The Good, the Bad and the Bait: Detecting and Characterizing Clickbait on YouTube

Savvas Zannettou, Sotirios Chatzis, Kostantinos Papadamou, Michael Sirivianos
Department of Electrical Engineering, Computer Engineering and Informatics
Cyprus University of Technology
Limassol, Cyprus
sa.zannettou@edu.cut.ac.cy, {sotirios.chatzis,michael.sirivianos}@cut.ac.cy, ck.papadamou@edu.cut.ac.cy

***Abstract*—The use of deceptive techniques in user-generated video portals is ubiquitous. Unscrupulous uploaders deliberately mislabel video descriptors aiming at increasing their views and subsequently their ad revenue. This problem, usually referred to as "clickbait," may severely undermine user experience. In this work, we study the clickbait problem on YouTube by collecting metadata for 206k videos. To address it, we devise a deep learning model based on variational autoencoders that supports the diverse modalities of data that videos include. The proposed model relies on a limited amount of manually labeled data to classify a large corpus of unlabeled data. Our evaluation indicates that the proposed model offers improved performance when compared to other conventional models. Our analysis of the collected data indicates that YouTube recommendation engine does not take into account clickbait. Thus, it is susceptible to recommending misleading videos to users.**

I. Introduction

Recently, YouTube surpassed cable TV in terms of popularity within teenagers [28]. This is because YouTube offers a vast amount of videos, which are always available on demand. However, because videos are generated by the users of the platform, known as *YouTubers*, a plethora of them are of dubious quality. The ultimate goal of YouTubers is to increase their ad revenue by ensuring that their content will get viewed by millions of users. Several YouTubers deliberately employ techniques that aim to deceive viewers into clicking their videos. These techniques include: (i) use of eye-catching thumbnails, such as depictions of abnormal stuff or attractive adults, which are often irrelevant to video content; (ii) use of headlines that aim to intrigue the viewers; and (iii) encapsulate false information to either the headline, the thumbnail or the video content. We refer to videos that employ such techniques as *clickbaits*. The continuous exposure of users to clickbaits cause frustration and degraded user experience (see Fig. 1) .

The clickbait problem is essentially a peculiar form of the well-known spam problem [38], [37], [36], [17], [20]. In spam, malicious users try to deceive users by sending them misleading messages mainly to advertise websites or perform attacks (e.g., phishing) by redirecting users to malicious websites. Nowadays, the spam problem is not as prevalent as a few years ago due to the deployment of systems that diminish it. Furthermore, users have an increased awareness of typical spam content (e.g., emails, etc.) and they can effortlessly discern it. However, this is not the case for clickbait, which usually contains hidden false or ambiguous information that users or systems might not be able to perceive.

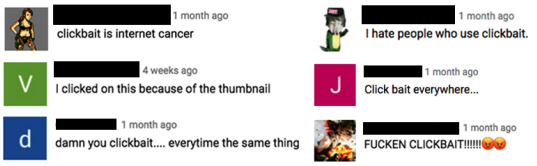


Fig. 1: Comments that were found in clickbait videos. The users' frustration is apparent (we omit users' names for ethical reasons).

Recently, the aggravation of the fake news problem has induced broader public attention to the clickbait problem. For instance, Facebook aims at removing clickbaits from its newsfeed [27], [14]. In this work, we focus on YouTube for various reasons: i) anecdotal evidence suggests that the problem exists in YouTube [8] and ii) to the best of our knowledge, YouTube relies on users to flag suspicious videos and then manually review them. To this extent, this approach is deemed to be inefficient. Hence, the need for an automated approach that minimizes human intervention is indisputable.

To attain this goal, we leverage some recent advances in the field of Deep Learning [22] by devising a novel formulation of variational autoencoders (VAEs) [21], [23] that fuses different modalities of a YouTube video, and infers latent correlations between them.

The proposed model infers a latent variable vector for each video that encodes a high-level representation of the content and the correlations between the various modalities. The significance of learning to compute such concise representations is that: (i) this learning procedure can be robustly performed by leveraging large unlabeled data corpora; and (ii) the obtained representations can be subsequently utilized to drive the classification process, with very limited requirements in labeled data.

To this end, we formulate the encoder part of the devised VAE model as a 2-component finite mixture model [10]. That is, we consider a set of alternative encoder models that may generate the data pertaining to each video. The decision of which specific encoder (corresponding to one possible class) generates each is obtained via a trainable probabilistic gating network [29]; this constitutes an integral part of the developed

autoencoder. The whole model is trained in an end-to-end fashion, using the available training data, both the unlabeled and the few labeled ones. The latter are specifically useful for appropriately fitting the postulated gating network that infers the posterior distribution of mixture component (and corresponding class) allocation.

Contributions. We propose a deep generative model that allows for combining data from as *diverse modalities* as video headline text, thumbnail image and tags text, as well as various numerical statistics, including statistics from comments. Most importantly, the proposed model allows for successfully addressing the problem of learning from limited labeled samples and numerous unlabeled ones (semi-supervised learning). This is achieved by postulating a deep variational autoencoder that *employs a finite mixture model as its encoder*. In this context, mixture component assignment is regulated via an appropriate gating network; this also constitutes the eventually obtained classification mechanism of our deep learning system. We provide a large scale analysis on YouTube; we show that, with respect to the collected data, its recommendation engine does not consider how misleading a recommended video is. Hence, it is susceptible to recommending clickbait videos to its users.

II. Methodology

By leveraging YouTube's Data API, between August and November of 2016, we collect metadata of videos published between 2005 and 2016. Specifically, we collected the following data descriptors for 206k videos: (i) basic details like headline, tags, etc.; (ii) thumbnail; (iii) comments from users; (iv) statistics (e.g., views, likes, etc.); and (v) related videos based on YouTube's recommendation system. We started our retrieval from a popular (108M views) clickbait video [1] and iteratively collected all the related videos as were recommended by YouTube. Note that this approach enables us to study interesting aspects of the problem, by constructing a graph that captures the relations (recommendations) between videos.

To get labeled data, we opted for two different approaches. First, we manually reviewed a small subset of the collected data by inspecting the headline, the thumbnail, comments from users, and video content. Specifically, we watched the whole video and compared it to the thumbnail and headline. A video is considered clickbait only if the thumbnail and headline deviate substantially from its content. However, this task is both cumbersome and time consuming; thus, we elected to retrieve more data that are labeled. To this end, we compiled a list of channels (available at [2]) that habitually employ clickbait techniques and channels that do not. To obtain the list of channels, we used a pragmatic approach; we found channels that are outed by other users as clickbait channels. For each channel, we retrieved up-to 500 videos, hence creating a larger labeled dataset. The overall labeled dataset consists of (i) 1,071 clickbaits and 955 non-clickbaits obtained from the manual review process and (ii) 8,999 clickbaits and 8,470 non-clickbaits obtained from the distinguished list of channels. The importance of this dataset is two-fold, as it allow us to study the problem and is instrumental for training our deep learning model.

A. Manually Reviewed Ground Truth Dataset Analysis

In order to better grasp the problem, we perform a comparative analysis of the manually reviewed ground truth.

Category. Table 1 reports the categories we find on the videos. In total, we find 15 categories but we only show the top five in terms of count for brevity. We observe that most clickbaits exist in the Entertainment and Comedy categories, whereas non-clickbaits are prevalent in the Sports category. This indicates that, within this dataset, YouTubers employ clickbait techniques on videos for entertainment.

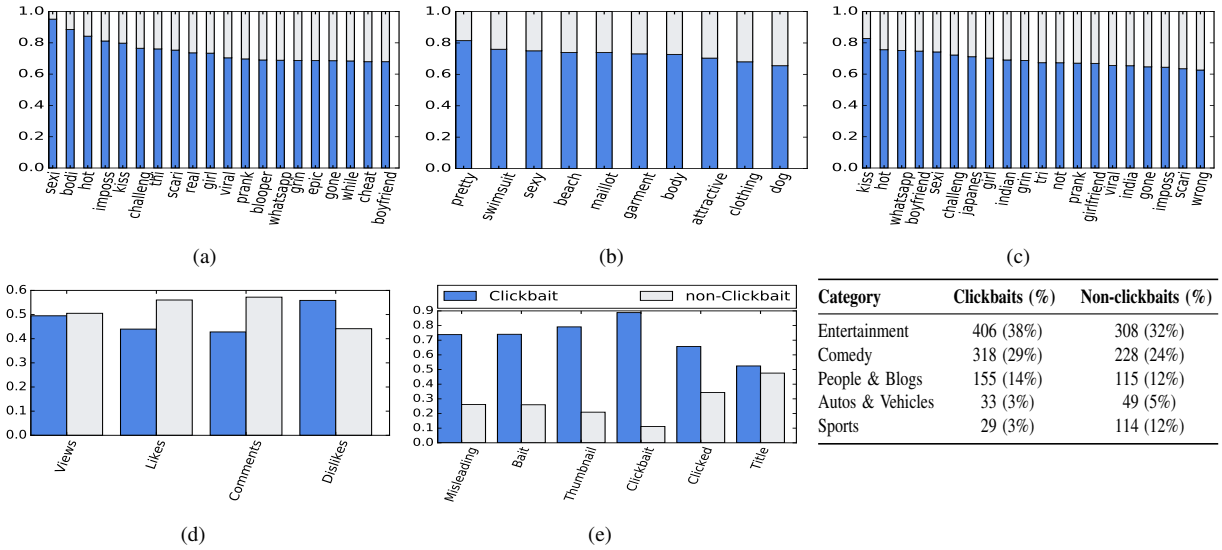
Headline. YouTubers normally employ deceptive techniques on the headline like the use of exaggerating phrases. To verify that this applies to our ground truth dataset, we perform stemming to the words that are found in clickbait and non-clickbait headlines. Fig. 2 (a) depicts the ratio of the top 20 stems that are found in our ground truth clickbait videos (i.e., 95% of the videos that contain the stem "sexi" are clickbait). In essence, we observe that magnetizing stems like "sexi" and "hot" are frequently used in clickbait videos, whereas their use in non-clickbaits is low. The same applies to words used for exaggeration, like "viral" and "epic".

Thumbnail. To study the thumbnails, we make use of Imagga [19], which offers descriptive tags for an image. We perform tagging of all the thumbnails in our ground truth dataset. Fig. 2(b) demonstrates the ratio of the top 10 Imagga tags that are found in the manually reviewed ground truth. We observe that clickbait videos typically use sexually-appealing thumbnails in their videos in order to attract viewers. For instance, 81% of the videos' thumbnail of which contains the "pretty" tag are clickbaits.

Tags. Tags are words that are defined by YouTubers before publishing and can dictate whether a video will emerge on users' search queries. We notice that clickbaits use specific words on tags, whereas non-clickbaits do not. Fig. 2 (c) depicts the ratio of the top 20 stems that are found in clickbaits. We observe that many clickbait videos use tags like "try not to laugh", "viral", "hot" and "impossible"; phrases that are usually used for exaggeration.

Statistics. Fig. 2 (d) shows the normalized score of the video statistics for both classes of videos. Interestingly, clickbaits and non-clickbaits videos have similar views; suggesting that viewers are not able to easily discern clickbait videos, hence clicking on them. Also, non-clickbait videos have more likes and less dislikes than clickbaits. This is reasonable as many users feel frustrated after watching clickbaits.

Comments. We notice that users on YouTube implicitly flag suspicious videos by commenting on them. For instance, we note several comments like the following: "the title is misleading", "i clicked because of the thumbnail", "where is the thumbnail?" and "clickbait". Hence, we argue that comments from viewers is a valuable resource for assessing videos. To this end, we analyze the ground truth dataset to extract the mean number of occurrences of words widely used



Category	Clickbaits (%)	Non-clickbaits (%)
Entertainment	406 (38%)	308 (32%)
Comedy	318 (29%)	228 (24%)
People & Blogs	155 (14%)	115 (12%)
Autos & Vehicles	33 (3%)	49 (5%)
Sports	29 (3%)	114 (12%)

Fig. 2 & TABLE I: Analysis of the manually reviewed ground truth dataset. Normalized mean scores for: (a) stems from headline text; (b) tags derived from thumbnails; (c) stems from tags that were defined by uploaders; (d) video statistics; and (e) comments that contain words for flagging suspicious videos. Table 1 shows the top five categories (and their respective percentages) in our ground truth dataset.

Source	Destination	Norm. Mean
clickbait	clickbait	4.1
clickbait	non-clickbait	2.73
non-clickbait	clickbait	2.75
non-clickbait	non-clickbait	3.57

TABLE II: Normalized mean of related videos for clickbait and non-clickbait videos in the ground truth dataset

for flagging clickbait videos. Fig. 2 (e) depicts the normalized mean scores for the identified words. We observe that these words were greatly used in clickbait comments but not in non-clickbaits. Also, it is particularly interesting that comments referring to the video's thumbnail were found 2.5 times more often in clickbait than in non-clickbaits.

Graph Analysis. Users often watch videos according to YouTube's recommendations. From manual inspections, we have noted that when watching a clickbait video, YouTube is more likely to recommend another clickbait video. To confirm this against our data, we create a directed graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ the videos and $\boldsymbol{E}$ the connections between videos pointing to one another via a recommendation. Then, for all the videos, we select their immediate neighbors in the graph, and count the videos that are clickbaits and non-clickbaits. Table II depicts the normalized mean of the number of connected videos for each class. We apply a normalization factor to mitigate the bias towards clickbaits, which have a slightly greater number in our ground truth. We observe that, when a user watches a clickbait video, they are recommended 4.1 clickbait videos on average, as opposed to 2.73 non-clickbait recommendations. A similar pattern holds for non-clickbaits; a user is less likely to be served a clickbait when watching a non-clickbait.

YouTube's countermeasures. To get an insight on whether YouTube employs any countermeasures, we calculate the number of offline (either deleted by YouTube or removed by the uploader) videos in our manually reviewed ground truth, as of January 10, 2017 and April 30, 2017. We found that only 3% (January 10th) and 10% (April 30th) of the clickbaits are offline. Similarly, only 1% (January 10th) and 5% (April 30th) of the non-clickbaits are offline. To verify that the ground truth dataset does not consist of only recent videos (thus, just published and not yet detected) we calculate the mean number of days that passed from the publication date up to January 10, 2017. We find that the mean number of days for the clickbaits is 700, while it is 917 days for the non-clickbaits. The very low offline ratio, as well as the high mean number of days, indicate that YouTube is not able to tackle the problem in a timely manner.

III. Clickbait Detection Model

A. Processed Modalities

Our model processes the following modalities: (i) Headline: For the headline, we consider both the content and the style of the text. For the content of the headline, we use sent2vec embeddings [26] trained on Twitter data. For the style of the text, we use the features proposed in [6]; (ii) Thumbnail: We scale down the images to 28x28 and convert them to grayscale. This way, we decrease the number of trainable parameters for our developed deep network, thus speeding training time up without compromising achievable performance; (iii) Comments: We preprocess user comments to find the number of occurrences of words used for flagging videos. We consider the following words: "misleading, bait, thumbnail, clickbait, deceptive, deceiving, clicked, flagged, title"; (iv) Tags: We encode the tags' text as a binary representation of the top 100

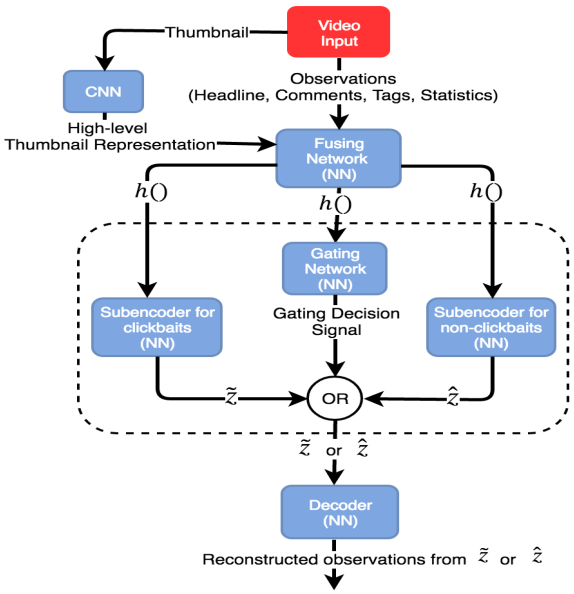


Fig. 3: Overview of the proposed model. The dotted rectangle represents the encoding component of our model.

words that are found in the whole corpus; and (v) Statistics (e.g., likes, views, etc.).

B. Model Formulation

In Fig. 3, we provide an overview of the proposed model. The thumbnail is initially processed, at the encoding part of the proposed model, by a CNN [40]. We use a CNN that comprises four convolutional layers, with 64 filters each, and ReLU activations. The first three of these layers are followed by max-pooling layers. The fourth is followed by a simple densely connected layer, which comprises 32 units with ReLU activations. This initial processing stage allows for learning to extract a high-level, 32-dimensional vector of the thumbnail, which contains the most useful bits of information for driving classification.

The overarching goal of the devised model is to limit the required availability of labeled data, while making the most out of large corpora of (unlabeled) examples. To this end, after this first processing stage, we split the encoding part into two distinct subencoders that work in tandem. Both these subencoders are presented with the aforementioned 32-dimensional thumbnail representation, fused with the data stemming from all the other available modalities. This results in a 855-dimensional input vector, first processed by one dense layer network (Fusing Network) that comprises 300 ReLU units. Due to its large number of parameters, this dense layer network may become prone to overfitting; hence, we regularize using the prominent Dropout technique [34]. We use a Dropout level of $d = 0.5$; this means that, at each iteration of the training algorithm, 50% of the units are randomly omitted from updating their associated parameters. The obtained 300-dimensional vector, say $h()$, is the one eventually presented to both postulated subencoders.

The rationale behind this novel configuration is motivated by a key observation; the two modeled classes are expected to entail significantly different patterns of correlations and latent underlying dynamics between the modalities. Hence, it is plausible that each class can be adequately and effectively modeled by means of distinct, and different, encoder distributions, inferred by the two subencoders. Each of these subencoders are dense-layer networks comprising a hidden layer with 20 ReLU units, and an output layer with 10 units.

Since the devised model constitutes a VAE, the output units of the subencoders are of a stochastic nature; specifically, we consider stochastic outputs, say $\tilde{\boldsymbol{z}}$ and $\hat{\boldsymbol{z}}$, with Gaussian (posterior) densities, as usual in the literature of VAEs [21], [23]. Hence, what the postulated subencoders actually compute are the means, $\tilde{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\mu}}$, as well as the (diagonal) covariance matrices, $\tilde{\boldsymbol{\sigma}}^2$ and $\hat{\boldsymbol{\sigma}}^2$, of these Gaussian posteriors. On this basis, the actual subencoder output vectors, $\tilde{\boldsymbol{z}}$ and $\hat{\boldsymbol{z}}$, are sampled each time from the corresponding (inferred) Gaussian posteriors. Note that our modeling selection of *sharing* the initial CNN-based processing part between the two subencoders allows for *significantly reducing the number of trainable parameters*, without limiting the eventually obtained modeling power.

Under this mixture model formulation, we need to establish an effective mechanism for inferring which observations (i.e., videos) are more likely to match the learned distribution of each component subencoder. This is crucial for effectively selecting between the samples of $\tilde{\boldsymbol{z}}$ or $\hat{\boldsymbol{z}}$ at the output of the encoding stage of the devised model. In layman terms, this can be considered to be analogous to a (soft) classification mechanism. This mechanism can be obtained by computation of the posterior distribution of mixture component membership of each video (also known as "responsibility" in the literature of finite mixture models [25]). To allow for inferring this posterior distribution, in this work we postulate a gating network. This is a dense-layer network, which comprises one hidden layer with 100 ReLU units, and is presented with the same vector, $h()$, as the two postulated subencoders. It is trained alongside the rest of the model, and it is the only part of the model that requires availability of labeled data for its training.

Note that this gating network entails only a modest number of trainable parameters, since both the size of its input as well as of its single hidden layer are rather small. As such, it can be effectively trained even with *limited availability of labeled data*. This is a key merit of our approach, which fully differentiates it from conventional classifiers that are presented with raw observed data, which typically are prohibitively high-dimensional.

To conclude the formulation of the proposed VAE, we need to postulate an appropriate decoder distribution, and a corresponding network that infers it. In this work, we opt for a simple dense-layer neural network, which is fed with the (sampled) output of the postulated finite mixture model encoder, and attempts to reconstruct the original modalities. Specifically, we postulate a network comprising one hidden

layer with 300 ReLU units, and a set of 823 output units, that attempt to reconstruct the original modalities, with the exception of the thumbnail. The reason why we ignore the thumbnail modality from the decoding process is the need of utilizing deconvolutional network layers to appropriately handle it, which is quite complicated. Hence, we essentially treat the thumbnail modality as *side-information,* that regulates the inferred posterior distribution of the latent variables $\boldsymbol{z}$ (encoder) in the sense of a *covariate,* instead of an observed modality in the conventional sense. It is empirically known that such a setup, if appropriately implemented, does not undermine modeling effectiveness [30].

Let us denote as $\boldsymbol{x}_n$ the set of observable data pertaining to the nth available video. We denote as $\boldsymbol{x}_n = \{\boldsymbol{x}_n^{he}, \boldsymbol{x}_n^{th}, \boldsymbol{x}_n^{ta}, \boldsymbol{x}_n^{co}, \boldsymbol{x}_n^{st}\}$ the set of five distinct modalities, i.e., headline, thumbnail, tags, comments, and statistics, respectively. Then, based on the above description, the encoder distribution of the postulated model reads:

$$q(\boldsymbol{z}_n|\boldsymbol{x}_n) = q(\tilde{\boldsymbol{z}}_n|\boldsymbol{x}_n)^{q(c_n=1|\boldsymbol{x}_n)} q(\hat{\boldsymbol{z}}_n|\boldsymbol{x}_n)^{q(c_n=0|\boldsymbol{x}_n)} \tag{1}$$

Here, $\boldsymbol{z}_n$ is the output of the encoding stage of the proposed model that corresponds to $\boldsymbol{x}_n$, $\tilde{\boldsymbol{z}}_n$ is the output of the first subencoder, corresponding to the clickbait class, $\hat{\boldsymbol{z}}_n$ is the output of the second subencoder, corresponding to the non-clickbait class, and c_n is a latent variable indicator of whether $\boldsymbol{x}_n$ belongs to the clickbait class or not. We also postulate

$$q(\tilde{\boldsymbol{z}}_n|\boldsymbol{x}_n) = \mathcal{N}(\tilde{\boldsymbol{z}}_n|\tilde{\boldsymbol{\mu}}(\boldsymbol{x}_n;\tilde{\boldsymbol{\theta}}), \mathrm{diag}\,\tilde{\boldsymbol{\sigma}}^2(\boldsymbol{x}_n;\tilde{\boldsymbol{\theta}})) \tag{2}$$

$$q(\hat{\boldsymbol{z}}_n|\boldsymbol{x}_n) = \mathcal{N}(\hat{\boldsymbol{z}}_n|\hat{\boldsymbol{\mu}}(\boldsymbol{x}_n;\hat{\boldsymbol{\theta}}), \mathrm{diag}\,\hat{\boldsymbol{\sigma}}^2(\boldsymbol{x}_n;\hat{\boldsymbol{\theta}})) \tag{3}$$

Here, the $\tilde{\boldsymbol{\mu}}(\boldsymbol{x}_n;\tilde{\boldsymbol{\theta}})$ and $\tilde{\boldsymbol{\sigma}}^2(\boldsymbol{x}_n;\tilde{\boldsymbol{\theta}})$ are outputs of a deep neural network, with parameters set $\tilde{\boldsymbol{\theta}}$, that corresponds to the clickbait class subencoder; it comprises a first CNN-type part that processes the observed thumbnails, and a further densely-connected network part that fuses and processes the rest of the observed modalities, as described previously. Similarly, the $\hat{\boldsymbol{\mu}}(\boldsymbol{x}_n;\hat{\boldsymbol{\theta}})$ and $\hat{\boldsymbol{\sigma}}^2(\boldsymbol{x}_n;\hat{\boldsymbol{\theta}})$ are outputs of a deep neural network with parameters set $\hat{\boldsymbol{\theta}}$, that corresponds to the non-clickbait class subencoder.

The posterior distribution of mixture component allocation, $q(c_n|\boldsymbol{x}_n)$, which is parameterized by the aforementioned gating network, is a simple Bernoulli distribution that reads

$$q(c_n|\boldsymbol{x}_n) = \mathrm{Bernoulli}(\varpi(h(\boldsymbol{x}_n);\boldsymbol{\varphi})) \tag{4}$$

Here, $\varpi(h(\boldsymbol{x}_n);\boldsymbol{\varphi}) \in [0,1]$ is the output of the gating network, with trainable parameters set $\boldsymbol{\varphi}$. This is presented with an intermediate encoding of the input modalities (shared with the subencoder networks), $h(\boldsymbol{x}_n)$, as described previously, and infers the probability of $\boldsymbol{x}_n$ belonging to the clickbait class.

Lastly, the postulated decoder distribution reads

$$p(\boldsymbol{x}_n|\boldsymbol{z}_n) = \mathcal{N}(\boldsymbol{x}_n^{he}, \boldsymbol{x}_n^{ta}, \boldsymbol{x}_n^{co}, \boldsymbol{x}_n^{st}|\boldsymbol{\mu}(\boldsymbol{z}_n;\boldsymbol{\phi}), \mathrm{diag}\,\boldsymbol{\sigma}^2(\boldsymbol{z}_n;\boldsymbol{\phi})) \tag{5}$$

where the means and diagonal covariances, $\boldsymbol{\mu}(\boldsymbol{z}_n;\boldsymbol{\phi})$ and $\boldsymbol{\sigma}^2(\boldsymbol{z}_n;\boldsymbol{\phi})$, are outputs of a deep network with trainable parameters set $\boldsymbol{\phi}$, configured as described previously.

C. Model Training

Let us consider a training dataset $X = \{\boldsymbol{x}_n\}_{n=1}^N$ that consists of N video samples. A small subset, X^l, of size M of these samples is considered to be labeled, with corresponding labels set $Y = \{y_m\}_{m=1}^M$. Then, following the VAE literature [21], model training is performed by maximizing the evidence lower bound (ELBO) of the model over the parameters set $\{\tilde{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}, \boldsymbol{\varphi}, \boldsymbol{\phi}\}$. The ELBO of our model reads:

$$\begin{aligned}\log p(X) \geq & \mathcal{L}(\tilde{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}, \boldsymbol{\varphi}, \boldsymbol{\phi}|X) = -\sum_{n=1}^N \mathrm{KL}\big[q(\boldsymbol{z}_n|\boldsymbol{x}_n)||p(\boldsymbol{z}_n)\big] \\ & + \gamma \sum_{n=1}^N \mathbb{E}[\log p(\boldsymbol{x}_n|\boldsymbol{z}_n)] + \sum_{\boldsymbol{x}_m \in X^l} \log q(c_m = y_m|\boldsymbol{x}_m)\end{aligned} \tag{6}$$

Here, $\mathrm{KL}\big[q||p\big]$ is the KL divergence between the distribution $q(\cdot)$ and the distribution $p(\cdot)$, while $\mathbb{E}[\cdot]$ is the (posterior) expectation of a function w.r.t. its entailed random (latent) variables. Note also that, in the ELBO expression (6), the introduced hyperparameter γ is a simple regularization constant, employed to ameliorate the overfitting tendency of the postulated decoder networks, $p(\boldsymbol{x}_n|\boldsymbol{z}_n)$. We have noticed that this simple trick yields a significant improvement in generalization capacity.

In Eq. (6), the posterior expectation of the log-likelihood term $p(\boldsymbol{x}_n|\boldsymbol{z}_n)$ cannot be computed analytically, due to the nonlinear form of the decoder. Hence, we must approximate it by drawing Monte-Carlo (MC) samples from the posterior (encoder) distributions (2)-(3). However, MC gradients are well-known to suffer from high variance. To resolve this issue, we utilize a smart re-parameterization of the drawn MC samples. Specifically, following the related derivations in [21], we express these samples in the form of a differentiable transformation of an (auxiliary) random noise variable $\boldsymbol{\epsilon}$; this random variable is the one we actually draw MC samples from:

$$\tilde{\boldsymbol{z}}_n^{(s)} = \tilde{\boldsymbol{\mu}}_n + \tilde{\boldsymbol{\sigma}}_n \cdot \boldsymbol{\epsilon}_n^{(s)},\ \boldsymbol{\epsilon}_n^{(s)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{7}$$

$$\hat{\boldsymbol{z}}_n^{(s)} = \hat{\boldsymbol{\mu}}_n + \hat{\boldsymbol{\sigma}}_n \cdot \boldsymbol{\epsilon}_n^{(s)} \tag{8}$$

Hence, such a re-parameterization reduces the computed expectations into averages over samples from a random variable with low (unitary) variance, $\boldsymbol{\epsilon}$. This way, by maximizing the obtained ELBO expression, one can yield low-variance estimators of the sought (trainable) parameters, under some mild conditions [21]. Turning to the maximization process of $\mathcal{L}(\tilde{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}, \boldsymbol{\varphi}, \boldsymbol{\phi}|X)$, this can be effected using modern stochastic optimizations, such as AdaGrad [16] or RmsProp [39].

IV. Experimental Evaluation

A. Experimental Setup

Our model is implemented with Keras [12] and TensorFlow [24].

MC Samples. To perform model training, we used $S = 10$ drawn MC samples, $\boldsymbol{\epsilon}^{(s)}$; we found that increasing this value does not yield any statistically significant accuracy improvement, despite the associated increase in computational costs.

Initialization. We employ Glorot Uniform initialization [18]. This scheme allow us to train deep network architectures without the need of layerwise pretraining. This is effected by initializing the network weights in a way which ensures that the signal propagated across the network layers remains in a reasonable range of values, irrespectively of the network depth (i.e., it does not explode to infinity or vanish to zero).

Stochastic optimization. We found that RmsProp works better than the AdaGrad algorithm suggested in [21]. The RmsProp hyperparameter values we used comprise an initial learning rate equal to 0.001, $\rho = 0.9$, and $\varepsilon = 10^{-8}$.

Prediction generation. To predict the class of a video, $\boldsymbol{x}_n$, we compute the mixture assignment posterior distribution $q(c_n|\boldsymbol{x}_n)$, inferred via the postulated gating network $\varpi(h(\boldsymbol{x}_n); \boldsymbol{\varphi})$. On this basis, assignment is performed to the clickbait class if $\varpi(h(\boldsymbol{x}_n); \boldsymbol{\varphi}) > 0.5$.

Baselines. To evaluate our model, we compare it against two baseline models. First, a simple Support Vector Machine (SVM) with parameters $\gamma = 0.001$ and $C = 100$. Second, a supervised deep network (SDN) that comprises (i) the same CNN as the proposed model; and (ii) a 2-layer fully-connected neural network with Dropout level of $d = 0.5$.

Evaluation. We train our proposed model using the entirety of the available unlabeled dataset, as well as a randomly selected 80% of the available labeled dataset, comprising an equal number of clickbait and non-clickbait examples. Subsequently, the trained model is used to perform out-of-sample evaluation; that is, we compute the classification performance of our approach on the fraction of the available labeled examples that were not used for model training.

Model	Accuracy	Precision	Recall	F1 Score
SVM	0.882	0.909	0.884	0.896
SDN	0.908	0.920	0.907	0.917
Proposed Model (U = 25%)	0.915	0.918	0.926	0.923
Proposed Model (U = 50%)	0.918	0.918	0.934	0.926
Proposed Model (U = 100%)	**0.924**	**0.921**	**0.942**	**0.931**

TABLE III: Performance metrics for the evaluated methods. We also report the performance of our model when using only 25% or 50% of the available unlabeled data.

Table III reports the performance of the proposed model as well as the two considered baselines. We observe that neural network-based approaches, such as a simple neural network and the proposed model, outperform SVMs in terms of all the considered metrics. Specifically, the best performance is obtained by the proposed model, which outperforms SVMs by 3.8%, 1.2%, 5.8% and 3.5% on accuracy, precision, recall, and F1 score, respectively. Further, to assess the importance of using unlabeled data, we also report results with reduced unlabeled data. We observe that, using only 25% of the available unlabeled data, the proposed model undergoes a substantial performance decrease, as measured by all the employed performance metrics. This performance deterioration only slightly improves when we elect to retain 50% of the available unlabeled data.

Corpus-Level Inference Insights. Having demonstrated the performance of our model, we now provide some insights into the obtained inferential outcomes on the whole corpus of data. From the 206k examples, our model predicts that 84k (41%) of them are clickbaits whereas 122k (59%) are non-clickbaits. The considerable percentage of clickbaits in the corpus, in conjunction with the data collection procedure, suggests that, with respect to the collected data, YouTube does not consider misleading videos in their recommendations. Note also that we have performed an analysis of the whole corpus akin to the ground truth dataset analysis of Section II.A. The obtained results follow the same pattern as for the ground truth dataset. Specifically, we have found that the normalized mean values, reported in Table II for the labeled data, for the whole corpus become equal to 11.18, when it comes to pairs of clickbait videos, whereas for clickbait and non-clickbaits the mean is 2.62. This validates our deduction that it is more likely to be recommended a clickbait video when viewing a clickbait video on YouTube.

V. Related Work

The clickbait problem is also identified by prior work that proposes tools for alleviating the problem in various web portals. Specifically, Chen et al. [11] provide useful information regarding the clickbait problem and future directions for tackling the problem using SVM and Naive Bayes approaches. Rony et al. [32] analyze 1.67M posts on Facebook in order to understand the extent and impact of the clickbait problem as well as users' engagement. For detecting clickbaits, they propose the use of sub-word embeddings with a linear classifier. Potthast et al. [31] focus on the Twitter platform where they suggest the use of Random Forests for distinguishing tweets that contain clickbait content. Furthermore, Chakraborty et al. [9] propose the use of SVMs in conjunction with a browser add-on for offering a detection system to end-users of news articles. Moreover, Biyani et al. [6] recommend the use of Gradient Boosted Decision Trees for clickbait detection in news articles. They also demonstrate that the degree of informality in the content of the landing page can help in discerning clickbait news articles. To the best of our knowledge, Anand et al. [4] is the first work that suggests the use of deep learning techniques for mitigating the clickbait problem. Specifically, they propose the use of Recurrent Neural Networks in conjunction with word2vec embeddings for identifying clickbait news articles. Agrawal [3] propose the use of CNNs in conjunction with word2vec embeddings for discerning clickbait headlines in Reddit, Facebook and Twitter. Other efforts include browser add-ons [15], [13], [7] and manually associating user accounts with clickbait content [33], [35], [5].

Remarks. In contrast to the aforementioned works, we focus on the YouTube platform and propose a deep learning model that: (i) successfully and properly fuse and correlate the diverse set of modalities related to a video; and (ii) leverage large unlabeled datasets, while imposing much limited requirements in labeled data availability.

VI. Conclusion

In this work, we have explored the use of variational autoencoders for tackling the clickbait problem on YouTube. Our approach constitutes the first proposed semi-supervised deep learning technique in the field of clickbait detection. This way, it enables more effective automated detection of clickbait videos in the absence of large-scale labeled data. Our analysis indicates that YouTube recommendation engine does not take into account the clickbait problem in its recommendations.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 Research and Innovation program, under the Marie Skłodowska-Curie ENCASE project (Grant Agreement No. 691025). We also gratefully acknowledge the support of NVIDIA Corporation, with the donation of the Titan Xp GPU used for this research.

References

[1] Clickbait video. https://www.youtube.com/watch?v=W2WgTE9OKyg.
[2] List of ground truth channels. https://goo.gl/ZSabkn, 2018.
[3] A. Agrawal. Clickbait detection using deep learning. In *NGCT*, 2016.
[4] A. Anand, T. Chakraborty, and N. Park. We used Neural Networks to Detect Clickbaits: You won't believe what happened Next! *arXiv preprint arXiv:1612.01340*, 2016.
[5] Anti-Clickbait. https://twitter.com/articlespoiler, 2015.
[6] P. Biyani, K. Tsioutsiouliklis, and J. Blackmer. "8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality. 2016.
[7] B.s detector browser extension. http://bsdetector.tech/.
[8] R. Campbell. You Wont Believe How Clickbait is Destroying YouTube!, 2017. https://tritontimes.com/11564/columns/you-wont-believe-how-clickbait-is-destroying-youtube/.
[9] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly. Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media. 2016.
[10] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou. Signal Modeling and Classification Using a Robust Latent Space Model Based on t Distributions. *TOSP*, 2008.
[11] Y. Chen, N. J. Conroy, and V. L. Rubin. Misleading Online Content: Recognizing Clickbait as False News. In *ACM MDD*, 2015.
[12] F. Chollet. Keras. https://github.com/fchollet/keras, 2015.
[13] Clickbait remover for Facebook. http://bit.ly/2DIrpj6.
[26] M. Pagliardini, P. Gupta, and M. Jaggi. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. *arXiv*, 2017.
[14] J. Constine. Facebook feed change fights clickbait post by post in 9 more languages, 2017. http://archive.is/l8hIt.
[15] Downworthy browser extension. http://downworthy.snipe.net/.
[16] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 2010.
[17] H. Gao, Y. Chen, K. Lee, D. Palsetia, et al. Towards Online Spam Filtering in Social Networks. In *NDSS*, 2012.
[18] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.
[19] Imagga. Tagging Service, 2016. https://imagga.com/.
[20] N. Jindal and B. Liu. Review spam detection. In *WWW*, 2007.
[21] D. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014.
[22] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.
[23] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary Deep Generative Models. In *ICML*, 2016.
[24] A. Martın, A. Ashish, B. Paul, B. Eugene, C. Zhifeng, C. Craig, C. G. S, D. Andy, D. Jeffrey, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
[25] G. McLachlan and D. Peel. *Finite Mixture Models*. 2000.
[27] A. Peysakhovich. Reducing clickbait in facebook feed, 2016. http://newsroom.fb.com/news/2016/08/news-feed-fyi-further-reducing-clickbait-in-feed.
[28] Piperjaffray. Survey, 2016. http://archive.is/AA34y.
[29] E. A. Platanios and S. P. Chatzis. Gaussian Process-Mixture Conditional Heteroscedasticity. *TPAMI*, 2014.
[30] I. Porteous, A. Asuncion, and M. Welling. Bayesian Matrix Factorization with Side Information. In *AAAI*, 2010.
[31] M. Potthast, S. Köpsel, B. Stein, and M. Hagen. Clickbait Detection. In *ECIR*, 2016.
[32] M. M. U. Rony, N. Hassan, and M. Yousuf. Diving Deep into Clickbaits: Who Use Them to What Extents in Which Topics with What Effects? *arXiv:1703.09400*, 2017.
[33] SavedYouAClick. https://twitter.com/savedyouaclick, 2014.
[34] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *JMLR*, 2014.
[35] StopClickBait. http://stopclickbait.com/, 2016.
[36] G. Stringhini, M. Egele, A. Zarras, T. Holz, C. Kruegel, and G. Vigna. B@bel: Leveraging Email Delivery for Spam Mitigation. In *USENIX Security*, 2012.
[37] G. Stringhini, T. Holz, B. Stone-Gross, C. Kruegel, and G. Vigna. BOTMAGNIFIER: Locating Spambots on the Internet. In *USENIX Security*, 2011.
[38] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *CSA*, 2010.
[39] T. Tieleman and G. Hinton. Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude. 2012.
[40] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV*, 2014.