Active Learning Based News Veracity Detection with Feature Weighting and Deep-Shallow Fusion

Sreyasee Das Bhattacharjee, Ashit Talukder, Bala Venkatram Balantrapu
Department of Computer Science, University of North Carolina, Charlotte
Email:sdasbhat@uncc.edu, atalukder@gmail.com , bbalantr@uncc.edu

***Abstract*—The objective of a news veracity detection system is to identify various types of potentially misleading or false information, typically in a digital platform. A critical challenge in this scenario is that there are large volumes of data available online. However, obtaining samples with annotations (i.e. ground-truth labels) is difficult and a known limiting factor for many data analytic tasks including the current problem of news veracity detection. In this paper, we propose a human-machine collaborative learning system to evaluate the veracity of a news content, with a limited amount of annotated data samples. In a semi-supervised scenario, an initial classifier is learnt on a small, limited amount of the annotated data followed by an interactive approach to gradually update the model by shortlisting only relevant samples from the large pool of unlabeled data that are most likely to improve the classifier performance. Our prioritized active learning solution achieves faster convergence in terms of the classification performance, while requiring about 1-2 orders of magnitude fewer annotated samples compared to fully supervised solutions to attain a reasonably acceptable accuracy of nearly 80%. Unlike traditional deep learning architecture, the proposed active learning based deep model designed with a smaller number of more localized filters per layer can efficiently learn from small relevant sample batches that can effectively improve performance in the weakly-supervised learning environment and thus is more suitable for several practical applications. An effective dynamic domain adaptive feature weighting scheme can adjust the relative importance of feature dimensions iteratively. Insightful initial feedback gathered from two independent learning modules (a NLP shallow feature based classifier and a deep classifier), modeled to capture complementary information about data characteristics are finally fused together to achieve an impressive 25% average gain in the detection performance.**

***Index Terms*—Fake News Detection, Rumor, Active Learning, Deep Classification, Decision Fusion**

I. Introduction

The task of news veracity detection is to employ technology to identify deceptive (unreliable) digital news content in the web platform. With a plethora of information available from competing resources of varying quality, it is often hard for the users to gauge the trustworthiness of an online news content. In addition to traditional media outlets such as television, radio channels and printed form of daily newspaper which are still regarded as the top trusted news sources, the latest trend shows a critical bias towards Internet based resources. While, per the Gallup polls [1], only 32% of the Americans trust their mass media resources to report the news 'fully, accurately and fairly', the effect of this recent trend towards relying heavily on the new alternative digital information sources, like blogs and social media leave the readers more susceptible to incomplete, deceptive and misleading information. Another similar study in the UK [2] has also shown that in some cases, even some of the well-known newspapers are also found to be least reliable. Tabloid journalism, widely considered as 'yellow' or 'bad' journalism, essentially 'simplifies, personalizes and thrives on sensation and scandal' [3]. Currently, a significant portion of the online media economy is based on economic benefits from 'views' through advertising revenue and therefore clicks and page views translate directly to a dollar amount. The digital ad revenue in US was 50.7 billion in 2015 [4] and is expected to grow rapidly in the future. These trends suggest that an overlap of commercial and editorial interests may be potentially detrimental to the future of impartial journalism [5].

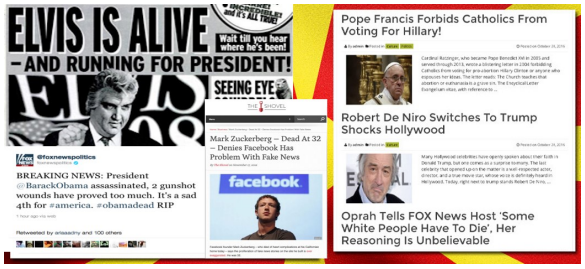

Fig. 1. Examples of Fake News in Social Media.

Thus an increase in the diversity of social news information sources, private journalism (versus journalism from media organizations) and low trust in the news institutions create a highly suitable environment for the 'rapid spread of information that is either intentionally or unintentionally misleading or provocative' [6]. Many people now consider social media as their primary source of news, especially in the case of breaking-news situations, where people crave for rapid updates on the developing events in real-time. More than 85% of all trending topics (trending topics are those, which are discussed more than other news) in Twitter are news in some form [7]. While the social networking sites like Twitter typically allow only a limited piece of information (due to character limits of twitter posts), it enables access to a large audience by network propagation, and thus remains more likely to be a fertile ground for the creation and spread of unsubstantiated and unverified information about events happening in the world. Some examples events of fake news are shown in Figure 1. Irresponsible reporting can have some

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

serious consequences, which can be damaging both socially and economically. For example, a 2008 hoax reporting the massive heart attack of then Apple CEO Steve Jobs, caused a sharp fall of the company's stock price by about 10% [8]. Another characteristic example is the 2011 London Riots. After the event unfolded, The Guardian shared an insightful graphic showing the patterns of the origination and spread of a number of fabricated information related to this event. The report also showed that analyzing the news content along with the proper evaluation of the reliability of their respective sources could help determining the factuality (or veracity) of these information. With timely action, it could also support the emergency services in making better-informed decisions.

In summary, content producers are increasingly more motivated towards speed and provocative display in contrast with restraint, reliability, and accuracy. Simultaneously, large numbers of content consumers lack the time and sufficient skills to critically evaluate and interpret the reliability and veracity of the information being relayed. Therefore, there is a critical need for an automated detection system that can reliably and quickly evaluate the veracity of a given news content such that they can be appropriately tagged prior to widespread dissemination. Beyond the mainstream media, rumor discovering websites like Snopes.com and PolitiFact.org evaluate the information credibility of a news content. However, such websites heavily rely on social media observers to tag and nominate potential fake news, which are then extensively evaluated by the human analysts appointed by the particular sites. Such manual fact-checking process is definitely expensive while causing longer delays in performing the identification task. Therefore, a strictly supervised learning scheme is not very effective in this problem scenario, as every event has some amount of more implicit unique characteristic features, which need to be captured for a more accurate identification performance. While news articles may involve multimedia content, in this paper, we restrict ourselves to the task of evaluating the veracity of only the textual news content, since text is usually the primary skeleton bearing the most critical part of the information about a specific event. In this paper, we address the problem of evaluating the veracity of a news content, which can either be a short twitter like message, or a long news text of paragraph(s). The primary contributions of this work are threefold: (1) The proposed human-machine collaborative learning system is suitable for fast identification of the potential emerging deceptive (or fake) news by leveraging the strengths of machines and humans. Having been modeled for the weakly-supervised problem scenario, the framework is capable of quickly and effectively adopting the contextual domain information using only a small amount of annotated samples and thereby found to be more appropriate for fake news detection in a real setting. Most importantly, this interactive framework allows for dynamic adaption of the model in real-world scenarios where news characteristics could change over time. Furthermore, the prioritized active learning based approach allows for our solution to be fine-tuned to specific domains such as political news, health, security and military, or specific regions. (2) Unlike traditional deep learning architectures that typically require a large number of annotated samples and an extensive training for lengthy time periods, the proposed deep learning module is designed with a smaller number of more localized filters per layer, which can learn using a smaller set of annotated samples and thereafter gradually improve itself with the updated annotated data pool in the active learning environment. (3) An effective dynamically adaptive feature weighting scheme can evaluate the relative importance of the feature dimensions (e.g. more important topics for a specific news genre) during the learning session. Finally, a set of complementary data information is captured in two separate learning modules modeled in parallel, which are later combined through a decision fusion scheme to exploit the individual strengths of each of these modules in an unified framework for improved detection performance. The rest of the paper is organized as follows: Section II briefly describes related works. The proposed method is explained in Section III. Section IV and V respectively present the experimental results and conclusion.

II. Related Work

The task of automatic evaluation for the veracity of a news content is a daunting task, involving collaborations between researchers from multiple disciplines [9; 10]. The key concepts of most of these works involve determining information quality and trustworthiness of a specific piece of information being published for public access. Other relevant metrics that are used to evaluate information veracity include Accuracy (Free-of-error), Reliability, Objectivity (Bias), Believability (Likelihood, Plausibility of arguments), Popularity, Competence and Provenance [11; 12].

With the recent surge in the social microblogging services, the current research focus has shifted to investigation of how false news (or rumor) are manifested and get disproportionate attention of the readers. Ghaoui [13] evaluates trustworthiness, credibility assessment and information verification in online communication to detect rumors. Seo [14] employs a graph based approach by investigating the monitoring nodes among all, which receive the given information. Qazvinian et al. [15] suggest label-dependent features in creating their User based USR and URL features. Nurse et al. [16] have proposed a framework that builds a policy based approach based on several trust and quality metrices. Leskovec et al. [17] use evolution of quotes to identify and analyze the spread of misinformation over time. Ratkiewicz et al. [18] detect misleading political memes in Twitter using tweet features like hashtags, links and mentions. Mendoza et al. [19] analyze Twitter data to understand the user behavior during the emergency situation by exploring the re-tweet network topology. Yang et al. [20] use several meta-data details like content information, client details, account, location and propagation along with the client-based features to evaluate the post on a popular Chinese microblog called Sina. A very comprehensive work is presented by Vosoughi [21], where various models are tested aiming for detecting and verifying rumors in Twitter. Given a

specific event, rumors are detected through clustering of its assertive arguments. Several semantic and syntactic features are used to model a Logistic Regression classifier. Regarding the verification of a rumor, the models utilize features considering the diffusion of information, multiple linguistics, user-related aspects and temporal propagation dynamics. Samadi et al [22] focus on evaluating only short, simple and single sentence-length factual claims by learning evidence classifiers. Popat et al.[23] retrieve diverse articles about the claim, and model the mutual interaction by stance detection. However, most of these methods rely on the availability of a large amount of annotated samples for an effective learning.

While evaluating the veracity of a news content is non-trivial, it becomes more difficult due to the presence of textual entailment (recognizing whether the meaning of one given statement can be inferred from another given statement), which results in accuracies lower than 70% on balanced lab data sets containing short social media posts [24]. Thus, most of these methods exploit information beyond the content of the posts, usually by analyzing the collective behavior of users typically respond to a targeted post. In fact, most of these features can be collected only when the focus of interest is tweet like messages, for which these contextual information like hashtags, diffusion information etc. are available. However, in cases of published news items as a result of 'yellow' journalism may not have all these information handy in general. Moreover, even if part of these information be available, they may be collected after the potentially damaging mis-information has been sufficiently circulated for a while. In fact these features become available, only when the specific news item has reached and been responded by many users. Such reactive approaches may not be appropriate for many real-life scenarios, where the goal is to stop the spread of such harmful false-information as early as possible to minimize its devastating effect.

Some recent works have attempted to address this specific scenario. Kwon et al. [25] and Friggeri et al. [26] tracked the judgments made by rumor debunking websites such as Snopes.com. Zhao et al. [27] present a technique to identify the trending rumors, that include disputed factual claims. However, the framework having heavily relied on contextual features like popularity scores etc., is not sufficiently generic to handle the identification task of typical fake-news. Rubin et al. [28] categorize fake news in three categories a) serious fabrications b) large-scale hoaxes; c) humorous fake contents; and attempt to identify the satirical cues to discriminate between real and fake news. However, these specialized cues can address only a small section of the whole spectrum of the problem. In this work, we propose an interactive learning based framework for fake news detection using only text content based features to address the problem scenario where reliable annotations may be available only for small sample sizes. As proved with experimental validations, the proposed method is sufficiently generic and can seamlessly handle both short Twitter based text content as well as long paragraph length news items within a single integrated framework while minimizing human effort in data inspection and annotation.

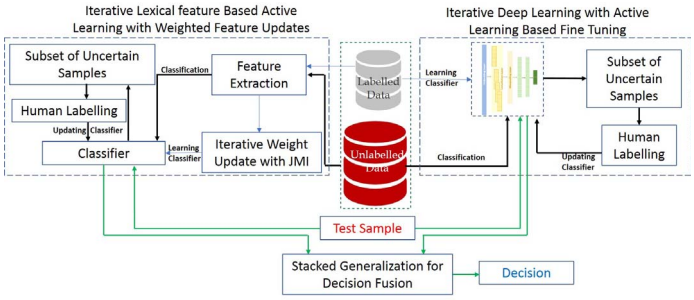

Fig. 2. Workflow for the proposed active Learning based framework for Detection of Fake News using an iterative approach.

Algorithm 1 Proposed Method

1: **Input:** $\mathcal{L}$: The labeled dataset, $\mathcal{U}$: the un-labeled dataset, the underlying Classification Model for Fusion Task
2: **Output:** The Learnt Fusion Classifier θ^{comb}
3: $\theta^{craft}, \mathcal{L}_1 \leftarrow$ NLP Shallow Feature Based Learning Model with Topic Re-Weighting by Algorithm 2
4: $\theta^{deep}, \mathcal{L}_2 \leftarrow$ Deep Feature Based Learning Model by Algorithm 3
5: $\mathcal{L} \leftarrow \mathcal{L}_1 \cup \mathcal{L}_2$
6: $C_{deep} = \theta^{deep}$.predict($\mathcal{L}$)
7: $C_{craft} = \theta^{craft}$.predict($\mathcal{L}$)
8: $C_{comb} \leftarrow C_{deep} \cup C_{craft}$
9: Train the fusion classifier θ^{comb} on C_{comb} with Label set $\mathcal{L}$

III. Proposed Method

Given a set of annotated data $\mathcal{D} = \{(\mathbf{x}^i, c^i)\}_i$, $\mathbf{x}^i$ represents a news text content with the label c^i, where the label c^i is known only for a small subset $\mathcal{L} \subset \mathcal{D}$ and for the rest $\mathcal{U} = \mathcal{D} \setminus \mathcal{L}$ the labels are typically unknown i.e. $\mathcal{U} = \{(\mathbf{x}, ?)\}$. The task is to learn an effective model for detecting or evaluating the veracity of a news content, which can either be a short twitter text or a long paragraph detailing about a specific (alleged) incident.

The annotated text collection in $\mathcal{L}$ is used to create a baseline classifier θ_0, which uses samples from $\mathcal{U}$ to iteratively fine-tune itself following an active learning strategy to obtain a final classifier θ with a rich level of domain knowledge incorporated within. The proposed human-machine collaborative system based on an active learning framework can efficiently handle a problem scenario with insufficient labeled data. However, unlike typical greedy algorithms which perform an iterative process, each time by selecting a document $(\mathbf{x}, ?) \in \mathcal{U}$, query a labeler for its label and augment the new annotated sample within the existing training collection $\mathcal{L}$, the proposed interactive learning strategy shows an effective alternative by using a *machine-driven data subset selection scheme for identifying a smaller set of relevant unannotated data* that require human analysis and annotation for bulk tagging and thus ensuring faster convergence.

The proposed framework has two parallel independent learning modules: (i) Shallow feature based architecture (θ_0^{craft}), in which a text is commonly defined as a mixture representation of some topics. The entire learning module with domain adaptive feature weight updating scheme is elaborated in section III-B. (ii) CNN based architecture (θ_0^{deep}), which uses pre-learnt word embedding that aims at mapping semantic meaning into a geometric space and that is achieved by associating a numeric vector to define every word in a dictionary, such that the distance between any two vectors would reflect the semantic similarity extent between two associated words. The geometric space formed by these vectors is called an embedding space. Section III-C describes the architecture in details. Each label $c^i \in \mathcal{Y}$ is a discrete variable and $|\mathcal{Y}| = l$. In our work, a Logistic Regression classifier is learnt to build the NLP shallow architecture.

Framework Overview: Algorithm 1 describes the entire workflow of the method. Starting with $\mathcal{L} \subset \mathcal{D}$ as the initial annotated collection, two independent learning modules θ_0^{deep} and θ_0^{craft} follow two parallel interactive learning frameworks and continue their respective learning sessions, until the desired stopping criteria are met or budget exhausted (for example, detection accuracy on the validation data collection reaches a certain threshold or the difference in the achieved detection accuracies between two consecutive check-points is typically very low). The corresponding final classifiers are denoted as θ^{deep} and θ^{craft}. Thus the two modules (shallow and deep) are learnt to evaluate the reliability of a text content by capturing its complimentary set of information (detailed content based and an overall aggregated representation), which are later combined for an improved classification performance. Following the proposed interactive learning strategy, both these modules can be efficiently updated with only a limited amount of annotated samples and thereby making it more effective for a larger range of practical problems. The proposed active learning based model updating framework, which is adopted by each of these two modules to get updated in an iterative fashion is illustrated in Figure 2. In the following subsections we will describe the design of each of these modules in details.

A. Prioritized Active Learning

An effective, classifier-agnostic active learning based approach is proposed to deal with the constraint of limited number of annotated samples that builds a machine-driven mechanism for sample down-selection from the un-annotated training data; therefore, human users/analysts view and annotate only a very small portion of relevant samples for manual annotations. We follow the following uncertainty sampling strategies to identify the ambiguous data samples:(1) *Scheme 1*: samples for which the underlying classifier is uncertain, i.e. the sample instances that are in a close proximity to the decision boundary of the classifier model [29], (ii) *Scheme 2*: the samples for which the mean differences in the confidence scores for the possible class labels are considerably low [30] (iii) *Scheme 1+ Scheme 2*: a combination of the two uncertainty metrics, which considers both classifier uncertainty and confidence while selecting the samples for human evaluation/annotation. This sample prioritization helps automatic identification of a small relevant set of samples, whose characteristics are relatively unique as compared to the annotated samples that the existing model has already been exposed to. While a low maximum confidence score on an unannotated sample implies the lack of the model's understanding on this type of data, a low difference in the class confidence scores highlights the fact that the model may not be sufficiently equipped to show discriminative characteristics in identifying these kinds of data patterns. Therefore, for both the metrics employed in our interactive sample selection strategy, it is important to emphasize the model-update phase specifically utilizing only such previously unseen data patterns. However, as the iteration continues, the classification capability of the classifier improves, which may result in the decrease of ambiguous samples prompting the process to terminate faster than expected and without having incorporated sufficient intelligence in the learnt model. In order to guarantee the reliability, we allow a lower threshold to select this first set of ambiguous samples, which is followed by a second level of elimination to shortlist only the most relevant ambiguous samples for which human intervention and review are required, and annotating these samples are ideally most useful in improving the classifier model. In contrast to tagging each and every element of $\mathcal{U}$, annotating only these machine-selected shortlisted samples makes the annotation task significantly less burdensome on humans, and results in a more exhaustive learning while retaining a faster convergence rate at the same time since only a very small portion of the more relevant unlabeled data needs to be manually examined and labeled by the human users. Furthermore, having been modeled in an interactive framework, the proposed framework shows the ability for dynamic adaptation for several specialized news domains' like political, health, defense etc. The uncertainty scores for an unannotated sample $\mathbf{x} \in \mathcal{U}$ is computed as follows:

$$\begin{aligned} U_1(\mathbf{x}|\theta) &= max(P(c=1|\mathbf{x}), P(c=0|\mathbf{x})) \\ U_2(\mathbf{x}|\theta) &= |P(c=1|\mathbf{x}) - P(c=0|\mathbf{x})| \end{aligned} \tag{1}$$

Given the underlying classifier θ, $\mathbf{x} \in \mathcal{U}$ is treated as uncertain sample if the maximum class confidence score is less, i.e. $U_1(\mathbf{x}|\theta) < \beta$ or the classifier exhibits a high degree of ambiguity about the class labels of $\mathbf{x}$, i.e. $U_2(\mathbf{x}|\theta) < \eta$. The subset of unannotated samples selected by the machine are sorted (prioritized) based on the corresponding confidence scores assigned by the existing classifier to further shortlist the most K uncertain samples for human inspection. As can be seen in Figure 4 & 5 and also discussed in Section IV-D that the Prioritized Active Learning usually shows a fast convergence rate to a model typically learnable in a strictly supervised scenario but essentially employs nearly 1-2 orders of magnitude less annotated data samples. Thus with the proposed framework the user needs to spend significantly lesser amount of effort in annotating the data samples, which makes the process more viable and effective in a practical scenario of dealing with a large database.

Algorithm 2 Proposed NLP Shallow Feature Based Active Learning with Feature Weighting

procedure FEATURE SELECTION
  Input: $\mathcal{L}$: The labeled dataset, N: Number of top relevant features desired
  Output: $\mathcal{S}$: The subset of N top relevant features
  Initialization: $\mathcal{F} \leftarrow$ The set of all n features,
$\mathcal{S} \leftarrow$ empty set
  (Computation of Mutual Information for each feature dimension with class label): $\forall j$ compute $I(\mathbf{F}_j, \mathbf{C})$
  $\mathcal{S} \leftarrow arg\ max\ I(\mathbf{F}_j, \mathbf{C})$
  loop: Repeat until $|\mathcal{S}| = N$
  $k_0 = arg\ max_{\mathbf{F}_k \in \mathcal{F} \setminus \mathcal{S}}[min_{\mathbf{F}_s \in \mathcal{S}}(I(\mathbf{F}_k, \mathbf{F}_s; \mathbf{C}))]$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{F}_{k_0}\}$
  goto *loop*
procedure ACTIVE LEARNING WITH ITERATIVE WEIGHT UPDATES
  Input: $\mathcal{L}$: The labeled dataset, $\mathcal{U}$: the un-labeled dataset, $\mathcal{S}$: the relevant feature set, B: budget, the underlying Classification Model, predefined parameters r and o
  Output: θ^{craft}: The Learnt Classifier θ, the set of feature weights $\{w_j\}_1^n$
  $\forall j \in \mathcal{S}, w_j \leftarrow r$ and $\forall j \in \mathcal{F} \setminus \mathcal{S}, w_j \leftarrow o$
  Train the initial classifier θ_0^{craft} on $\mathcal{L}$ with weighted features
  $\theta^{craft} \leftarrow \theta_0^{craft}$
  loop: Repeat until budget B exhausted
  for $\mathbf{x} \in \mathcal{U}$ check if $CheckUncertainty(\mathbf{x}|\theta^{craft}) == True$ using Eqn 1
  Request label for K most uncertain samples $\{\mathbf{x}_i\}_{i=1}^K$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathbf{x}_i\}_{i=1}^K$
  $\forall j$, $w_j \leftarrow w_j + \nabla w_j$, where ∇w_j is as in Eqn. 3
  Update θ^{craft} on $\mathcal{L}$
  goto *loop*

B. Shallow Feature Based Active Learning Module with Feature Weighting

Each text sample $\mathbf{x}^i$ is represented in terms of n-dimensional feature ($\mathbf{f}^i$). While the proposed NLP shallow feature based learning model is invariant to the choice of the underlying feature representation scheme, in this work we have used the topic-model distribution as the descriptor. Topic modeling which is an unsupervised learning algorithm, can automatically discover themes of a document collection by clustering the word into a set of topics. The topic model is typically handy in terms of capturing the synonyms, hypernyms and hyponyms of a given word such as the words "human" (hypernym) and "Alex" (hyponym) would be clustered in the same topic. An important advantage of using the topic distribution as the feature is its compact representation, by reducing the feature dimensions and mapping semantically similar terms into the feature dimension. Several techniques have been proposed for this purpose, such as Latent Semantic Analysis [31], Probabilistic Latent Semantic Analysis [32], Latent Dirichlet Allocation (LDA) [33] etc. In our experiments, we have used LDA for the task, which defines topic as a distribution over a fixed vocabulary, where each document can display them in different proportions. It is typically used as a keyword selection mechanism [34] by selecting the top words from the topic based on their entropy. $\mathcal{D}$ is clustered into n topics, resulting into a n dimensional feature representation $\mathbf{f}^i$. In other words, each text sample $\mathbf{x}^i \in \mathcal{D}$ is defined in terms of the vector $\mathbf{f}^i = \{f_j^i\}_{j=1}^n$.

1) Feature Weight Assignment with Mutual Information: Given the feature set $\mathcal{F} = \{\mathbf{F}_j\}_{j=1}^n$ where $\mathbf{F}_j = [f_j^1, f_j^2, \cdot\cdot, f_j^{|\mathcal{L}|}]^T$, the first step is to design an efficient feature weight initialization and selection process that can be effectively used as inputs to a machine learning module in the next stage. In order to address this, we identify a smaller subset $\mathcal{S} \subset \mathcal{F}$ of more relevant feature dimensions, which essentially aims to build a sparse model that emphasizes and relies on the more important feature attributes in the training data by assigning larger corresponding weights to them, compared to the other less-relevant features. The proposed Joint Mutual Information (JMI) [35] based iterative strategy follows a 'maximum of the minimum' approach to iteratively identify the subset $\mathcal{S}$ of the most relevant features in a greedy manner. At every iteration, based on the following optimization function, a feature $\mathbf{F}_{k_0}$ is selected by computing the Joint Mutual Information Maximization [36] $I(\mathbf{F}_k, \mathbf{F}_s; \mathbf{C})$ to add to the existing $\mathcal{S}$, where k_0 is defined as follows.

$$k_0 = arg\ max_{\mathbf{F}_k \in \mathcal{F} \setminus \mathcal{S}}[min_{\mathbf{F}_s \in \mathcal{S}}(I(\mathbf{F}_k, \mathbf{F}_s; \mathbf{C}))] \quad (2)$$

$\mathbf{C} = [c^1, c^2, \cdot\cdot c^{|\mathcal{L}|}]^T$ and c^i represents the class label for $\mathbf{x}^i \in \mathcal{L}$. This iterative greedy search continues to shortlist the top relevant feature subset $\mathcal{S}$ of size N within the feature space. In our experiments we typically choose $N = 0.4n$. The initial feature weight $\{w_j\}_{j=1}^n$ is assigned as $w_j = r, \forall \mathbf{F}_j \in \mathcal{S}$ and $w_j = o, \forall \mathbf{F}_j \in \mathcal{F} \setminus \mathcal{S}$, where r and o are some predefined constants with $r > o$. Procedure Feature Selection in the algorithm 2 describes the method.

2) Active Learning with Topic Re-weighting: In an iterative approach, a set of unannotated samples are chosen using the method described in Section III-A, on which the labels are requested from the annotators and the active learning module also updates the feature weights w_j of $\mathbf{F}_j$ by logistic regression with l_2 regularization for a two class (label can be either 0 or 1) problem as follows:

$$\begin{aligned}\nabla w_j = \alpha[& \sum_{\mathbf{x} \in \mathcal{U}: \mathbf{F}_j \in \mathcal{S}} r \times f_j^i \times (c - P(c = 1|\mathbf{x})) \\ & + \sum_{\mathbf{x} \in \mathcal{U}: \mathbf{F}_j \in \mathcal{F} \setminus \mathcal{S}} o \times f_j^i \times (c - P(c = 1|\mathbf{x}))] \\ & - w_j\end{aligned} \quad (3)$$

where α is the complexity parameter that controls the effect of weight updates. One can also opt to dynamically re-initialize $\mathcal{S}$ at certain user-defined checkpoints of the iterative process. However, for simplicity sake we have adopted a fixed definition for $\mathcal{S}$, obtained from the first step using $\mathcal{D}_0$. The

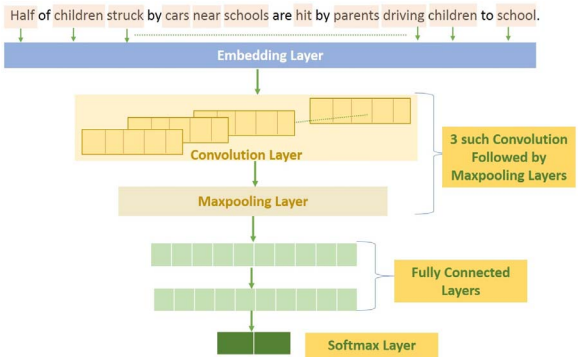

Fig. 3. The Deep Architecture Designed for the Task.

entire shallow feature based active learning module framework is illustrated in the procedure Active Learning with Iterative Weight Updates of algorithm 2.

C. Deep Architecture with Active Learning Based Fine-tuning Module

The simple yet effective deep learning architecture designed for this task is motivated by the CNN architecture proposed by Zhang & LeCun [37]. The input to the proposed deep model is a text content $\mathbf{x}^i$. The second layer is an embedding layer representing each input word in terms of a vector of a specific length. At the second layer input, each document is thus represented in terms of an embedding matrix of size $n \times E$, where E represents the embedding dimensions. The third, fifth and seventh layers are the temporal convolution layers. Each of these layers computes a 1-D convolution between input and output. We use N_m^{conv} convolution kernels with size $K_m \times 1$ for the m^{th} $(m = 3, 5, 7)$ convolution layer, which performs an 1D convolution over one position in the embedding vectors of K_m consecutive words. In other words, at layer h, a convolution operation involves a filter $\mathbf{w}^h \in R^{K_m E}$ words to produce a feature o_i at layer h generated by:

$$o_i^{(h)}(\mathbf{x}^i) = g(\mathbf{w}^h \cdot \hat{o}_i^{(h-1)} + b^{(h)}) \tag{4}$$

where $\hat{o}_i^{(0)}$ is a concatenation of K_m consecutive words. $b^{(h)} \in R$ is the bias at layer h and g is a rectified linear unit. The filter $\mathbf{w}^h$ is applied across all possible windows of characters in the text to produce a feature map. Each convolution layer is followed by a max-pooling layer (creating the fourth, sixth and eighth layer of the network), which performs max-pooling over the length $(n - K_m + 1)$ resulting in K_m $1 \times E$ vectors. The rectified linear units (ReLU) are used as the activation functions for all the layers. The output of the eighth layer is flattened and passed through three consecutive densely connected layers (ninth, tenth, and eleventh), where the eleventh one is a fully connected softmax layer whose output is the probability distribution over labels. In our model, we have used $N_3^{conv} = N_5^{conv} = 128$ and $N_7^{conv} = 64$, $K_3 = N_5^{conv} = 5$ and $K_7 = 35$. In this paper, we have used 100 dimensional GloVe (Global Vectors for Word Representation) embeddings [1]. However, users can also choose their alternate models, suitable for their application.

In order to address the issue of overfitting, dropput based regularization is employed, which randomly chooses a percentage κ of hidden units during the forward backpropagation step. This is used to cancel the contribution of some randomly chosen weight vectors. A scaled version of the learnt weight $(w_{sc} = \kappa \cdot w)$ without applying the dropout, is used at the inference step.

1) Deep Network Fine-tuning with Active Learning: At every iteration, a set of uncertain samples chosen using the method described in Section III-A is annotated by the user to obtain the augmented $\mathcal{L}$. In order to fine-tune the existing deep model, the standard back propagation algorithm is employed to update the last two fully connected layers' weight parameters in $\mathbf{W}$. More specifically, if F denotes the loss function defined in Eqn. 5 as follows:

$$F(\mathbf{W}) = -\frac{\sum_{l\in\{0,1\}} \sum_{i=1}^{|\mathcal{L}|} \mathbf{1}\{c_i = l\} \log\, p(c_i = l|\mathbf{x}_i; \mathbf{W})}{|\mathcal{L}|} \tag{5}$$

where $\mathbf{1}\{.\}$ is the indicator function, W represents the CNN weight parameters and $p(c_i = l|\mathbf{x}_i; \mathbf{W})$ computes the probabilistic score of the sample $\mathbf{x}_i$ for the class l

Given the updated $\mathcal{L}$, the task is formulated as solving the minimization problem defined as: $\min_{\mathbf{W}} F(\mathbf{W})$. As such, the partial derivative of F with respect to the network parameter $\mathbf{W}$ according to the Eqn 5 is:

$$\begin{aligned}
\frac{\partial F}{\partial \mathbf{W}} &= \partial \frac{-\frac{1}{|\mathcal{L}|} \sum_{l\in\{0,1\}} \sum_{i=1}^{|\mathcal{L}|} \mathbf{1}\{c_i = l\} \log\, p(c_i = l|\mathbf{x}_i; \mathbf{W})}{\partial \mathbf{W}} \\
&= -\frac{1}{|\mathcal{L}|} \sum_{l\in\{0,1\}} \sum_{i=1}^{|\mathcal{L}|} \mathbf{1}\{c_i = l\} \log\, \partial \frac{p(c_i = l|\mathbf{x}_i; \mathbf{W})}{\partial \mathbf{W}} \\
&= -\frac{1}{|\mathcal{L}|} \sum_{i=1}^{|\mathcal{L}|} (\mathbf{1}\{c_i = l\} - p(c_i = l|\mathbf{x}_i; \mathbf{W})) \frac{\partial a_l(\mathbf{x}_i; \mathbf{W})}{\partial \mathbf{W}}
\end{aligned} \tag{6}$$

where $\{a_l(\mathbf{x}_i; \mathbf{W})\}_{l\in\{0,1\}}$ represents the activation function at the second fully connected layer before feeding to the softmax layer and class specific probability score is defined as

$$p(c_i = l|\mathbf{x}_i; \mathbf{W}) = \frac{e^{a_l(\mathbf{x}_i; \mathbf{W})}}{\sum_{k\in\{0,1\}} e^{a_k(\mathbf{x}_i; \mathbf{W})}} \tag{7}$$

Important to note that the proposed deep architecture with active learning based weight updating strategy can seamlessly be extended for multi-class problem scenario.

D. Stacked Generalization for Decision Fusion

In order to combine these individual decisions, stacked Generalization or stacking [38] is used as an ensemble algorithm where a new linear classification model θ^{comb} is trained to combine the confidence scores from θ^{deep} and θ^{craft} to obtain a aggregated decision. The class confidence scores generated

[1] https://nlp.stanford.edu/projects/glove/

by both θ^{deep} and θ^{craft} for the updated annotated data samples in $\mathcal{L}_1 \cup \mathcal{L}_2$ (where $\mathcal{L}_1$ is the annotated data collection obtained by θ^{craft} and $\mathcal{L}_2$ is the annotated data collection obtained by θ^{deep}) are column-wise augmented to generate the data samples for learning θ^{comb}. In fact, a decision profile for $\mathbf{x} \in \mathcal{L}_1 \cup \mathcal{L}_2$ is defined in terms of its decision profile vector $D(\mathbf{x})$ as follows:

$$D(\mathbf{x}) = \begin{bmatrix} \theta^{craft}(\mathbf{x})[0] & \theta^{craft}(\mathbf{x})[1] & \theta^{deep}(\mathbf{x})[0] & \theta^{deep}(\mathbf{x})[1] \end{bmatrix}$$

where $\theta^{craft}(\mathbf{x})[i]$ (or $\theta^{deep}(\mathbf{x})[i]$) represents the class confidence score of the classifier θ^{craft} (or θ^{deep}) for the class i ($\forall i \in \{0, 1\}$). The elements in the entire collection $\mathcal{L}_1 \cup \mathcal{L}_2$ represented in terms of their decision profiles are used to learn a fusion classifier θ^{comb}. In our experiments, we have used a logistic regression model for this task.

Algorithm 3 Proposed Learning Algorithm for fine-tuning the Deep Model

1: **Input:** $\mathcal{L}$: The labeled dataset, $\mathcal{U}$: the un-labeled dataset, B: budget, the underlying Classification Model
2: **Output:**The Learnt Classifier θ^{deep}, network parameters $\mathbf{W}$, the augmented Annotated Dataset $\mathcal{L}$
3: Train the initial classifier θ_0^{deep} on $\mathcal{L}$
4: *loop*: Repeat until budget B exhausted
5: for $\mathbf{x} \in \mathcal{U}$ check if $CheckUncertainty(\mathbf{x}|\theta^{deep}) == True$ using Eqn 1
6: Request label for the K most uncertain samples $\{\mathbf{x}_i\}_{i=1}^{K}$
7: $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathbf{x}_i\}_{i=1}^{K}$
8: fine-tune θ^{deep} via updating $\mathbf{W}$ by minimizing the Eqn 5 on $\mathcal{L}$

IV. Experiments

A. Dataset

The proposed news content veracity detection framework is evaluated using three publicly available datasets to investigate its effectiveness in addressing the problem scenario in a real-life setting. The first one is the KDnugget's Fake News dataset [2] , which has almost $11,000$ articles published during 2015-2016 along with their titles and tagged as either real or fake. The entire corpus is built crawling 5,279 real news with New York Times and NPR APIs and 5279 fake news items to ensure an uniform distribution of the samples from both the classes. A randomly chosen collection of 2500 samples from each of the classes constitute $\mathcal{D}$ (i.e. $|\mathcal{D}| = 5000$). To start with, only 350 samples per category was used to build the annotated data pool $\mathcal{L}$ leaving 4300 sample in $\mathcal{U}$, which are used in the active learning iterations to gradually update the system. The left over 2279 samples from each of the two categories in the dataset, creates the test collection.

The second collection from Harvard Dataverse [39] contains tweet contents regarding 60 rumor and 51 non-rumor events. Number of tweets per event varies in the range $1,000-20,000$, resulting in approximately $0.95M$ rumor and $1.1M$ non-rumor tweets in our dataset. A randomly chosen $55K$ rumor and $55K$ non-rumor tweets constitute $\mathcal{D}$ among which $5K$ randomly drawn samples of each of the rumor and non-rumor categories are used to build the initial $\mathcal{L}$ resulting in $|\mathcal{L}| = 10K$. The rest $100,000$ elements of $\mathcal{D}$ are retained in $\mathcal{U}$.

Finally, third is the very recent Liar Dataset [40] for the Fake News detection. This includes $12.8K$ human labeled short statements reported primarily in the time-frame 2007-2016, from POLITIFACT.COM API, and each statement is evaluated by a POLITIFACT.COM editor for its truthfulness. Each data sample belongs to one of the six fine-grained categories based on the truthfulness ratings: *pants-fire*, *false*, *barely-true*, *half-true*, *mostly-true*, and *true*. Except for 1,208 *pants-fire* cases, the instances for all other labels range from $2,106$ to $2,632$. For our experiments, we have tagged news contents from both *mostly-true*, and *true* categories as real, while samples from the *pants-fire*, *false*, *barely-true* categories are identified as fake. Thereby, we have 4511 instances of fake news and 6003 instances of real news in the entire database collection. As the dataset contains samples from 6 different categories, with typically a lesser number of instances from each category, it is difficult segment the collection in three different parts ($\mathcal{L}$, $\mathcal{U}$ and test set) to learn a reliable model from this dataset. Therefore in our experiments, we used a small random sub-collection of the Harvard-Dataverse Twiiter collection (with $5K$ labeled samples) to train the initial classifiers (θ_0^{craft} and θ_0^{deep}). The Twitter collection consisting of short news texts has a similar structural characteristic as the Liar Dataset with short textual statements as its elements. A random subset of $100K$ twitter samples from the same Harvard-Dataverse Twiiter collection was used as the collection of the validation samples for updating the classifiers in the proposed interactive learning framework. The entire Liar Dataset collection was used for testing only In addition to proving the effectiveness of the proposed method, this set of experiments also shows the generalization capability of the learnt model which is often an important criteria in various application scenarios.

A successful evaluation on these recent datasets, consisting of news samples both in the traditional long text as well as the short 140 character length twitter text formats, shows sufficient evidences for the applicability of the proposed framework for several practical applications.

B. Implementation Details

Our active learning based approach is not dependent on the choice of the underlying classifier and we have performed the experiments using popular choices of classifiers like Logistic Regression and Ridge Classifier. As such using Ridge Classifier for this task also provides nearly equivalent performance and therefore in this paper, we have just reported the results using Logistic Regression for modeling the shallow feature based architecture θ^{craft}. In all our experiments, across all the three datasets, we have computed topic distribution details from the entire corpus using $n = 200$ topics through 2000 epochs to define $\mathbf{f}^i$ defined in section III-B. Scikit-learn implementation [41] of the classifiers with its default

[2]https://github.com/GeorgeMcIntire/fake_real_news_dataset

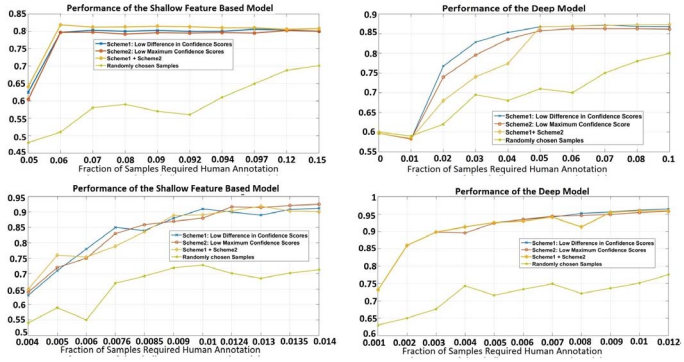

Fig. 4. Performance Evaluation on KDnugget's Fake News dataset (in the top row) and Harvard Dataverse Twitter Dataset (in the row below).

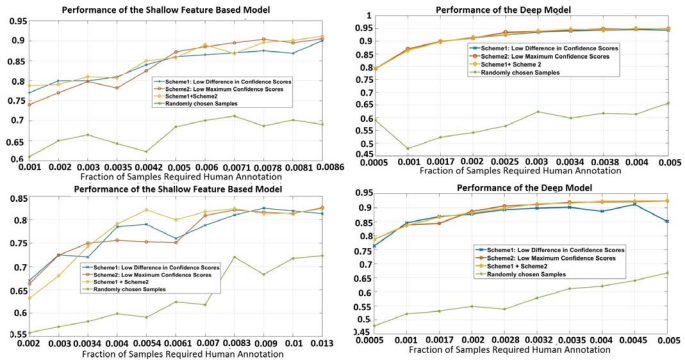

Fig. 5. Performance Evaluation on the larger collection of Liar News dataset, in which instances were relabeled for a two-class classification task so that samples from both *mostly-true*, and *true* categories are tagged as real, while samples from *pants-fire*, *false* and *barely-true* categories are tagged as fake.

parameter settings are used for our experiments. To identify the uncertain samples using the scores defined in Eqn 1, we have used $\beta = 0.7$ and $\eta = 0.5$. These samples are further shortlisted to identify the top K most uncertain samples, where K is computed to capture the top 40% of the entire list. The dropout rate used for the deep learning model was $\kappa = 0.5$.

C. Handling Data Imbalance

Starting with a small collection of uniformly distributed annotated collection, in order to ensure that the system learns from a reasonably balanced dataset, at any given iteration of the learning phase, if the minimum class population ratio is less than a certain threshold (in our experiment we have considered 70%), we use Synthetic Minority Over-sampling Technique (SMOTE) [42] to oversample the minority class instances from the existing $\mathcal{L}$ to obtain an uniform distribution for both the classes in the updated annotated training collection.

D. Results

In order to evaluate the performance of the proposed framework on all the above mentioned three datasets which are mostly balanced in nature, accuracy is used as the compact evaluation metric computed by relating FP (False Positives), FN (False Negatives), TP (True Positives) and TN (True Negatives) and defined as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Figure 4 shows the performance of the proposed active learning-based method in the KDnugget's Fake News dataset and Harvard Dataverse Twitter Rumor dataset, in which the gradual improvement of the performance for both the learning modules (shallow feature based architecture in (a) and deep architecture in (b)) is reported in 10 different checkpoints. Graphs in the figure clearly justifies the fact that even though the performance of the initial classifiers (θ_0^{craft} and θ_0^{deep}) are low, the proposed method shows significant improvement in the task by using only a small amount of human annotated uncertain samples, identified using a very intuitive formulation. Moreover, since the framework is designed to shortlist the a smaller subset of more ambiguous samples, the iterative learner updating process converges faster. In fact, the performance quickly stabilizes after a few initial iterations. For the KDnugget's fake news dataset, Starting from about 0.53 accuracy score obtained by θ_0^{craft} using 700 randomly selected annotated data samples, the learning process stabilizes at an early iteration stage with an improved accuracy score at around 0.82, using only around 7% human annotated samples, which consists of just about 350 un-annotated news content from the validation data collection. In a similar line, learning for the deep architecture also converges at 0.88 accuracy score by exploiting just about 5% human annotated samples, comprising of 250 un-annotated news content. The second dataset used in the paper is the Twitter collection, which is the biggest of all the three dataset used in this paper. Although the performance (around 63% accuracy) of the initial classifier θ_0^{craft} learnt using 10K randomly selected annotated samples, is not very impressive to start with, using only 0.9% of the entire entire pool of $100,000$ un-annotated samples, the improved NLP shallow feature based architecture gains an accuracy of 0.88 showing a near convergence state in its classifier updating process. On the other hand, the deep architecture shows more promise and utilizes only about 0.6% human annotated samples to obtain a much higher accuracy of 0.935 and stabilizes quickly thereafter. As defined in Eqn. 1, two different uncertain sampling strategies are adopted to identify the most ambiguous samples for human annotation task. The graphs in the figure also display the effect of choosing two different schemes in isolation as well as in combination. As observed, a combination offers a slightly better final classification performance compared to a scenario of adopting only one of them in general. However, scheme 2 which uses $U_1(.)$ (as defined in Eqn. 1) as the evaluating score for identifying the uncertain samples from the un-annotated collection, demonstrates an overall most stable performance in both these dataset. In fact, choosing the samples at random in every iteration of the learning seems to be least effective. Since both θ_0^{craft} and θ_0^{deep} are updated in parallel following two completely independent yet identical prioritized interactive

TABLE I
PERFORMANCE SUMMARY OF THE PROPOSED METHOD

Dataset		NLP Shallow Architecture				Deep Architecture			Stacked Generalization	
	Learning Strategy	Prior. Active Learning with Unif. Feature Weight	Initial Learning (θ_0^{craft})	Prior. Active Learning (θ^{craft})	Supervised Learning	Initial Learning (θ_0^{deep})	Prior. Active Learning (θ^{deep})	Supervised Learning	Prior. Active Learning (θ^{comb})	Supervised Learning
KDnugget's Fake News dataset	Accuracy	0.793	0.532	0.825	0.8512	0.605	0.883	0.9133	0.9064	0.927
	#Training Samples	1200	350	1150	5000	350	1000	5000	1572	5000
HD Twitter Collection	Accuracy	0.8458	0.634	0.8819	0.901	0.627	0.9350	0.952	0.9481	0.956
	#Training Samples	$10.9K$	$10K$	$10.9K$	$110K$	$10K$	$10.6K$	$110K$	$16.3K$	$110K$
Liar Dataset T/F Sub-Coll.	Accuracy	0.9017	0.573	0.9152	0.9205	0.641	0.9536	0.9617	0.9616	0.9698
	#Training Samples	$10.53K$	$5K$	$10.6K$	$100K$	$5K$	$10.25K$	$100K$	$17.89K$	$100K$
Liar Dataset Larger Coll.	Accuracy	0.8093	0.4951	0.8205	0.8414	0.5380	0.9257	0.9392	0.9294	0.9463
	#Training Samples	$11.1K$	$5K$	$10.9K$	$100K$	$5K$	$10.35K$	$100K$	$18.2K$	$100K$
	Avg. Accuracy	0.8375	0.5485	0.8607	0.8807	0.6026	0.9243	0.9414	0.9364	0.9493

learning framework, it does not affect the modeling time. An uniform performance gain achieved by θ^{craft} (and θ^{deep}) compared to θ_0^{craft} (and θ_0^{deep}) demonstrates the effectiveness of the proposed learning scheme in general.

The performance of the proposed method on the Liar dataset is shown in Figure 5. For this dataset, the whole set of experiments was repeated twice in an identical setting; first using the test samples from a smaller sub-collection consisting of samples from only *true* and *false* categories, next using the test samples as the entire dataset, which were relabeled appropriately for a two-class classification task so that only samples from *mostly-true*, and *true* categories are tagged as real, while samples from the *pants-fire*, *false* and *barely-true* categories are identified as fake. The top row of the Figure 5 illustrates the performance on the True/False sub-collection, while the performance gain achieved by the proposed method in the larger collection is demonstrated in the bottom row. The curves for the True/False sub-collection shows considerably better final performance compared to that obtained for the larger-subset. This is intuitively aligned to the fact that the labeling of this sub-collection is more reliable compared to the larger dataset labeling, in which the collection from 5 different categories were coarsely merged together to suitably form a collection of two class samples. In fact, both shallow feature based architecture and the Deep module show significant gain over the baseline initial classifiers (θ_0^{craft} and θ_0^{deep}) in both experimental settings. As seen in the figure, for the first set of experiments, the shallow feature based architecture needs about 0.6% validation samples, while for the deep architecture the requirement is still lesser (just about 0.25%) to reach convergence. On the other hand, in the second set of experiments, the shallow feature based learning module stabilizes after observing the annotation for about 0.9% of the validation samples, while for the deep architecture, this requirement is only about 0.35%.

This demonstrates an impressive performance of the proposed weakly-supervised active learning based modeling scheme across several baseline classifiers, that can ensure an effective gain in the detection accuracy by requiring less human intervention and utilizing a much lesser amount of annotated samples, compared to a strictly supervised scenario. On the other hand, both these independent learning modules capturing a complementary set of data details, are thus equipped with sufficient insight on the several aspects of the data characteristics, which are then exploited by an effective decision fusion technique to obtain a more comprehensive final decision. Both Figure 4 & 5 also show the performance choosing random samples for annotations during iterative learning.

Comparative Study: Two parallel learning modules performed using two independent learning architectures generate a larger set of annotated samples, which includes the original collection of annotated samples as well as the newly identified uncertain samples from the validation set that were tagged by the human annotator to update the classifiers. For the sake of simplicity we accept a little notation abuse and denote this augmented annotated data collection as $\mathcal{L}$. As seen in the Table I displaying the results for three datasets used for the experiments in the paper, the fusion has helped boost the performance by about 1.2% on average. By using Feature re-weighting on a selected set of feature dimensions, around 3% gain was achieved on average. More importantly, for all the three different learning architectures, the proposed prioritized active learning based method shows a nearly comparable performance, while utilizing much lesser amount of training samples, compared to total number of samples in $\mathcal{D}$ required for learning in a strictly supervised scenario. For example, in Liar Dataset, while in a strictly supervised scenario, around $100K$ annotated samples in the entire training collection was required by a linear classifier to achieve the accuracy 94.63%, the proposed method picks out just around $18.2K$ more relevant samples requiring manual annotation to achieve an competitive accuracy of 92.94%, which clearly proves its effectiveness to minimize the effort that would be otherwise

be needed to annotate a larger sample collection. Similarly in KDNugget's Fake News Dataset, compared to a strictly supervised scenario, in which a linear classifier needs about 5000 annotated samples to report an accuracy 92.7%, the proposed method has gained an accuracy of 90.64% with about 1572 selected annotated samples. In all experiments, the supervised learning was performed using Logistic Regression classifier. The entire set of 10 iterations using shallow learning model takes about 2-4 minutes on average, while finetuning of the deep model takes about 6-8 minutes, depending on the size of the initial $\mathcal{L}$.

V. Conclusion

This paper addresses the problem of identifying the veracity of a news content in a weakly supervised scenario. In our active learning based method an initial classifier model is learnt on small set of annotated samples, and is subsequently iteratively updated by utilizing a small number of most relevant data samples most likely to improve the classifier performance. We provide a framework that could accommodate dynamic changes in news characteristic and facilitates dynamic adaptation of the model for several specialized news domains like political, health, defense etc. Unlike the traditional deep learning models which require a huge amount of labeled data, the proposed simple yet efficient deep architecture is specifically designed to learn in the weakly-supervised scenario and is able to improve its generalization ability through interactive human inspection and annotation for small relevant samples that are likely to provide the most insight on the data characteristics. On the other hand, the proposed domain adaptive feature weighting scheme to evaluate relative feature dimension importance enables the shallow module learn a set of complimentary information, which are later fused for an impressive performance gain over the baselines. The proposed method is broadly applicable to a variety of practical classification and regression problems where large amounts of annotated data may not be available, including cybersecurity monitoring, defense and security, finance, and other applications.

References

[1] "Americans' trust in mass media sinks to new low," http://www.gallup.com/poll/195542/americans-trust-mass-media-sinks-new-low.aspx, September 2016.

[2] N. R. Reilly, R., "Power, principles and the press," http://www.theopenroad.com/wp-content/uploads/2012/09/Power-princi-ples-and-the-press-Open-Road-and-Populus1.pdf, 2012.

[3] H. rnebring and A. M. Jnsson, "Tabloid journalism and the public sphere: a historical perspective on tabloid journalism," *Journalism Studies*, vol. 5, no. 3, pp. 283–295, 2004.

[4] A. M. . D. Page, "State of the news media," http://www.journalismorg/2015/04/29/state-of-the-news-media-2015, 2015.

[5] N. Couldry and J. Turow, "Advertising, big data and the clearance of the public realm: Marketers' new approaches to the content subsidy," *International Journal of Communication*, vol. 8, pp. 1710–1726, 2014.

[6] L. Howell, "Global risks 2013. retrieved from cologny/geneva switzerland," http://www3.wefo-rum.org/docs/WEF_GlobalRisks_Report_2013.pdf, 2013.

[7] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *Proceedings of the international conference on World wide web*, 2010.

[8] G. Sandoval, "Who's to blame for spreading phony jobs story?" http://www.cnet.com/news/whos-to-blame-for-spreading-phony-jobs-story, 2008.

[9] C. Lioma, B. Larsen, W. Lu, and Y. Huang, "A study of factuality, objectivity and relevance: Three desiderata in large-scale information retrieval?" *CoRR*, vol. abs/1610.01327, 2016.

[10] K. P. K. Kumar and G. Geethakumari, "Detecting misinformation in online social networks using cognitive psychology," *Human-centric Computing and Information Sciences*, vol. 4, no. 1, 2014.

[11] J.-E. Mai, "The quality and qualities of information," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 4, 2013.

[12] R. Lukyanenko and J. Parsons, "Information quality research challenge: Adapting information quality principles to user-generated content," *J. Data and Information Quality*, vol. 6, no. 1, pp. 3:1–3:3, Mar. 2015.

[13] C. Ghaoui, *Encyclopedia of Human Computer Interaction*. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, 2005.

[14] E. Seo, P. Mohapatra, and T. Abdelzaher, "Identifying rumors and their sources in social networks," 2012.

[15] *Rumor Has It: Identifying Misinformation in Microblogs*, 2011.

[16] J. R. Nurse, I. Agrafiotis, M. Goldsmith, S. Creese, and K. Lamberts, "Two sides of the coin: measuring and communicating the trustworthiness of online information," *Journal of Trust Management*, vol. 1, no. 1, p. 5, 2014.

[17] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[18] J. Ratkiewicz, M. Conover, M. R. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer, "Detecting and tracking the spread of astroturf memes in microblog streams," *CoRR*, vol. abs/1011.3768, 2010.

[19] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the International Conference on World Wide Web*, 2011.

[20] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, 2012.

[21] S. Vosoughi, "Automatic detection and verification of rumors on twitter," http://hdl.handle.net/1721.1/98553, 2015.

[22] M. Samadi, P. Talukdar, M. Veloso, and M. Blum, "Claimeval: Integrated and flexible framework for claim evaluation using credibility of sources," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[23] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the web and social media," in *Proceedings of the International Conference on World Wide Web Companion*, 2017.

[24] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," in *Proceedings of the International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, 2006.

[25] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *Proceeding of the IEEE International Conference on Data Mining*, 2013.

[26] A. Friggeri, L. Adamic, D. Eckles, and J. Cheng, "Rumor cascades," 2014.

[27] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in *Proceedings of the International Conference on World Wide Web*, 2015, pp. 1395–1405.

[28] V. L. Rubin, "Deception detection and rumor debunking for social media," 2017.

[29] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 3–12.

[30] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Proceedings of the International Conference on Advances in Intelligent Data Analysis*, 2001.

[31] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[32] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceeding of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[33] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[34] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, pp. 1205–1224, 2004.

[35] H. H. Yang and J. Moody, "Feature selection based on joint mutual information," in *In Proceedings of International ICSC Symposium on Advances in Intelligent Data Analysis*, 1999, pp. 22–25.

[36] "Feature selection using joint mutual information maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520 – 8532, 2015.

[37] X. Zhang and Y. LeCun, "Text understanding from scratch," *CoRR*, vol. abs/1502.01710, 2015.

[38] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[39] S. Kwon, M. Cha, and K. Jung, "Rumor detection over varying time windows." 2017. [Online]. Available: http://dx.doi.org/10.7910/DVN/BFGAVZ

[40] W. Y. Wang, ""liar, liar pants on fire": A new benchmark dataset for fake news detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. Vancouver, BC, Canada: ACL, July 2017.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[42] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002.