Multimodal Detection of Fake Social Media Use Through a Fusion of Classification and Pairwise Ranking Systems

Taruna Agrawal*, Rahul Gupta+, Shrikanth Narayanan*
*Signal Analysis and Interpretation Lab, University of Southern California, Los Angeles, CA, USA
+Amazon.com, USA

***Abstract*—The problem of detecting misinformation and fake content on social media is gaining importance with the increase in popularity of these social media platforms. Researchers have addressed this content analysis problem using machine learning tools with innovations in feature engineering as well as algorithm design. However, most of the machine learning approaches use a conventional classification setting, involving training a classifier on a set of features. In this work, we propose a fusion of a pairwise ranking approach and a classification system in detecting tweets with misinformation that include multimedia content. Pairwise ranking allows comparison between two objects and returns a preference score for the first object in the pair in comparison to the second object. We design a ranking system to determine the legitimacy score for a tweet with reference to another tweet from the same topic of discussion (as hashtagged on Twitter), thereby allowing a contextual comparison. Finally, we incorporate the ranking system outputs within the classification system. The proposed fusion obtains an Unweighted Average Recall (UAR) of 83.5% in classifying misinforming tweets against genuine tweets, a significant improvement over a classification only baseline system (UAR: 80.1%).**

Index Terms: Fake multimedia detection, Learning to rank

I. Introduction

Advances in social media platforms have led to widespread use and sharing of multimedia [1]. Platforms such as Twitter and Facebook allow dispersal of information and opinions which can be optionally aided with multimedia content. Consequently, there is a need for a mechanism to check the credibility of such content; a lack of which can sometimes lead to an abuse of social media platforms, and proliferation of fake information. Researchers have addressed this problem and have proposed several novel solutions using machine learning techniques, such as in detecting social spam campaigns [2] and misinformation [3]. Primarily, these studies have explored the application of classical classification methods to the problem of interest. In this work, we focus on the detection of multimedia misuse on Twitter (e.g. when a tweet accompanied with a multimedia object may be misleading or unrelated to the multimedia object). For this purpose, we propose an augmentation of classification systems with a learning to rank scheme, trained for establishing a preferential order amongst a set of objects. The supplementary pairwise ranking scheme is trained to prefer legitimate social media expressions over misinformation. Furthermore, the ranking scheme provides a legitimacy score for a Twitter expression in context of another expression from the same topic of discussion (as can be determined by hashtags on Twitter). This design contributes to a contextual assessment, helping to normalizing the differences in data distributions arising from different topics of discussion. Through these added advantages, we aim to advance the detection of fake social media content.

Several previous studies have conducted exploratory research on the abuse of social media platforms [4], [5]. Examples case studies include detection of social spammers [6], investigating the rise of social bots [7] and rumor propagation [8]. Machine learning tools have also been used to aid the detection of such content including the use of a bag of words approach [9], regression prediction models [10] as well as fuzzy logic techniques [11]. Boididou et al. [12] summarize a few challenges in computational verification in social media and discuss a few machine learning approaches in detecting fake content in social networks. Often, machine learning tools are also used as part of a larger system such as pruning images in collaborative photo collection [13] and data mining in social media [14]. The Verifying multimedia use task during MediaEval benchmarking initiative 2015 [15] led to further investigation in detecting fake content on Twitter with proposed approaches including the use of a two level classification system (a message level and a topic level classification) [16], agreement based retraining [17] and multimodal fusion [18]. A common theme amongst these approaches is the use of a conventional classification system. We explore a ranking scheme in our work which learns a preference order amongst a set of objects [19] and has been used in several applications such as information retrieval [20] and natural language processing [20]. Ranking methods have also been used in ranking social media content such as Twitter [21] and in recommender systems [22]. In particular, we train a pairwise ranking scheme [23], which given a pairwise preference between two objects learns a function to capture the preference orders. The novelty of our work is in the design of the ranking system followed by its integration with the classification methods for the purpose of detecting misuse of social media platforms.

Our goal in this work is to detect multimedia content propagated through tweets (over the twitter platform) that carries fake impressions or conveys incorrect information. In order to train the classification and pairwise ranking systems, we initially extract embedding based lexical features, features from the twitter platform and a few multimedia features. We train the classification and ranking systems based on these features. Training the pairwise ranking system also requires the creation of preference labels based on the original legitimacy labels. Finally, we integrate the ranking method within a traditional classification system for final evaluation. Our results indicate that system utilizing ranking scores within the classification system significantly outperforms a classification only system. Our ranking with classification system achieves an Unweighted Average Recall (UAR) of 83.5% in detecting “fake” vs “real” multimedia usage in Twitter over a traditional classification system UAR of 80.1%.

In the next section, we describe the database used for our experiments. Section III describes the features we use followed by a description of the methodology in Section IV. Finally, we discuss the results in Section V and present our conclusions in Section VI.

II. Database

We use the dataset provided as part of the *Verifying Multimedia Use* task during MediaEval benchmarking initiative 2015 [15]. The dataset consists of a set of tweets related to an event or a place and the tweets are accompanied with multimedia in form of images. Each of these pairs of tweets and images are then labeled as either carrying false impressions (fake) or faithfully conveying reality (real). A tweet is marked to be real if the associated image corresponds to the event that the tweet refers to. On the other hand a fake tweet contains images that do not correspond the event referred to in the tweet. Tweets that contain images with the purpose of humor may not be considered real or fake and were not included in the dataset. The training partition of the datasets consists of ∼5k real and ∼7k fake tweets while the testing partition consists of ∼1.2k real and ∼2.5k fake tweets. The event/place associated with the tweet is also available (e.g. "Syria", "Boston") and they are disjoint between the training and testing partitions. We refer to these event/place tags as *topics* and later use them for designing our ranking system. Note that these event/place tags can typically be obtained from hashtags associated with tweets. We refer the reader to [15] for more information regarding the dataset.

III. Features

We use three sets of features in our work: (i) Image based features, (ii) Twitter user based features and, (iii) Tweet based features. Below, we discuss each of these features and their representations for training the proposed machine learning algorithms.

A. Image based features

We use a set of forensic features extracted on images corresponding to the tweets as suggested in [15]. The motivation behind using forensic features in predicting the legitimacy of tweet is the fact that doctored images are often associated with fake tweets. Therefore, during prediction, the used of forensic features can help determine if an image is doctored. For an image corresponding to a tweet, we use the following set of features: (i) probability map of the aligned double JPEG compression [24], (ii) probability map of the non-aligned double JPEG compression [24], (iii) potential primary quantization steps for the first 6 DCT coefficients of the aligned double JPEG compression [24], (iv) potential primary quantization steps for the first 6 DCT coefficients of the non-aligned double JPEG compression [24], (v) Block artifact grid [25] and, (vi) Photo-Response Non-Uniformity [26]. These features are extracted as matrices for each image and we further extract statistics (mean, maximum, minimum, mode, standard deviation, quartiles: 5%, 25%, 50%, 75%, 95%) over the feature matrices to obtain a constant dimensionality feature vectors across all the training instances.

B. Twitter content and user based features

The twitter user based features consists of features corresponding to the user who made the tweet. These features include statistics such as number of user's followers, the number of times the user is included in a twitter list and, whether the user is verified. A full list of these features can be obtained from [12] (Table 1). These features help quantify the credibility of the user as well as are representative of patterns in the tweet content.

C. Tweet based features

Finally, we also extract features from the lexical composition of the tweet itself. The lexical composition of the tweet can contain indicators regarding the legitimacy of an expression as has been demonstrated in other experiments [3]. One could directly use n-gram based features [27] from the tweets or learn vector representations for the tweets [28], to be used later during machine learning model training/testing. The n-gram based features, despite being easy to extract, yield a sparse representation. Consequently, model training with them often need large amounts of data due to the high feature dimensionality. On the other hand vector representations are compact and are learnt through deep learning models (e.g. `doc2vec` [28]). Recently, such vector representations have been used in several applications such as sentiment classification [29] and designing question-answering systems [30]. One can train the vector representation models on out-of-domain datasets and obtain representations on the in domain dataset. We use the `doc2vec` framework in our experiments [28], that learns a paragraph matrix which is then used to obtain vector representations for a paragraph/sentence. We train the `doc2vec` model on the Sentiment140corpus [31] consisting of ∼1.5M tweets. Although the dataset is mismatched to the task at hand, it contains a large collection of tweets that can be used to learn representations for tweets in an unsupervised framework (not requiring fake/true labels). After training the `doc2vec` model, we obtain the vector representation for tweets in the training and testing datasets. We also conducted preliminary classification experiments comparing `doc2vec` representations to n-gram based features, yielding better results for the former.

After obtaining the features described above for every image, we concatenate them to obtain a feature vector for each tweet. We represent the feature vector for a tweet i as $\boldsymbol{x}_i$.

IV. Methodology

Based on the features mentioned above, we train a classification model, a ranking model and then finally combine the two. We describe these models in detail below.

A. Classification scheme

A classification scheme learns a function to map the feature vector $\boldsymbol{x}_i$ to the label space ($\in$\{fake, real\}). Also, during training a conventional classification system does not consider relationship that may exist between data samples (e.g. a set of tweet features drawn from same topic) and each feature sample $\boldsymbol{x}_i$ is treated to be independent of other samples. It is not straightforward to use the topic information (place/event tags) in a classification setting as topics during the test time may not exist in the training set or may even be unavailable for a test tweet. Our baseline method to infer the legitimacy of a tweet is a Support Vector Machine (SVM) classifier trained on a concatenation of the features described in the previous section. The classifier choice was tuned amongst a Deep Neural Network (DNN), Logistic regression and an SVM classifiers by using an inner cross-validation framework on the training set. The inner cross-validation framework was designed so as to have tweets from different topics in different splits (to mimic the real world scenario where a test tweet may belong to an unseen topic). We also Z-normalize [32] the training set features and use statistics on the training set to Z-normalize the test set. The parameters of the SVM classifier (Box constraint and Kernel) were also tuned using the inner cross-validation framework.

B. Ranking scheme

Apart from the classification scheme discussed above, we also design a pairwise ranking scheme to infer the legitimacy of a tweet. In a pairwise ranking scheme, a comparison is made between two instances based on their features and the preferred object is scored higher [23], [33]. For the purpose of our experiments, we train a ranking system to prefer the real tweets over the fake tweets. Given the feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from tweets i and j, we compute a *comparison vector* $[\boldsymbol{x}_i - \boldsymbol{x}_j]$ (the subtraction operation providing a notion of difference between the two tweets). We chose the pair of tweets i and j in this comparison from the same topics, in order to encourage comparison between tweets in the context of a topic. The

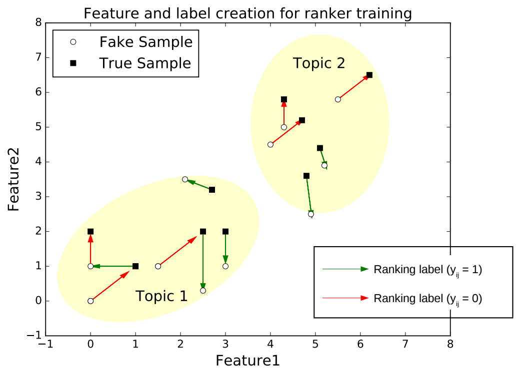

Fig. 1. Representation for data creation for ranker training. Green vectors correspond to ranker label $y_{ij} = 1$ and red vectors correspond to $y_{ij} = 0$.

pairwise preference label y_{ij} corresponding to $[\boldsymbol{x}_i - \boldsymbol{x}_j]$ is generated based on the following rule.

$$y_{ij} = \begin{cases} 1 & \text{if tweet } i \text{ is real and } j \text{ is fake} \\ 0 & \text{if tweet } i \text{ is fake and } j \text{ is real} \end{cases} \tag{1}$$

We do not generate comparison vectors and pairwise preference labels for tweets which are both fake or both true. Vectors generated by comparing tweet i against tweet j ($[\boldsymbol{x}_i - \boldsymbol{x}_j]$) is negative of comparing tweet j against i ($[\boldsymbol{x}_j - \boldsymbol{x}_i]$). In case we generate labels and comparison vectors for a pair of tweets with same labels, we end up with two opposite vectors with the same ranking label. Empirically, this negatively impacts the performance of the ranker. Overall, the design of ranking system provides following advantages over the classification system:

(i) Firstly, the system is trained on ranking a tweet contextually based on other tweets from the same topic. In the classification system, the classifier observes no context for a given tweet based on other tweets from the same topic. Therefore, the ranking system offers the advantage of evaluation in context of a topic.

(ii) Secondly, the ranker is trained on the translations from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ ($[\boldsymbol{x}_i - \boldsymbol{x}_j]$), instead of $\boldsymbol{x}_i$ or $\boldsymbol{x}_j$ themselves. This is different from the classification system which is trained directly on the tweet vectors. The relative position of a tweet based on other tweets in that topic during training can help normalize the difference in feature distributions arising from different topics.

(iii) Thirdly, the pairwise comparison between vectors leads to an increased amount of data. An larger dataset with same feature dimensionality can be used to train more complex machine learning models (e.g. DNNs). We depict the training data creation for ranking (using a synthetic example) in Figure 1 and summarize the ranker training next.

1) Ranker training: We first obtain the comparison vectors $[\boldsymbol{x}_i - \boldsymbol{x}_j]$ between tweets from each topic and the labels for each pair is obtained as shown in equation 1. The comparison vector $[\boldsymbol{x}_i - \boldsymbol{x}_j]$ is obtained on the Z-normalized feature vectors as specified in Section IV-A and we do not further normalize the pairwise differences themselves during ranker training and testing. We then train a DNN to predict the ranking labels given the pairwise difference features. The DNN is trained to optimize the cross entropy loss between targets and predictions. The number of hidden layers and nodes in each layer is tuned using inner cross-validation framework as we discussed in section IV-A. The chosen system is the one that yields that the lowest cross-entropy on the held out set.

2) Ranker testing: During testing, we assume that for a given test sample $\boldsymbol{x}_i^{\text{test}}$ we may not be aware of its topic and may not have tweets from the same topic to derive the comparison vector. We therefore consider two scenarios in which: (i) we do not have a second reference tweet (represented as $\boldsymbol{x}_j^{\text{test}}$) from the same topic to compute the comparison vector $[\boldsymbol{x}_i^{\text{test}} - \boldsymbol{x}_j^{\text{test}}]$ and, (ii) we have a reference tweet from the same topic to obtain the comparison vector. We discuss our approach to these scenarios below.

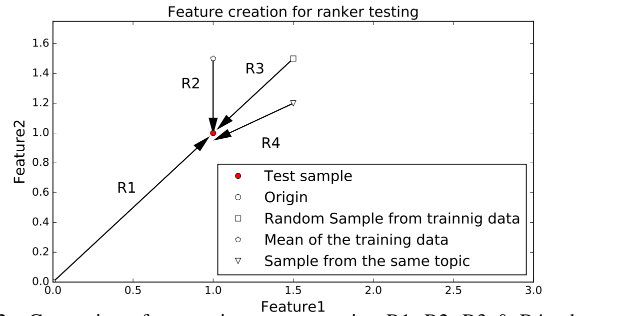

Fig. 2. Generation of comparison vectors using R1, R2, R3 & R4 schemes during testing.

Tweet from the same topic unavailable: In this scenario, we create a synthetic vector $\boldsymbol{x}_j^{\text{test}}$ to be a constant reference for all incoming test samples while computing the comparison vector $[\boldsymbol{x}_i^{\text{test}} - \boldsymbol{x}_j^{\text{test}}]$. Consequently, we obtain a ranking score for all test instances with reference to a constant vector. We test the ranking system with following options as the synthetic reference vector $\boldsymbol{x}_j^{\text{test}}$:

(R1) *Reference $\boldsymbol{x}_j^{test}$ is set to origin*: In this case the ranker score for each test instance is obtained with respect to a zero vector. This schemes obtains ranker scores based on the absolute position of the test instance in the feature space.

(R2) *Reference $\boldsymbol{x}_j^{test}$ is set to a random chosen training instance*: The ranker score for each test instance is obtained with respect to a randomly selected instance from the training data. The random selection is motivated from obtaining ranking score for the test sample with respect to a sample drawn from the data distribution. For this scheme, we test multiple randomly chosen vectors using the inner cross-validation framework as described in section IV-A and select the one that returns the least cross-entropy on the held out set.

(R3) *Reference $\boldsymbol{x}_j^{test}$ is set to the mean of the training data*: In this scheme, the ranker score for each test instance is obtained with respect to the mean of the data distribution, as estimated from the training set.

Tweet from the same topic available (R4): In this case, we assume that another tweet from the same topic as the test tweet is available. This case exactly matches the ranker training setting, as the comparison vectors $[\boldsymbol{x}_i - \boldsymbol{x}_j]$ are obtained from the tweet pairs i and j, drawn from the same topic. In terms of implementation, we randomly chose a tweet from a given topic in the test set as the reference tweet. We obtain one ranker score for every tweet in the test set, with reference to the randomly selected tweet from the corresponding topic. We note that during testing, the comparison vector for the selected reference tweet itself will be a zero vector.

Figure 2 summarizes test comparison vector creation for R1, R2, R3 and R4 (using a synthetic example). We emphasize the fact during testing, it is important to obtain a reference score with respect to a single tweet across all the tweets from a given topic. Having different references for every comparison only provides a comparison scores in the pairwise sense, not useful for a global assessment of tweet legitimacy. Obtaining ranking scores from a constant point of reference allows thresholding or learning a classifier to make decision regarding the legitimacy of the tweet, as discussed next.

TABLE I
RESULTS FOR CLASSIFYING "FAKE" VS "REAL" TWEETS USING VARIOUS CLASSIFICATION AND RANKING SCHEMES. *SCHEMES ASSUME THE PRESENCE OF A REFERENCE TWEET DURING TESTING. #RESULT IS SIGNIFICANTLY BETTER THAN THE BASELINE CLASSIFICATION SYSTEM USING MCNEMAR TEST [34].

System	UAR	Class accuracies	
		Fake	Real
Baseline Classification	80.1	76.1	84.1
Ranker (Reference R1)	80.7	77.4	84.0
Ranker (Reference R2)	79.9	78.8	81.0
Ranker (Reference R3)	79.4	76.2	82.6
Ranker (Reference R4)*	75.9	67.4	84.5
Classification with R1, R2, R3 ranker scores	82.6#	76.7	88.6
Classification with R1, R2, R3, R4 ranker scores*	83.5#	77.3	89.8

3) Evaluating the ranker results: Using one of the reference vectors selection schemes discussed above, we obtain a ranker score for each of the test instances. Later, we use these scores to infer the legitimacy of the tweet within the classification framework. Additionally, in order to test the ranker performances, we also convert the ranker scores to the final classes of interest (real/fake) by thresholding. We use a naive thresholding scheme for this purpose and assign all the test instances with a positive ranker score to be real (fake otherwise). The results using each of the four reference vectors schemes is discussed in the Section V.

C. Classification with ranking scores

In order to fuse the ranking schemes with the classification scheme, we append the ranker score as a feature to the set of features discussed in Section III. In training the classification system, we again consider the two cases mentioned before, regarding the availability of a reference tweet for ranking.

Tweet from the same topic unavailable: After training the ranker as discussed in Section IV-B1, we obtain the ranker scores on the training as well as the testing datasets, using the synthetic reference schemes R1, R2 and R3. The obtained ranker scores are then appended to the existing set of features. We then train a new SVM classifier on the training set with the expanded set of features and evaluation is performed on the testing set.

Tweet from the same topic available: In this case, apart from appending the scores obtained from R1, R2 and R3 schemes, we also append the scores obtained from the scheme R4 to train an SVM classifier. We randomly chose a reference tweet from each topic in the training and testing sets and obtain ranker scores using the trained ranker. The SVM classifier is trained on the expanded feature vector on the training set and evaluation is performed on the testing set.

We show the results for the proposed methods in the next section. These results including ranker schemes are separated based on the availability of a reference tweet.

V. RESULTS

We list the UARs along with the class-wise accuracies in Table I for the baseline classification system, ranker system and classification with ranker scores. From the results, we observe that the classification model aided with the ranker scores performs better than the baseline in both cases assuming availability/unavailability of a reference tweet during ranker testing. We further discuss these results in the next section.

A. Discussion

From the results, we observe that the baseline system performs significantly above chance. We noted that the tweet based features alone (without use of image and user features) provide an UAR of 72.3%. This reflects the fact that the unsupervised creation of the tweet based features from a model trained on out-of-domain data provides ample discriminatory power by itself. The `doc2vec` framework provides a low dimensional representation for lexical features, which can easily be used in conjunction with other features.

With respect to the ranking system, we observe that ranking with respect to the three synthetic reference points (R1, R2 and R3) yield approximately the same results. Based on our experiments, we recommend a careful creation of the synthetic reference point $\boldsymbol{x}_j^{\text{test}}$ based on tuning using a cross-validation framework. Our ranking experiments yield outcomes competitive to the baseline classification due to a meticulous reference point selection, which is later kept constant to obtain ranking scores on the test instances. The scheme R4 yields a slightly lower score than R1, R2 and R3 schemes. This may be due to fact that the naive thresholding scheme for inference based on ranker scores. R4 has different reference vector for each topic which is not ideal for a single threshold based schemes. Nevertheless, combining the R4 scores within the classification system outperforms all the other systems.

We also note that there are several sources of noise in obtaining the ranker scores, such as: (i) creation of the preference labels is discrete (0, 1 as in equation 1) instead of a preferred soft score also indicating the strength of preferring $\boldsymbol{x}_i$ over $\boldsymbol{x}_j$, and (ii) the reference points during testing are synthetically created (R1, R2, R3) or selected at random (R4). Despite these factors, the ranking system performs fairly close to the classification system (no significant difference between R1, R2 and R3 performance and performance of the classification system). We anticipate that the advantages we pointed to in section IV-B help overcome these shortcomings providing competitive results with the classification system. Finally, the improvements observed after fusion of classification and ranking system is encouraging for further exploration of the proposed approach.

VI. CONCLUSION

Detecting misinformation in social media is a problem of importance given the increasing prevalence of social media platforms. In this work, we propose a new framework for identifying tweets that do not correspond to the media item associated with them. Our framework uses a combination of ranking and classification methods, where the ranking framework provides the advantages of comparison with respect to a reference tweet from the same topic and normalizing the data distribution differences arising due to difference in topics of discussion. Our results indicate that incorporating the ranker scores within the classification systems significantly outperforms a stand alone classification system.

In the future, we aim to test more ranking and classification schemes in identifying fake social media content, particularly methods for obtaining a decision from ranked scores. One could also try other ranking schemes apart from the pairwise ranking scheme explored in this work. The proposed method could be extended to a wider set of problems in detecting fake content e.g. spams and social bots apart from the presented case study. Finally, we also aim to explore the presented method for other multimedia types (e.g. videos, vines) for detecting fake content.

REFERENCES

[1] M. Naaman, "Social multimedia: highlighting opportunities for search and mining of multimedia data in social media applications," *Multimedia Tools and Applications*, vol. 56, no. 1, pp. 9–34, 2012.

[2] X. Zhang, S. Zhu, and W. Liang, "Detecting spam and promoting campaigns in the twitter social network," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 1194–1199.

[3] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1589–1599.

[4] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer, "Detecting and tracking political abuse in social media." *ICWSM*, vol. 11, pp. 297–304, 2011.

[5] C. L. Hanson, B. Cannon, S. Burton, and C. Giraud-Carrier, "An exploration of social circles and prescription drug abuse through twitter," *Journal of medical Internet research*, vol. 15, no. 9, p. e189, 2013.

[6] X. Hu, J. Tang, Y. Zhang, and H. Liu, "Social spammer detection in microblogging." in *IJCAI*, vol. 13. Citeseer, 2013, pp. 2633–2639.

[7] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *arXiv preprint arXiv:1407.5225*, 2014.

[8] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 1103–1108.

[9] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+ machine learning," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 435–442.

[10] A. Gupta, H. Lamba, and P. Kumaraguru, "$1.00 per rt# bostonmarathon# prayforboston: Analyzing fake content on twitter," in *eCrime Researchers Summit (eCRS), 2013*. IEEE, 2013, pp. 1–12.

[11] R. Hassanzadeh, "Anomaly detection in online social networks: using data-mining techniques and fuzzy logic," Ph.D. dissertation, Queensland University of Technology, 2014.

[12] C. Boididou, S. Papadopoulos, Y. Kompatsiaris, S. Schifferes, and N. Newman, "Challenges of computational verification in social multimedia," in *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014, pp. 743–748.

[13] J. Yang, J. Luo, J. Yu, and T. S. Huang, "Photo stream alignment for collaborative photo collection and sharing in social media," in *Proceedings of the 3rd ACM SIGMM international workshop on Social media*. ACM, 2011, pp. 41–46.

[14] X. Jin, C. Lin, J. Luo, and J. Han, "A data mining-based spam detection system for social media networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 12, pp. 1458–1461, 2011.

[15] C. Boididou, K. Andreadou, S. Papadopoulos, D.-T. Dang-Nguyen, G. Boato, M. Riegler, and Y. Kompatsiaris, "Verifying multimedia use at mediaeval 2015," in *Working Notes Proceedings of the MediaEval 2015 Workshop*, 2015.

[16] Z. Jin, J. Cao, Y. Zhang, and Y. Zhang, "Mcg-ict at mediaeval 2015: Verifying multimedia use with a two-level classification model." in *MediaEval*, 2015.

[17] C. Boididou, S. E. Middleton, S. Papadopoulos, D. Nguyen, D. Tien, M. Riegler, G. Boato, A. Petlund, and Y. Kompatsiaris, "The vmu participation@ verifying multimedia use 2016," 2016.

[18] C. Maigrot, V. Claveau, E. Kijak, and R. Sicre, "Mediaeval 2016: A multimodal system for the verifying multimedia use task," in *MediaEval 2016:" Verfiying Multimedia Use" task*, 2016.

[19] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 129–136.

[20] T.-Y. Liu, "Applications of learning to rank," in *Learning to Rank for Information Retrieval*. Springer, 2011, pp. 181–191.

[21] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum, "An empirical study on learning to rank of tweets," in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 295–303.

[22] A. Karatzoglou, L. Baltrunas, and Y. Shi, "Learning to rank for recommender systems," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 493–494.

[23] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," *Artificial Intelligence*, vol. 172, no. 16, pp. 1897–1916, 2008.

[24] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of jpeg artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, 2012.

[25] M. Goljan, J. Fridrich, and M. Chen, "Defending against fingerprint-copy attack in sensor-based camera identification," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 227–236, 2011.

[26] W. Li, Y. Yuan, and N. Yu, "Passive detection of doctored jpeg image via block artifact grid extraction," *Signal Processing*, vol. 89, no. 9, pp. 1821–1829, 2009.

[27] W. B. Cavnar, J. M. Trenkle *et al.*, "N-gram-based text categorization," *Ann Arbor MI*, vol. 48113, no. 2, pp. 161–175, 1994.

[28] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents." in *ICML*, vol. 14, 2014, pp. 1188–1196.

[29] S. Lee, X. Jin, and W. Kim, "Sentiment classification for unlabeled dataset using doc2vec with jst," in *Proceedings of the 18th Annual International Conference on Electronic Commerce: e-Commerce in Smart connected World*. ACM, 2016, p. 28.

[30] Y. Belinkov, M. Mohtarami, S. Cyphers, and J. Glass, "Vectorslu: A continuous word vector approach to answer selection in community question answering systems," *SemEval-2015*, p. 282, 2015.

[31] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, p. 12, 2009.

[32] J. Mariéthoz and S. Bengio, "A unified framework for score normalization techniques applied to text-independent speaker verification," *IEEE signal processing letters*, vol. 12, no. 7, pp. 532–535, 2005.

[33] P. Donmez and J. G. Carbonell, "Optimizing estimated loss reduction for active sampling in rank learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 248–255.

[34] E. McCrum-Gardner, "Which is the correct statistical test to use?" *British Journal of Oral and Maxillofacial Surgery*, vol. 46, no. 1, pp. 38–41, 2008.