baselie	0.66	0.5	1.0
Gaussian Baseline	0.54	0.5	0.58
Video Features	0.58	0.60	0.56
Channel Features	0.62	0.62	0.62
Comment Liked Features	0.52	0.52	0.53
Comment Relevant Features	0.57	0.62	0.53
Comment Liked + Relevant Features	0.51	0.50	0.52
All Comments	0.557	0.566	0.555
Video, Channel, Liked + Relevant (first month comments only)	0.686	0.699	0.715
Video, Channel, All Comments (All available comments first year)	**0.82**	**0.83**	**0.83**

our hypothesis that all comments does capture a deeper representation of user interaction in determining spam or legitimate content. But still, none of these features alone yield significantly better results than the baselines described above. Finally, we see that the best results are obtained by combining all comments from 1st year with Video and Channel features that yield a mean F-score of 0.82 with high precision and recall. This shows that any feature themselves are not enough and by combining all features together, our classifier performs significantly better than all baselines.

2.4.3 Feature Importance Ranking

In order to verify the importance ranking of these features we compute top ten features from standard RF feature selection method provided by scikit learn library. We compute the importance score of each feature in every iteration of cross-validation. We then average the scores for each feature and then provide the top ten features in order of importance in Table 2.7. Previously, we observed that including comment features significantly improves the results. This is also evident if we look at the importance ranking of the features in table 2.7. Two of the comment features, *ratioDirected* and *ratioConversation*, are among the top 3 important features as per the RF feature ranking. It confirms our intuition that videos with spam content are less likely to have conversations but will have most comments directed towards the video criticizing it. Table 2.7 also shows that the like and dislike statistics of the video are good indicators of the spamicity of the video because the dislike-to-like ratio and average number of dislikes per video feature high in the list. However, it is important to note that a legitimate but unpopular video may also have such statistics and hence video features are not sufficient by themselves.

2.4.4 Temporal Characterization of Nirmalya

From the last section, we saw that using comment features helped improve our detection model. We now study the temporal behavior of Nirmalya with respect to comments. An important issue in such a problem is detecting spam at an early stage of upload and thwarting them. However, the video statistics which performed quite well in the previous section may not be available for early detection (say within 1 month) since the like, dislike and comment statistics for a video will only stabilize after several months of upload.

Here we characterize the performance of our classifier first by using all comments features only and then by considering the comments for a specific period of time from the upload date of the video. Again 4-fold cross validation is ran 10 times and

Table 2.7.
Feature importance ranking. 'ch','v','sub','conv' refers to channel, video, subscriber and conversation respectively.

Importance Ranking	**RF Feature Selection**
1	videoDislikeLikeRatio
2	commentRatioConversation
3	commentRatioDirected
4	videoCommentCountRatio
5	VideoDislikeCountRatio
6	SubscriberCountRatio(video)
7	commentDay1
8	commentDay324
9	videoLikeCountRatio
10	channelViewCountRatio

the results are reported in Table 2.8. Standard deviation lies within 2%. It can be observed that in 10 days NIRMALYA achieves an F-score of 0.61 which is 13% better than the Gaussian baseline protocol. One use case of this temporal study is that content providers can characterize spam videos during early days of upload and manually investigate the content of such videos before it causes more harm. Still, only using the comments does not perform better than our majority class prediction baseline. This temporal characterization strengthens the fact that we need to use all the features to get best classification results.

2.4.5 Category-wise Performance of NIRMALYA

We also want to evaluate the robustness of using a system like NIRMALYA on the different categories of videos present in YouTube for spam detection. We use

Table 2.8.
Temporal performance using only all comments features.

Days from Upload	**F-score**	**Precision**	**Recall**
1	0.58	0.60	0.56
5	0.59	0.61	0.58
10	0.61	0.62	0.60
15	0.57	0.59	0.55
30	0.57	0.59	0.54
60	0.51	0.54	0.53
90	0.55	0.56	0.54
180	0.57	0.56	0.59
360	0.58	0.57	0.59

the same categories as shown in Table 2.4. For such an evaluation we decided to use a leave-one-out policy — train a model using videos from all categories except a particular category (say *News & Politics*) and then test on videos of that category. For the same category we also train a model containing videos from all categories (including itself) and then compare the results. We show the results in Table 2.9. For both experiments the RF learner with the same settings as in earlier experiments was trained and tested for the top 2 categories (categories were ranked by the significant number of videos in our dataset). The other categories had too few videos to be useful for any statistical significance. The results are shown in Table 2.9. It can be seen that a model trained without including a particular category and testing on that categories performs much worse than the model trained on all categories. We think that this could be explained by the observation that the nature of channel, video and comment characteristics change significantly across categories and therefore a model which is not trained on a particular category does not perform optimally on it. This phenomenon has been observed in a different yet related problem of detecting fake

product reviews on Amazon as studied in [34]. The authors address a similar problem in developing a well generalizable model across "Hotel", "Restaurant" and "Doctor" categories and find that the same model does not apply across these three categories. We therefore are able to see that NIRMALYA performs better when the training set includes videos from that category.

Table 2.9.
Category level performance of NIRMALYA on the top 2 categories. P, R, F represent the best Precision, Recall and F-Score for leave-one-out policy. P*, R*, F* is the best result obtained by training on all categories. For both experiments a Random Forest model was used with 4-fold cross-validation.

Category	P	R	F	P*	R*	F*
News & Politics	0.62	0.63	0.63	0.76	0.70	0.73
Entertainment	0.58	0.56	0.57	0.60	0.61	0.60

2.4.6 Evaluation on Test Dataset

To test the robustness of NIRMALYA, we train it on the training dataset and test it on the testing set. Instead of using all features, we will use the results of the features here related to the Video, Channel and All comments for first month. We limit the comments to one month, because for the videos uploaded in second half of 2016, we do not have first year comments available yet. So, to keep uniformity across the dataset, we will use the results of the said feature in this section. Our test set contains 856 legitimate videos and 38 spam videos (only 4%). We trained NIRMALYA on imbalanced training dataset (123 spam + 422 legitimate videos) and it predicted all the videos to be legitimate in the test dataset. If we use the original set without balancing, then classifier will favor legitimate videos because there are more of them. So, NIRMALYA was trained on balanced dataset of 246 videos (123 spam + 123 legitimate) from the training data set. We ran it ten times, because in

each iteration, the sub-sampled videos from the balanced dataset will be different. For ten iterations, if a spam video is predicted as spam in at least 6 runs out of 10, we mark it as correctly predicted. The average results for the ten iterations are presented in the Table 2.10. It can be seen that the classifier is able to catch 53% of spam videos and misclassify around 47% of spam videos. On spam class, NIRMALYA achieves an F-score of 0.12 which is 71.4% higher than the baseline classifier that always predicts legitimate class. Recall on spam class is 53% and precision is 75% higher than the baseline of 0.04. On the other hand, NIRMALYA classifies 67% of the legitimate content at the cost of misclassifying 33% legitimate videos as spam. It achieves a recall of 67%, precision of 97% and an F-score of 0.79 on legitimate class.

The fact that our training dataset contains only 4% of spam videos with a lot of videos having insufficient statistics in terms of low number of views, low number of comments etc., NIRMALYA is able to achieve in an accuracy of 66%. Also, using all comments from first year will further improve the results. Practically, NIRMALYA can be used by the content providers as starting point to reduce their efforts of manually flagging the videos. They can initially flag all spam classified videos to warn the viewers of possible spam content. And after manually checking, they can clear the flags from the videos which are actually legitimate. System owners or content provider can characterize around two third of legitimate videos correctly. The goal might be to review the statistics of legitimate content or provide ads on those legitimate videos for monetary benefits.

Table 2.10.
Classification of spam and legitimate videos on test dataset

		Predicted	
		Legitmate	**Spam**
True	**Legitimate**	**571(67%)**	285(33%)
	Spam	18(47%)	**20(53%)**

2.4.7 Error Analysis

Referring to Table 2.10, we see a misclassification rate of 47% for spam videos. In order to analyze why our classifier was unable to catch those spam marked videos, We had to manually investigate the uploaded videos. Some interesting insights were gained from the manual inspection. For example let us look at "The Flat Moon over the Flat Earth" video [9] , then we will provide general observations for all the spam videos that were misclassified as legitimate. Here the video uploader is claiming that moon is flat. This channel has around 106k subscribers. The reason for misclassification becomes clear when we investigate the statistics in context of our classifier's top features given in Table 2.7. First, the video has around 2100 likes and only around 650 dislikes. That gives a dislike to like ratio of of 0.3. Second, the channel is extremely popular with most videos having views greater than 10K. Third, we can see that most comments have long conversations and most of the comments are actually corroborating the uploader's claims. So, analyzing these factors, we come to a conclusion that our top features namely vDislikeLikeRatio, commentRatioConv, commentRatioDirected, vDislikeCountRatio and vLikeCountRatio and feature related to channel are behaving exactly the way that they should behave for legitimate videos. That is why our classifier is unable to classify this video as spam. Another observation for misclassified videos was that they contained very high number of dislikes as compared to likes but channel was extremely popular (with greater than 100K subscribers). In this case, dislike to like ratio (although the top discriminating feature) alone cannot help classify the video as spam because other channel, video and comment features dominate the decision of classifier. Some misclassified videos had roughly equal number of likes and dislikes and the channel was not very popular. They also contain very small conversation ratio and directed comments thus giving insufficient statistics to extract meaningful features to classify them correctly.

[9] https://www.youtube.com/watch?v=fH7BjIzXWOg

After doing error analysis on all misclassified spam videos, we also wanted to gain insights on why legitimate videos were classified as spam. Among the 285 legitimate videos classified as spam, we randomly sampled 5 videos and extracted general insights from them. Some of them were fan made videos of popular films or games. Although the video uploader described it in the description, but still those videos garnered very high number of dislikes as compared to likes. Also for some videos, channel features, commentRatioConv and commentRatioDirected had the similar behavior as observed for spam thus misleading the classifier into believing them as spam. However, further analysis is required to develop deeper insights.

2.5 Conclusions and Future Work

In this chapter, we presented NIRMALYA, a supervised learning framework to detect spam videos in online video sharing portals such as YouTube. Spam videos are defined as those having misleading metadata, in terms of title and description being unfaithful to the content of the video. Extensive experiments confirm that using a variety of video, channel, and comment features, NIRMALYA could detect the spam videos with a recall of 0.83 and an F-score of 0.82. Future work will involve doing a deeper analysis of the comments' contents to extract features and patterns indicative of the spam videos. We also plan to utilize processing of video frames to help in spam content detection. Further improvement can be made by analyzing transcripts of speech from YouTube API's and combine fact checking techniques to help improve the detection purpose.

REFERENCES

REFERENCES

[1] Andreas Vlachos and Sebastian Riedel. Fact checking: Task definition and dataset construction. *Association for Computational Linguistics*, page 18, 2014.

[2] Naeemul Hassan, Bill Adair, James T Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. The quest to automate fact-checking. *Computation and Journalism Symposium*, 2015.

[3] You Wu, Pankaj K Agarwal, Chengkai Li, Jun Yang, and Cong Yu. Computational fact checking through query perturbations. *ACM Transactions on Database Systems (TODS)*, 42(1):4, 2017.

[4] James Thorne and Andreas Vlachos. An extensible framework for verification of numerical claims. *European Chapter of the Association for Computational Linguistics 2017*, page 37, 2017.

[5] Andreas Vlachos and Sebastian Riedel. Identification and verification of simple claims about statistical properties. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2596–2601. Association for Computational Linguistics, 2015.

[6] Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 129–136. Association for Computational Linguistics, 2003.

[7] Lucas Graves, Brendan Nyhan, and Jason Reifler. Understanding innovations in journalistic practice: A field experiment examining motivations for fact-checking. *Journal of Communication*, 66(1):102–138, 2016.

[8] Naeemul Hassan, Chengkai Li, and Mark Tremayne. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1835–1838. ACM, 2015.

[9] Terry Flew, Christina Spurgeon, Anna Daniel, and Adam Swift. The promise of computational journalism. *Journalism Practice*, 6(2):157–171, 2012.

[10] Sarah Cohen, James T Hamilton, and Fred Turner. Computational journalism. *Communications of the ACM*, 54(10):66–71, 2011.

[11] Julien Leblay. A declarative approach to data-driven fact checking. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 147–153, 2017.

[12] Jeff Pasternack and Dan Roth. Knowing what to believe (when you already know something). In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 877–885. Association for Computational Linguistics, 2010.

[13] Marco Lippi and Paolo Torroni. Context-independent claim detection for argument mining. In *International Joint Conference on Artificial Intelligence*, volume 15, pages 185–191, 2015.

[14] Isaac Persing and Vincent Ng. End-to-end argumentation mining in student essays. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, 2016.

[15] Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web*, pages 613–624. International World Wide Web Conferences Steering Committee, 2016.

[16] Rada Mihalcea and Carlo Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the 2009 International Joint Conference of Natural Language Processing (ACL-IJNLP)*, pages 309–312. Association for Computational Linguistics, 2009.

[17] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics, 2011.

[18] Stephan Greene and Philip Resnik. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of The North American Chapter of The Association for Computational Linguistics*, pages 503–511. Association for Computational Linguistics, 2009.

[19] Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers – Volume 2*, pages 171–175. Association for Computational Linguistics, 2012.

[20] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of Machine Learning research*, 3(Jan):993–1022, 2003.

[21] Prateek Jain, Manav Ratan Mital, Sumit Kumar, Amitabha Mukerjee, and Achla M Raina. Anaphora resolution in multi-person dialogues. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004.

[22] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

[23] Y-Y Chou and Linda G Shapiro. A hierarchical multiple classifier learning algorithm. *Pattern Analysis & Applications*, 6(2):150–168, 2003.

[24] Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. Discriminative learning over constrained latent representations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of The Association for Computational Linguistics*, pages 429–437. Association for Computational Linguistics, 2010.

[25] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. 2017.

[26] Ashish Sureka. Mining user comment activity for detecting forum spammers in youtube. *arXiv preprint arXiv:1103.5044*, 2011.

[27] Fabricio Benevenuto, Tiago Rodrigues, Virgilio Almeida, Jussara Almeida, Chao Zhang, and Keith Ross. Identifying video spammers in online social networks. In *Proceedings of The 4th International Workshop on Adversarial Information Retrieval on The Web*, pages 45–52. ACM, 2008.

[28] Fabrício Benevenuto, Tiago Rodrigues, Virgílio Almeida, Jussara Almeida, and Marcos Gonçalves. Detecting spammers and content promoters in online video social networks. In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 620–627. ACM, 2009.

[29] Vlad Bulakh, Christopher W Dunn, and Minaxi Gupta. Identifying fraudulently promoted online videos. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 1111–1116. ACM, 2014.

[30] Gary M Weiss and Foster Provost. The effect of class distribution on classifier learning: an empirical study. *Rutgers University*, 2001.

[31] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015.

[32] Hila Becker, Mor Naaman, and Luis Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of The 3rd ACM International Conference on Web Search and Data Mining*, pages 291–300. ACM, 2010.

[33] Bimal Viswanath, M Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Towards detecting anomalous user behavior in online social networks. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 223–238, 2014.

[34] Jiwei Li, Myle Ott, Claire Cardie, and Eduard H Hovy. Towards a general rule for identifying deceptive opinion spam. In *Association of Computational Linguistics (1)*, pages 1566–1576. Citeseer.