Identification of Spam Comments using Natural Language Processing Techniques

Cristina Rădulescu
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

Mihaela Dinsoreanu
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

Rodica Potolea
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

***Abstract*—The high popularity of modern web is partly due to the increase in the number of content sharing applications. The social tools provided by the content sharing applications allow online users to interact, to express their opinions and to read opinions from other users. However, spammers provide comments which are written intentionally to mislead users by redirecting them to web sites to increase their rating and to promote products less known on the market. Reading spam comments is a bad experience and a waste of time for most of the online users but can also be harming and cause damage to the reader. Research has been performed in this domain in order to identify and eliminate spam comments. Our goal is to detect comments which are likely to represent spam considering some indicators: a discontinuous text flow, inadequate and vulgar language or not related to a specific context. Our approach relies on machine learning algorithms and topic detection.**

Keywords—Spam; Machine learning; Co-occurrence; Topic extraction; Post-comment similarity; Feature vector, Opinion, Sentiment.

I. INTRODUCTION

The social networks of the modern web allow users to communicate with each other by expressing their opinions in the form of comments related to some given posts. It is easier to read online reviews of products or to read online advice about trips, then to ask friends or relatives on their opinions. However, such online reviews can be spam reviews that do not provide useful content. The purpose of spam comments is to promote content or services that have little rating or to promote products which are less familiar on the market. An example of a social media site which allows users to communicate and share opinions by means of comments is YouTube. The site provides a method to eliminate a spam comment if it receives a high number of negative votes. This mechanism offers protection to the YouTube user community so that offensive or off-topic comments are no longer displayed to them. However, until the comments receive the required number of down votes they are still displayed. Clearly, due to the continuing and increasing stream of comments, it is impossible for the community to read and rate all the comments. The methods we present in this paper can help to automatically filter spam comments. In our research we aim to identify different types of spam comments based on various features such as: discontinuous text, inadequate and vulgar content and comment topic is not related to the specific context. Our approach is to first eliminate comments which contain discontinuous text, inadequate text and vulgar language. Types of comments with discontinuous text occupy much space in the web page and comments with vulgar language are disturbing and upsetting to the users. By discontinuous text we refer to comments which contain an increased number of punctuation marks, white spaces, capital letters, non ASCII characters, digits and new lines. After we eliminate these types of comments we detect spam comments which are not related to the specific context of the posting. For this purpose we propose a topic identification method and a topic similarity metric. Based on the stated principles, we built a system to detect spam comments by enforcing natural language processing techniques and machine learning algorithms. We evaluated our system on a benchmark dataset and compared the performance of three classification methods.

II. RELATED WORK

The initial research in the domain of spam detection was in the area of email spam. Sahami et. al. [1] used probabilistic learning methods in conjunction with a notion of misclassification cost to produce filters which are especially appropriate for the nuances of this task. They showed that by considering domain-specific features of this problem, in addition to the raw text of email messaged accurate filters can be produced, Carreras et. al. [5] developed a method to prove that AdaBoost is more effective than Naïve Bayes and decision trees. In the late 1990s, spam started to become a serious problem of the World Wide Web. Methods of web-spam filtering based on content analysis were further explored to detect spam pages. The decision trees algorithm was used in [4] by Davison to identify link based web-spam. He recognized and eliminated links between pages which were present for reasons other than merit using heuristics to drop "internal" links. Drost et. al. [10] used SVM to classify web spam with content based features. They formulated the problem of detecting link spam and proposed a solution for generating training data by revealing the effectiveness of classes of intrinsic and relational attributes. They used the following web

spam techniques: link farms, link exchange services, guest books and discussion boards.

In [2] Mishne et. al. used machine learning techniques and probability distributions over strings for the detection of spam comments. They compare the language models used in the blog post, in the comment and in the pages linked by the comments. In [3] Bhattari et. al. used the data corpus from the same set to detect spam comments from blogs and sites. In [6] the authors present an in-depth study of commenting and comment rating behavior on more than 6 million comments from YouTube videos. They make a comparison to see how the sentiment expressed in a comment influences the rating of the comment. The predictions can be used to indicate how predisposed a comment is to being low-quality and adjust the down vote comment accordingly. In [7] Ruihai Dong et. al. provide a method for topic extraction using natural language processing techniques. They use part of speech rules and collocation patterns to extract the topic from a text by retrieving a set of bigrams and unigrams.

III. OUR APPROACH TO COMMENT SPAM IDENTIFICATION

In our research we show that it is possible to detect spam comments with the proper selection of features which capture different characteristics of legitimate comments in order to differentiate them from spam comments. In our approach we consider as spam the following type of documents associated with a review: (i) incoherent comments with increased number of punctuation marks, new lines, stop words, non ASCII characters and white spaces, (ii) inadequate comments which contain offensive words and (iii) coherent comments which do not provide relevant content to a specific topic. Our approach makes use of natural language processing techniques in order to identify the relevant features of spam comments. We propose a supervised learning approach and experiment different sets of features to correctly classify comments as spam or not.

A. System architecture

The architecture of our system is composed of three main modules: Feature extraction module, Post-comment similarity module and Topic extraction module. In the first step of our solution we eliminate comments which are discontinuous, incoherent and contain vulgar expressions. The identification of these types of comments relies on the identification of countable features such as: links, white spaces, sentences, punctuation marks, word duplication, stop words, non ASCII characters, new lines, capital letters and offensive words. For each feature a normalized numeric value (its frequency) is computed and all normalized frequency entries build together the feature vector. It is necessary to perform normalization of the results in order to have the possibility to compare the numeric values with the threshold values from [3] and [9] in order to classify the comment as being spam or legitimate.

The feature extraction module makes an initial classification between spam and legitimate comments and it receives as input a labeled data set of natural language texts and a list of stop words. Preprocessing is performed on the texts with the following operations: stop words removal and tokenization. In order to examine if the comment and its associated post discuss the same topic we implemented the topic extraction module and we calculated the topic similarity degree between a comment and a post.

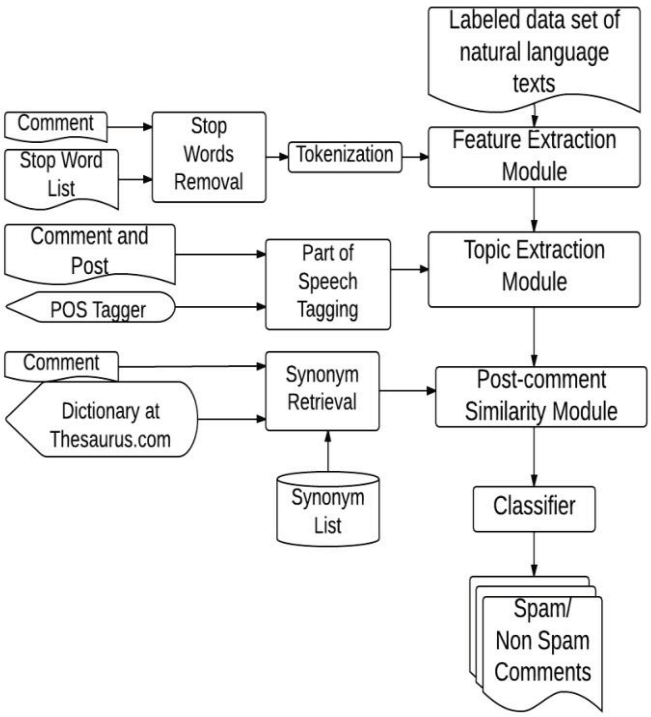

Figure 1: Architecture of the spam detection system

The topic similarity degree is based on the following metric: the normalized value of the sum of the frequencies of the topics from the comment in the post. We understand by topic the main theme which is discussed in the comment and it is given by a set of unigrams and bigrams with a predominant number of occurrences in the comment. We define a topic by using the following collocation patterns described in [7]: a noun followed by a noun or an adjective followed by a noun. We are interested in extracting topics based on these collocation patterns because the intuition is that nouns that frequently occur in a review and are associated with other nouns or adjectives expressing sentiment rich, opinion laden words are likely to be good topics the reviewer is writing about. This topic extraction process is carried out by combining a number of complex natural language processing techniques that involve morphologic, syntactic and semantic analysis of a text. We perform the following operations: part-of speech tagging, information extraction, word segmentation, sentence segmentation and topic detection. The topic extraction module receives as input a comment and a post. In order to examine if the comment and its associated post contain similar topics we implemented the post-comment similarity module. The post-comment similarity module detects whether the comment and the post contain similar topics based on the following similarity metric: the normalized value of the sum of the frequencies of occurrences of each word and its synonyms from the comment in the post. In order to compare the topics of a comment and a post we determine if their content is defined by the same unigrams or by synonyms of the unigrams. The post-comment

similarity module uses a synonym retrieval sub module which connects to the online dictionary, Thesaurus.com to retrieve all the synonyms for the words in the comment. After the synonym retrieval process the post-comment similarity formula is applied and the post-comment similarity degree is calculated.. The three main modules build together a model which is used to identify spam comments. The flow of the spam detection system is shown in Figure 1. The labeled data set of reviews/comments in natural language is the data corpus belonging to Mishne et. al. from [2] with 1024 comments which are both spam and legitimate. Moreover, the evaluation data corpus is retrieved from YouTube web site[1] by using the YouTube data API and from the Daily Telegraph[2] news web site. We created our test collection by formulating queries in a similar manner in which the user interacts with the site to retrieve comments and posts from different videos. The queries are formulated to perform searches for related videos so that we retrieve comments and posts from different videos but in the same domain. The domains used in our research are: politics, travelling, music, hobbies and sports. We manually classified the comments into spam and legitimate comments. We established that a comment is a legitimate comment or a spam comment based on the three criteria mentioned above. Therefore, we created our own evaluation data corpus. We investigated the performance of three different classifiers: Naïve Bayes, Support Vector Machines (SVM) and Decision Trees (DT).

B. Components of the system

1. Feature Extraction Module

In [3] and [9] the following characteristics of spam comments are defined: number of links, number of whitespaces, number of sentences, number of punctuation marks, word duplication, stop word ratio, number of non ASCII characters, number of capital letters and number of new lines. We calculate these numeric features for each comment in order to capture different characteristics which can better differentiate a spam comment from a legitimate comment.

We implemented the feature extraction module based on the characteristics of spam comments given in [3] and in [9].

a. Number of links in the comments

Spam comments tend to have an increased number of links due to the fact that spammers want to redirect users to web sites which have a small number of visitors so that the rating of the web site is increased. Moreover, spammers provide links to promote products which are not very popular or unknown to the market. That is why the number of links in the comments must be taken into accounts when detecting spam comment.

b. Number of white spaces in the comments

Spam comments tend to have a large number of white spaces so that the comment makes much more of an impact on the user who reads it.

c. Number of sentences in the comment

The number of sentences in a spam comment is smaller than the number of sentences in a legitimate comment due to the fact that legitimate comments usually have a coherent use of words and sentences clearly delimited. Spam comments on the other hand do not have a delimitation of sentences because they are incoherent and their content is inconsistent.

d. Number of punctuation marks in the comment

Spam comments tend to have an increased number of punctuation marks, especially exclamation marks and dots, in order to draw attention of the users who read the comment. The legitimate comments have punctuation marks only to delimit the sentences or to express strong feelings.

e. Word duplication

Spam comments tend to use repeated words in comparison with legitimate comments which have a continuous flow of context related text. The word duplication ratio is given by the ratio of the number of unique words in the comment and the total number of words in the comment. The formula is expressed in the following way:

$$Word\ Dupplication\ Ratio = \frac{Number\ of\ unique\ words\ in\ the\ comment}{Total\ number\ of\ words\ in\ the\ comment}$$

f. Stop words ratio

Legitimate comments tend to have a balanced number of stop words ratio in comparison with spam comments. The stop word ratio is calculated as the ration between the number of stop words from the comment and the total number of words in the comment. The formula can be expressed in the following way:

$$Stopword\ Ratio = \frac{Number\ of\ stopwords\ in\ the\ comment}{Total\ number\ of\ words\ in\ the\ comment}$$

g. Number of Non ASCII Characters in the comment

Spam comments tend to have a higher number of Non ASCII Characters in comparison with legitimate comments.

h. Number of Capital Letters from the comments

Capital Letters are used in spam comments to attract more attention than legitimate comments. Spam comments are very often written entirely using capital letters.

[1] www.youtube.com
[2] www.daily-telegraph.com

i. Number of new lines in the comment

Spam comments tend to have a high number of new lines in order to draw attention upon certain facts by creating a visual effect for the reader.

After the computation of the numeric features we perform an initial elimination of comments which have values of the numeric features higher than some threshold values described in [3] and in [9]. After this initial elimination of spam comments the system receives as input comments which do not contain discontinuous text flow and inadequate or vulgar language but only comments with fluent text. It is required to apply natural language processing techniques on the remaining comments in order to determine whether the comments contain relevant content with respect to a given topic.

2. Topic Extraction Module

The topic extraction module is designed to determine if there are common topics between a comment and a post (common topics being indicators of non-deceptive comments) and to find out if the content of the comment is related to the content of the related post. We considered two basic types of topics – bigrams and uni-grams which are extracted using a combination of shallow natural language processing techniques and statistical methods by combining ideas from [7]. To identify uni-gram topics we extract a collection of candidate nouns. To create a set of bigrams topics we extract all bigrams from both the comment and the post which conform to one of two basic part-of-speech co-location patterns: (1) an adjective followed by a noun (AN) such as *sunny day* and (2) a noun followed by a noun (NN) such as *trip advisor*.

To identify a possible set of candidate topics we perform the following steps:

In the first step we perform part of speech tagging using the Stanford Parser [8] in order to be able to detect the collocation patterns described above.

After this first step we search in the text for candidate uni-grams and bigrams which correspond to the collocation patterns mentioned above. After finding the occurrence of a uni-gram or a bigram they are stored with their number of occurrences.

Next we filter the candidate set of identified topics based on their frequency of occurrence in the given text keeping only those candidate topics which appear with a frequency threshold greater than 75%. We decided on this threshold value after we calculated the average mean of the frequency of occurrences of uni-grams and bigrams in legitimate comments. Through various experiments with different thresholds we observed that with this percentage the topics retrieved by the topic extraction module are consistent and numerous enough to reflect the topics of the text.

The final step is to compare the topics from the comment and from its associated post to check their topics similarity. In case we find at least 50% common topics between the comment and the post we consider them similar. The degree of similarity between a comment and a post has a value between 0 and 1 representing the percentage of common topics between them. The closer the value is to 1 the more common topics exist between a comment and a post. The flow of the topic extraction module is described in Figure 2.

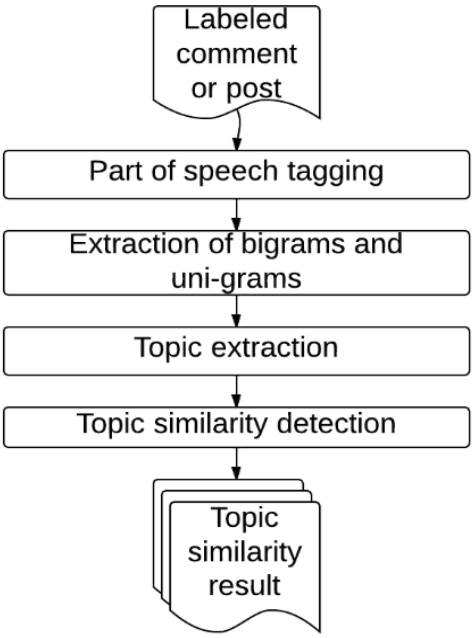

Figure 2: Flow of the topic extraction module

The formula for the topic similarity degree is given by the normalized value of the sum of the frequencies of occurrences of each topic from the comment in the post. The topic similarity similarity between a comment C_k and a post P_j is calculated in the following way:

$$Topic - similarity(C_k, P_j) = \frac{\sum_{i=1}^{n} w_{i,j}}{\max_s(\sum_{i=1}^{n} w_{i,j})}$$

The following notations are used:

C_k = comment processed
P_j = post processed
n = total number of words from comment C_k
w_{ij}=frequency of topic i from comment C_k in post P_j

3. Post-comment Similarity Module

The spammers generally provide spam comments which are not related to the context of the posting at all. In order to decide whether a comment is similar to a post we need to know which post the comment is related to. However, there exist comments which are generally considered spam comments to various types of posts even if they have fluent content. The proposed system identifies these types of comments which provide fluent content but are not relevant to a post because they discuss a different topic then the topic discussed in the post. Such a possibility is detected by analyzing the words from the comment with respect to the words in the post in order to check whether there exist similarities between them. These types of comments transmit a

general idea without referring to a specific topic from the post. An example of a comment spam which is not related to the context of the blog would be: *'Hello guys! I absolutely enjoyed watching your video. I considered watching it very helpful. Thank you for posting it!*'. However, this type of comment spam can be quite difficult for human analysts to detect it, because it looks quite similar to legitimate comments.

The first step the module performs is to preprocess the comment by applying the following operations: stop word removal and tokenizaton.

In the next step the comment is parsed at token level in order to retrieve a list of synonyms for each word from the comment. In order to identify a list of synonyms for a given word from the comment we implemented the synonym retrieval submodule that makes a connection to an online dictionary Thesaurus.com and makes a request to parse the source page and to extract all the synonyms for a specific word. The list of synonyms is retrieved and it is stored inside the post-comment similarity module.

For each word inside the comment we search if the same word appears in the associated post or if synonyms of the word appear in the associated post and we calculate the sum of the frequencies of occurrences of each word and its synonyms from the comment in the post. The formula is given by the normalized value of the sum of the frequencies of occurrences of each word and its synonyms from the comment in the post. The post-comment similarity between a comment C_k and a post P_j is calculated in the following way:

$$Post - comment\ similarity(C_k, P_j) = \frac{\sum_{i=1}^{n} w_{i,j} + \sum_{i=1}^{|H(s)|} w_{s,j}}{\max_{s}(\sum_{i=1}^{n} w_{i,j} + \sum_{i=1}^{|H(s)|} w_{s,j})}$$

The following notations are used:

C_k = comment processed
P_j = post processed
n = total number of words from comment C_k
|H(s)| = cardinality of set H
$H(s) = \{s | s = \text{synonym}(i)\}$
w_{ij}=frequency of word i from comment C_k in post P_j
synonym(x)=y, where y is the synonym of x
w_{sj} = frequency of word s from comment C_k in post P_j

The flow of the post-comment similarity module is described in Figure 3.

IV. DATA ACQUISITION

In our research we want to build a model out of labeled data in order to detect spam comments. The data corpus of Mishne et. al. [2] for the classification of comments as spam and not spam contains 1024 comments related to a post. These comments are retrieved from a small collection of 50 blog pages and they are manually classified, 67% from them being classified as spam.

The legitimate comments are usually of medium length and they provide questions related to the post or they provide content with relevant information to the post they are associated to. The spam comments usually contain text which is not related to the associated post at all and they contain a large number of links which are used to redirect the user to other web pages. They also contain advertising for some unknown products.

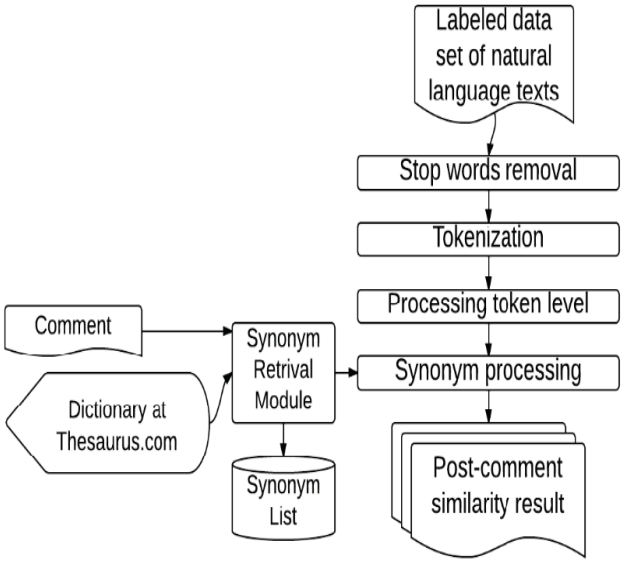

Figure 3: Flow of the post-comment similarity module

However, the dimension of the comments is quite unbalanced so that some comments are very long and some comments are very short (this situation occurs both for spam comments and for legitimate comments) and it is more difficult to analyze and compare the comments when they differ in the number of words they contain. Therefore a normalization process of the numeric features extracted from the comments is required in order to have outcomes which can be compared with respect to a common scale.

The experiments we performed are on comments similar with the comments from the data corpus and they are taken from YouTube and from the Daily Telegraph. The post from a YouTube page and the comments related to that post are very similar in their format with the comments and posts from the data corpus mentioned above. That is why the training can be done by using the data corpus from [2] and the testing can be performed on the data corpus obtained from YouTube and Daily Telegraph.

We manually annotated the set of extracted comments by taking into consideration the values of the numeric features with respect to the threshold values from [3] and [9] and also by extracting the number of common topics between the comment and the post.

V. EXPERIMENTS AND RESULTS

We performed experiments for every step of our implementation and then we compared the results obtained to observe the improvements from one stage to the other.

Stage 1: Our goal is to eliminate comments which contain discontinuous, incoherent text and contain offensive and vulgar language. These types of comments usually contain capital letters, non ASCII characters and digits to promote less known products. For the experiment we considered nine numeric features from [3] and [9] and they are: non ASCII characters, capital letters, new lines, digit count, sentence count, triviality words [6], word count, punctuation marks and number of links. The results are shown in the first row of Table 1. The results are satisfactory, but can be further improved by adding new features.

Stage 2: We observe that word duplication is also an important feature for the detection of spam comments. In the second stage of our implementation we add the word duplication ratio. Spam comments tend to use repeated words in comparison to legitimate comments which have a greater number of unique words. The word duplication ratio is given by the ratio of the unique words in the comment and the total number of words in the comment. In the second stage of the implementation we added the word duplication ratio. The results obtained are better in comparison to stage 1 of the implementation because the word duplication feature makes a distinction between spam and legitimate comments. We present the results obtained in Table 1.

Stage 3: In order to further improve the results in stage 2 we found the need of introducing another feature to our algorithm of spam detection namely the stop word ratio. We calculate for each comment the stop word ratio which is given by the number of stop words divided by the total number of words in the comment. Most spam comments are filled with keywords in the form of noun-phrases without the formation of a complete sentence which is expected to contain categories like verbs, prepositions, stop words, etc. From this inference we hypothesized that spam comments have an unbalanced number of stop words in comparison with legitimate comments. We provided our system with a list of stop words from [3]. The result improvement in stage 3 towards stage 2 is due to the introduction of the feature stop word ratio which is higher for legitimate comments than for spam comments.

Stage 4: In the last stage of our implementation we add two more features: post-comment similarity and topic similarity to eliminate comments which are not related to a specific context.
Legitimate comments are related to a topic which is given in its associated post. We consider that a feature indicating whether the comment contains topics which are referred to in the post would improve our results. That is why we introduce the feature of topic similarity which indicates whether the comment and its associated post contain common topics. In order to extract the topic similarity feature we use the topic extraction module which extracts the topics from the comment and from the post which occur with a frequency greater than a threshold of 75%. In case at least 50% of the extracted topics from the comment and from the post are similar the topic similarity feature will return a value between 0 and 1 which represents the percentage of common topics between the comment and the post.
Moreover, another feature which we consider of high importance is the percentage of similar words in the comment and in the post. If the comment and the post contain the same words or words with similar meaning it is very possible that they are context related. To detect if the comment and the post contain words having the same meaning we use the post-comment similarity module. The results obtained in stage 4 are very good which leads us to the conclusion that the topics of both the comment and the post are very important in order to decide to what type of comment we deal with.

The evaluation of the classification in the stages presented above is done by using three different types of classifiers such as: Decision Trees, Support Vector Machine and Naïve Bayes. The results of the precision and recall for each stage using different classifiers are described in Table 1.

TABLE I. RESULTS OBTAINED DURING DIFFERENT STAGES OF OUR IMPLEMENTATION

Input Features	No. of features	Precision (%)			Recall (%)		
		Decision Trees	SVM	Naïve Bayes	Decision Trees	SVM	Naïve Bayes
Stage 1	9	62.00	20.00	98.00	88.57	90.90	69.01
Stage 2	10	66.00	98.00	98.00	86.84	71.01	71.01
Stage 3	11	74.00	90.00	94.00	92.50	86.53	77.04
Stage 4	13	92.00	90.00	94.00	95.83	97.82	85.45

Moreover, we performed experiments on posts and comments taken from the Daily Telegraph news web site and YouTube social media site. The classification is performed using the Decision Trees classifier and the obtained results are presented in Table II. The precision and recall percentages for the comments extracted from YouTube are smaller in comparison with the values of the precision and recall on comments and posts from the Daily Telegraph because the comments are shorter and there are many words in the comments which are misspelled. The comments from the Daily Telegraph are more consistent and provide better spelling and this is why the results are better.

TABLE II. RESULTS OBTAINED FROM CLASSIFICATION OF COMMENTS FROM DAILY TELEGRAPH

Source	Precision (%)	Recall (%)
Daily Telegraph	95.83	83.66
YouTube	60.00	54.55

The results we obtained in our implementation of topic extraction module are shown in Table III. We chose a post from the interview domain in order to present the main topics which are extracted for different values of thresholds. We chose the value of 75% of the threshold because with this value we extract a sufficient number of main topics both from

the comment and the post. If the threshold has a lower value the main topics extracted by our topic extraction module will be too many and in this way we will have an increased number of comments and posts which will be topic related. For a post from the interview domain we obtained the results shown in Table III for topic extraction with threshold values greater than 50% and threshold values greater than 75%. We consider that a threshold with a value greater than 75% is enough in order to detect the main topics of the comments or of the posts.

TABLE III. RESULTS OBTAINED DURING DIFFERENT STAGES OF OUR IMPLEMENTATION OF THE TOPIC EXTRACTION MODULE

Threshold > 75%	Threshold > 50%
interview questions company organization	work situation past work interview questions company organization

In Table IV we show the results we obtained when calculating the average value of the post-comment similarity results for spam and legitimate comments. We observe that the average value for spam comments is close to 0 and the average value for legitimate comments is close to 1.

TABLE IV. POST-COMMENT SIMILARITY AVERAGE RESULTS FOR SPAM AND LEGITIMATE COMMENTS

	Spam comments	Legitimate comments
Post-comment similarity average	0.02	0.08

VI. CONCLUSION AND FUTURE WORK

We conducted an in-depth analysis on comments to shed some light on different features of spam comments. We were interested in building a system which detects spam comments with respect to a number of characteristics we defined. We wanted to analyze how the numeric features such as new lines, punctuation marks, links, white spaces, capital letters, vulgar words help eliminate incoherent and offensive comments and how the topic of the comment influences the detection of spam comments.

In our classification experiments, we demonstrated that our implementation of the spam detection system provides the best results by using the Decision Trees classifier. We observed that by adding the post-comment similarity and topic similarity the detection of spam comments is improved compared to the results obtained with the initial features..

In our future work, we want to introduce another module to our spam detection system. This module will be used in order to perform sentiment analysis on spam and legitimate comments. We want to make a comparison between the sentiment scores obtained for spam comments and the sentiment scores obtained for legitimate comments. The sentiment score can be further used as an input feature for the classifier which we built. The reason why we believe the classifier will provide better results by taking into consideration the sentiments transmitted by the writer in the comment is because in our research we observed that spam comments usually tend to express more negative feelings and legitimate comments on the other side tend express more positive feelings.

We think that the proposed technique has direct application on detection of spam comments.

REFERENCES

[1] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A Bayesian approach to filtering junk e-mail”. In AAAI-98 Workshop on Learning for Text Categorization, July 1998, Madison, Wisconsin, pp. 98-105.

[2] Gilad Mishne, David Carmel, and Ronny Lempel, “Blocking blog spam with language model disagreement”. In Proceedings of the First International Workshop on Adversarial Information Web (AIRWeb), Chiba, Japan, May 2005, pp. 1-6.

[3] A. Bhattari and D. Dasgupta, “A Self-supervised Approach to Comment Spam Detection based on Content Analysis”. In International Journal of Information Security and Privacy (IJISP), Volume 5, Issue 1, 2011, pp. 14-32.

[4] Davison B. D., “Recognizing Nepotistic Links on the Web”. In AAAI 2000 Workshop on Artificial Intelligence for Web Search, 2000, pp.23-28.

[5] Carreras, X. and Marquez, L., “Boosting trees for anti-spam email filtering”. In Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing, 2001, pp. 58-64.

[6] Stefan Siersdorfer and Sergiu Chelaru, “How useful are your comments?: analyzing and predicting youtube comments and comment ratings”. In Proceedings of the 19th international conference o World wide web, 2010, pp. 891-900.

[7] Ruihai Dong, Markus Schaal and Barry Smyth, “Topic extraction from online reviews for classification and recommandation”. In Proccedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013, pp. 1310-1317.

[8] The Stanford Natural Language Processing Group available at http://nlp.stanford.edu/software/corenlp.shtml.

[9] Serbanoiu, A., Rebedea T., “Relevance-Based Ranking of Video Comments on YouTube”. In CSCS ’13 Proceedings of the 2013 19th International Conference on Control Systems and Computer Science, 2013, Washington, USA, pp.225-231.

[10] I. Drost and T. Scheffer., “Thwarting the nigritude ultramarine: Learning to identify link spam”. In ECML’05 Proceedings of the 16th European conference on Machine Learning, 2005, Berlin, Germany, pp.96-107.