Online Supplementary Information for "Automated Coding of Political Campaign Advertisement Videos: An Empirical Validation Study."

Alexander Tarr[*] June Hwang[†] Kosuke Imai[‡]

May 2, 2022

[*] Graduate Student, Department of Electrical Engineering, Princeton University, Princeton NJ 08544. Phone: 978–270–8483, Email: atarr@princeton.edu

[†] Advisor, Consulate General of the Republic of Korea, Honolulu HI 96817. Phone: 609–751–7028, Email: wjhwangusa@gmail.com

[‡] Professor, Department of Government and Department of Statistics, Harvard University, Cambridge MA 02138. Phone: 617–384–6778, Email: Imai@Harvard.Edu, URL: https://imai.fas.harvard.edu

Cycle	Office	All Candidates	Candidates with YT Channels	Duration			Total
				15s	30s	60s	
2012	President	2	2 (100.0%)	40	263	97	400
	House	317	242 (76.3%)	39	986	198	1223
	Senate	64	48 (75.0%)	16	519	146	681
	Governor	25	20 (80.0%)	15	143	36	194
2014	House	250	191 (76.4%)	53	804	173	1030
	Senate	66	52 (78.8%)	57	714	202	973
	Governor	86	58 (67.4%)	70	636	170	876
	Total	810	613 (75.7%)	290	4065	1022	5377

Table S1.1: Summary of Channels and Video Files Found at YouTube. The table presents the number of candidates for each office listed in the Wesleyan Media Project (WMP) data, the number and percentage of these candidates for whom we found YouTube (YT) channels, and the number of downloaded video files from the said candidates (different types based on their length). Note that not all of the downloaded videos are campaign TV advertisements.

S1 YouTube Coverage

Table S1.1 summarizes the channels and video files found at YouTube using the procedure described in the main text. We find that approximately 75% of the general election candidates listed in the WMP data have official YouTube channels. In addition, a greater proportion of House candidates have YouTube channels than the Senate and Gubernatorial candidates. Lastly, we found a total of several thousand video files that have approximately the same lengths as those of campaign TV advertisement videos. The majority of video files are 30 seconds long.

A potential concern about using YouTube as the data source is its insufficient coverage. Although a majority of general election candidates set up and actively operate YouTube channels during their election bids (see Table S1.1), there still are a significant number of those who do not. In addition, some politicians decide to close down their channel or take down some or all of the previous campaign TV ads after the conclusion of the election. While missingness is likely a minor issue in the context of our analysis, we note that the performance of our algorithm as reported in this paper may not generalize to all videos in the 2012 and 2014 elections cycles.

Although we found official YouTube channels for about 75% of all general election candidates in the WMP data set, after removing the YouTube video files we were unable to match with any CMAG video, the proportion of the candidates who have at least one matched video file is reduced to approximately 65%. The coverage rate has generally improved from 2012 to 2014, except for gubernatorial elections, suggesting that the coverage may continue to improve over time as more political campaigns start using YouTube as a way to reach voters.

S2 Spectral Fingerprinting

The spectrogram is a two-dimensional representation of the frequency content of the signal as it varies with time. This frequency representation of the signal provides important information as to

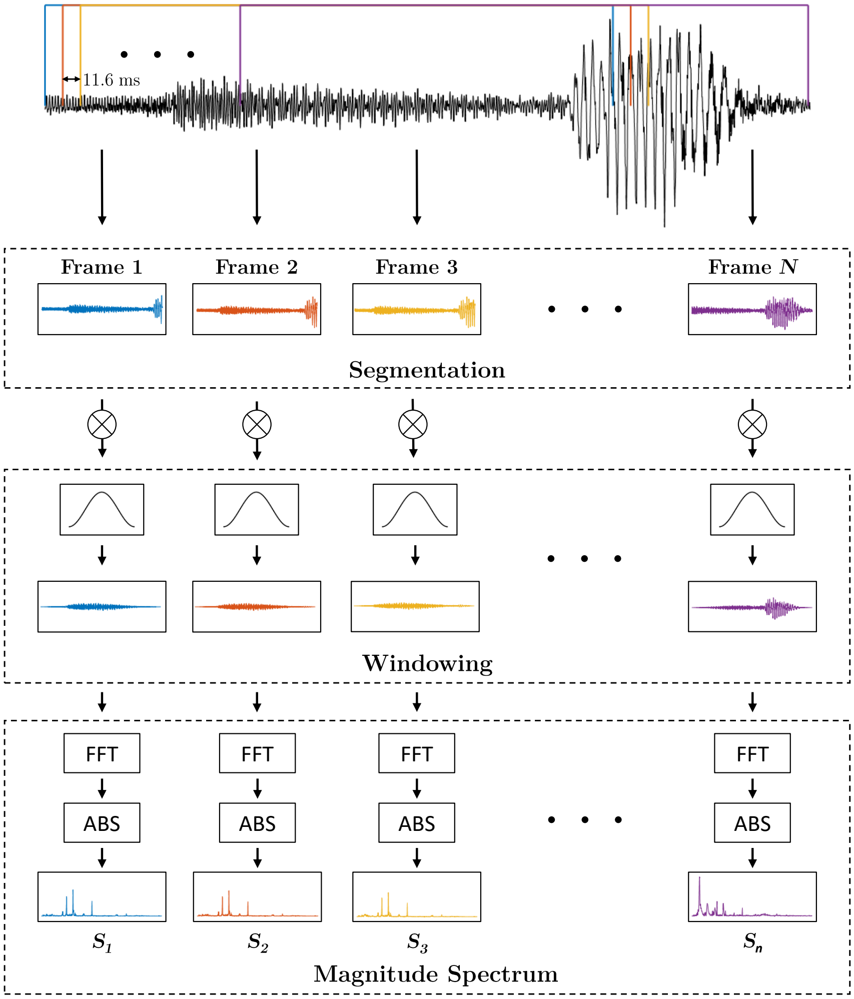


Figure S2.1: System Diagram for Computing the Spectrogram. The diagram illustrates the process of computing the spectrogram for a 0.5 second segment of audio for one of the campaign videos in our validation dataset. The signal is first partitioned into segments of length 0.3712 seconds, spaced apart by 11.6 milliseconds (ms), resulting in a collection of N frames, shown in the box labeled "Segmentation." Each segment is then multiplied by a Hann window to produce a smoothed waveform, as shown in the "Windowing" box. Finally, in the last box the magnitude spectrum is computed via a fast Fourier transform (FFT) followed by an absolute value (ABS). The magnitude spectrum for frame j corresponds to row j in the spectrogram and is denoted as S_j.

the perceptual qualities (i.e., tempo, timbre, pitch, etc.) of the audio track. For each unit of time in the spectrogram (also called a frame), we summarize the frequency content using a 32-bit unsigned integer derived from the frequency distribution and call the resulting vector the fingerprint. Each element of this vector is called the sub-fingerprint.

Figure S2.1 illustrates the entire process, and we begin with a brief introduction to audio signal processing. The audio we hear in a video is a continuous waveform that propagates through the air as a compression wave, vibrating the air molecules around our ears, which creates the sound.

These compression waves are generated from the vibration of speakers in our sound device, with the pattern for the speaker vibration dictated by a continuous electrical signal, called the analog signal. Since the video files cannot store all the values of the continuous analog signal, a discretized approximation is used by sampling the desired continuous waveform at fixed intervals of time, where the number of samples obtained per second of audio is called the sampling rate. For the matching procedure, we used digital audio signals from the videos with a sampling rate of 5 kHz, implying that for every one second of audio, we obtain 5,000 equally-spaced samples. A more detailed introduction on digital signal processing can be found in Oppenheim, Willsky and Nawab (1996).

Formally, given a digital audio signal $x[l]$ consisting of L samples, where L is determined by the duration of the audio file, the spectrogram is computed by first splitting the audio signal into overlapping segments, each of which has a length of $W = 1,856$ samples, corresponding to duration 0.3712 ($= W/5000$) seconds. We use short-duration segments because the frequency content of audio signals varies quickly over time. For example, music typically consists of several different instruments playing a sequence of notes in rapid succession. Splitting the full signal into smaller segments allows us to better isolate the individual instruments and notes, allowing us to better characterize the short-term variations of the audio signal.

We use an overlap factor of 31/32, meaning that consecutive segments share 31/32 of the same samples. Hence, the temporal spacing between the first sample of each consecutive segment is $11.6\text{ms} = 0.3712/32$. The reason we use a large overlap factor is that it helps with the matching procedure, as we explain later. Each frame is then multiplied by a Hann window to reduce high frequency noise introduced due to the segmentation (Harris, 1978). Finally, we compute the one-sided magnitude spectrum for each frame by taking the absolute value of the 2048-point fast Fourier transform (FFT) of the input frame. This yields a spectrogram $S \in \mathbb{R}^{N \times K}$, with element $S(n, k)$ corresponding to the magnitude of frequency component k in frame n. Note that the number of frequency bins K is fixed to 1025 for all fingerprints, while the time length N varies with the duration of the audio signal.

To reduce the dimensionality of frequency, we partition the resulting spectrogram into $M = 33$ non-overlapping, logarithmically-spaced frequency bands covering the frequency range 300Hz to 2,000Hz. Within each band m, we compute the energy, defined as

$$E(n, m) \;=\; \sum_{k=1}^{K_m} S(n, k)^2,$$

where K_m is the number of frequency bins. This step yields a matrix of energy values for each frequency band and spectrogram time segment. We then produce a binarized matrix $B \in \mathbb{R}^{N-1 \times M-1}$ through the following rule,

$$B(n, m) \;=\; \begin{cases} 1 & \text{if } E(n+1, m) - E(n+1, m+1) - (E(n, m) - E(n, m+1)) > 0 \\ 0 & \text{if } E(n+1, m) - E(n+1, m+1) - (E(n, m) - E(n, m+1)) \leq 0 \end{cases}, \tag{S1}$$

which indicates the sign of energy differences between adjacent bands and frames. Since there are 33 frequency bands, this procedure results in a 32-element binary array for each $n \in \{1, \ldots, N-1\}$. We convert these binary arrays into 32-bit unsigned integers and call the resulting vector F the fingerprint, with each element of the vector called the sub-fingerprint.

S3 Audio Matching Procedure

Matching was done in several steps. Here is a brief summary. We first generate a list of candidate matches for the unidentified video using a hash table to find videos in the database which contain the same sub-fingerprints as the unlabeled video. Then, for each candidate, we compute the bit error rate (BER) between the unmatched fingerprint for the middle eight seconds of a given video and that of the candidate fingerprint, which equals the average number of binary digits that differ between the two fingerprints. Largely following Haitsma and Kalker (2002), we declare the candidate fingerprint to be a match to the CMAG video if the BER was below a threshold of 0.35. If the candidate was not a match, we repeat this process for the next candidate until a match is found or all candidates have been exhausted. The two videos also have to fall in the same length range to be declared as matches.

We now explain each step in more detail. For each YouTube video, we compute the spectral fingerprints for the full-duration of all of the downloaded YouTube videos, producing a lookup table (LUT) of fingerprints to be used to match the CMAG videos to. The hash values in the LUT represent the sub-fingerprints while the values stored are the ID of the file and the index in the full fingerprint where the sub-fingerprint occurs. Note that because the videos have varying lengths, the spectral fingerprints will have varying lengths as well. For each unmatched CMAG video, we then compute the spectral fingerprint of the middle 12-second segment of the unmatched CMAG audio track and attempt to find its matching fingerprint in the YouTube video database.

There are several reasons why the duration input audio signal to the spectral fingerprinting process is fixed to 12 seconds. First, the entire clip is not needed to find a match, so using a shorter segment leads to significant computational performance. Second, many of the CMAG videos were improperly recorded and often contain extra silence or portions of other ads at the boundaries of file (the end or the beginning), so using the entire audio file would lead to false negatives in the matching process because, technically, the audio files contained different content. Third, while shorter duration could be used to effectively match videos, we found that some of the ads in the CMAG dataset had some segments that were identical to one another, while other segments were different. By using longer duration clips for matching, we reduce the probability of false positives.

Since the sub-fingerprints are computed at discrete intervals of time, there is no guarantee that segments obtained in the fingerprinting process of the truncated audio correspond to the same segments in its matching file in the YouTube database. We see now that the purpose of the large overlap factor is mitigate issues with spectrogram frame boundary misalignment between the CMAG video and the YouTube database video in the fingerprinting process. Since the spacing between frames is so small, the sub-fingerprints for the unmatched video should be very similar to the sub-fingerprints of its corresponding match in the database, even under worst-case misalignment.

Figure S3.2 shows the layout for the fingerprint database we use for matching. We will refer to this figure throughout this section as we explain the procedure for matching. Denote F_{unk} as the fingerprint we are trying to match, recalling that a fingerprint is an N-dimensional array of 32-bit unsigned integers, where N is determined by the length of the unmatched audio clip.[1] Also define $F_{\text{unk}}(i)$ as the ith sub-fingerprint, depicted by the small rectangles containing integers

[1] For a 12-second clip, $N = 1035$.

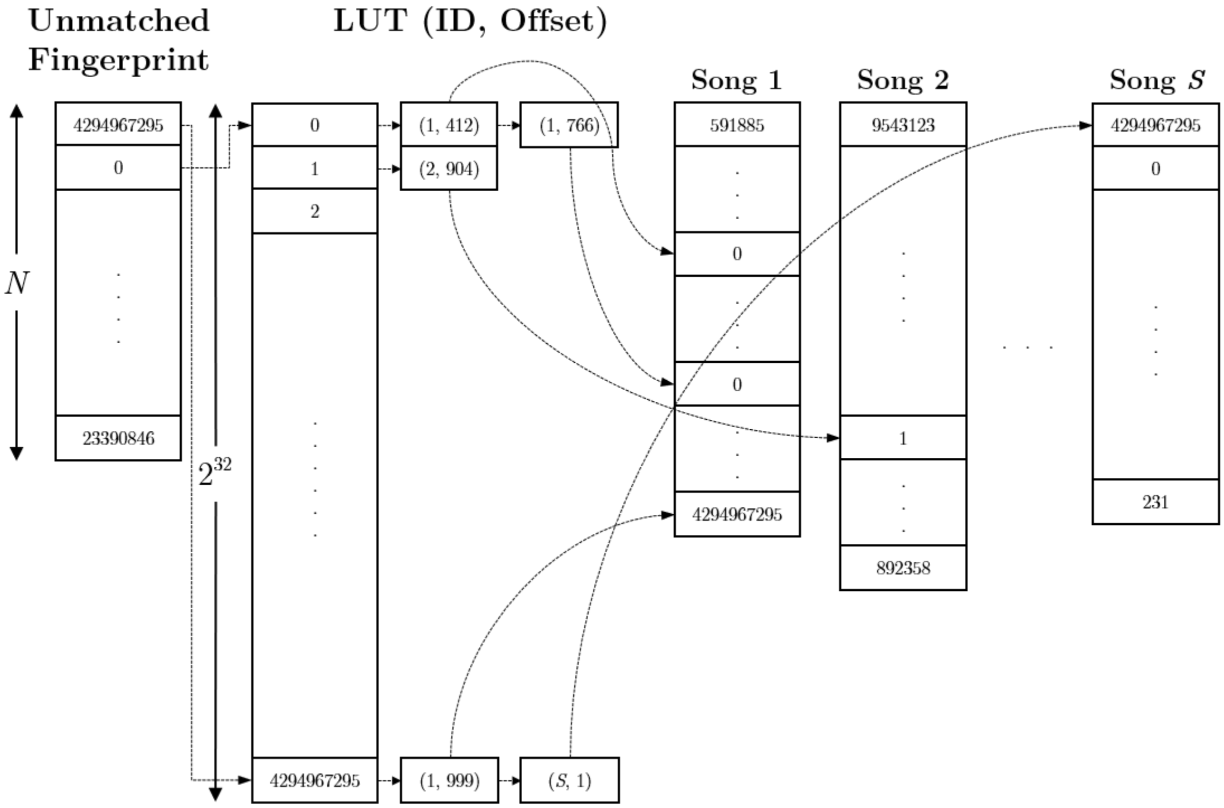


Figure S3.2: Diagram for fingerprint database structure and how candidates are chosen for the unmatched fingerprint. The small rectangles containing integers correspond to sub-fingerprints, while the small rectangles containing the (Song ID, Offset) tuples correspond to the values stored in the database. This figure is a modified version of Figure 6 in (Haitsma and Kalker, 2002).

in Figure S3.2. Finding the corresponding match to F_{unk} is an iterative procedure. For each sub-fingerprint $F_{\text{unk}}(i)$, we perform the following steps:

1. Generate a list of candidate matches using the current sub-fingerprint $F_{\text{unk}}(i)$. We extract the list of C tuples $\{(\text{SongID}_j, \text{Offset}_j)\}_{j=1}^{C}$ corresponding to $F_{\text{unk}}(i)$ in the LUT. Each tuple indicates that $F_{\text{unk}}(i)$ occurred in SongID at Offset_j. For example, $F_{\text{unk}}(2)$ occurred in Song 1 at offsets 412 and 766, as indicated by the arrows in Figure S3.2.

2. Given candidate j, obtain the fingerprint for SongID_j and truncate it to length N in way so that $F_{\text{unk}}(i)$ occurs in the same position of the truncated fingerprint as it does in F_{unk}. We see that $F_{\text{unk}}(2) = 0$ occurs in the second position of F_{unk}, while it occurs in the 412th and 766th position of Song 1. In order to compare the shorter, length N unknown fingerprint to Song 1, we need to truncate and align the fingerprint. Denote the truncated and aligned fingerprint as F_j^{trunc}.

3. Compute the bit error rate (BER) between F_{unk} and F_j^{trunc}, where the BER is the the average number of binary digits that differ between the two fingerprints.

4. Declare a match if $BER < 0.3$. Otherwise repeat steps 2–4.

We repeat steps 1–4 for each sub-fingerprint in F_{unk} until either a match is found or until we've checked all sub-fingerprints, at which point we declare no match.[2]

S4 Video Summarization

We first briefly summarize the video summarization algorithm. For any given video consisting of a set of N frames, denoted by V, the optimal summary maximizes uniqueness and representativeness while keeping the number of frames in the summary at a reasonable level. Here, the uniqueness quantifies how distinct frames are from one another in the summary, while the representativeness enforces the rule that for each frame in the video, there should exist at least one frame in the summary that is visually similar. The component for summary length is used to regularize the objective function since uniqueness and representativeness are maximized by selecting all frames in the video. Formally, this optimization problem is written as,

$$S^* = \operatorname*{argmax}_{S \subseteq V} \underbrace{\sum_{i \in V} \max_{j \in S} w_{ij}}_{\text{representativeness}} + \lambda_1 \underbrace{\sum_{i \in S} \min_{j \in S} d_{ij}}_{\text{uniqueness}} + \lambda_2 \underbrace{(N - N_S)}_{\substack{\text{\# of unselected} \\ \text{frames}}}, \tag{S2}$$

where w_{ij} is the cosine similarity between pixel values of frames i and j, d_{ij} is the chi-squared distance between the color histograms of frames i and j, N_S is the number of frames selected for summary S, and (λ_1, λ_2) control the relative weighting of the different terms.

In order to compute the representativeness w_{ij} and the uniqueness d_{ij} between two frames i and j, we first compute a feature representation for each frame. For the representativeness w_{ij}, we use a feature descriptor commonly used in computer vision for object detection called the histogram of oriented gradients (HOG) (Dalal and Triggs, 2005), which encodes the distribution of gradient directions for local color intensity. We compute pairwise distances between all frames in the video using the cosine similarity measure. For the uniqueness d_{ij}, we use the *Lab* histogram. *Lab* is a three-component color space that is more perceptually uniform with respect to human color vision than the standard RGB representation of images. We compute a three-dimensional histogram using 23 bins along each color dimension (L, a, and b) for every frame of a video file. To measure the distance d_{ij} between the *Lab* histogram representations of frames i and j, we use the additive χ^2 kernel,

$$d_{ij} = \sum_{b=1}^{23^3} \frac{(h_i(b) - h_j(b))^2}{h_i(b) + h_j(b)}$$

where $h_i(b)$ and $h_j(b)$ are the counts in bin b for frames i and j, respectively.

Since the problem of finding the optimal subset that maximizes the objective function in equation (S2) is known to be NP-hard, we use an approximation algorithm proposed by Chakraborty, Tickoo and Iyer (2015). The algorithm works by maintaining two solution sets initialized to $S_0 = \emptyset$ and $S_1 = V$. At each iteration, it randomly selects a frame without replacement from V and

[2] To control for the possibility of errors in computing the fingerprint of the unmatched video, we follow the suggestion of Haitsma and Kalker (2002) and search over sub-fingerprints generated by flipping the three-most unreliable bits. Here, unreliable bits refer to the three bits for which the energy differences given in equation (S1) are nearest the decision boundary of 0.

proposes either adding the frame to S_0 or removing it from S_1 with complementary probabilities based the relative change in the objective function that would result from the operation. After all frames are removed from V, the sets S_0 and S_1 will coincide, yielding a video summary. While the algorithm does not necessarily return the optimal summary, the flexibility in choosing the tuning parameters lets us to control the coarseness of the resulting summary. Following the suggestion given in Chakraborty, Tickoo and Iyer (2015), we set the tuning parameters to be, $\lambda_1 = 1$ and $\lambda_2 = 5$.

S5 Description of Face Recognition Algorithms

The face detection algorithm takes an image $\mathbf{x}$ as input and produces a set of non-overlapping bounding boxes $B = \{\mathbf{b}_i \in \mathbb{R}^4 \mid i = 1, 2, \ldots, k\}$, where k is the number of detected faces. This algorithm uses a CNN trained to learn features $\phi(\mathbf{x}, \mathbf{b})$ and a parameter vector $\mathbf{w}$ such that the set

$$B_* = \underset{B}{\operatorname{argmax}} \sum_{\mathbf{b} \in B} \mathbf{w}^\top \phi(\mathbf{x}, \mathbf{b})$$

corresponds to all bounding boxes for faces in image $\mathbf{x}$. This classification rule is enforced in the neural network by using a loss layer derived from the MMOD algorithm (King, 2015).

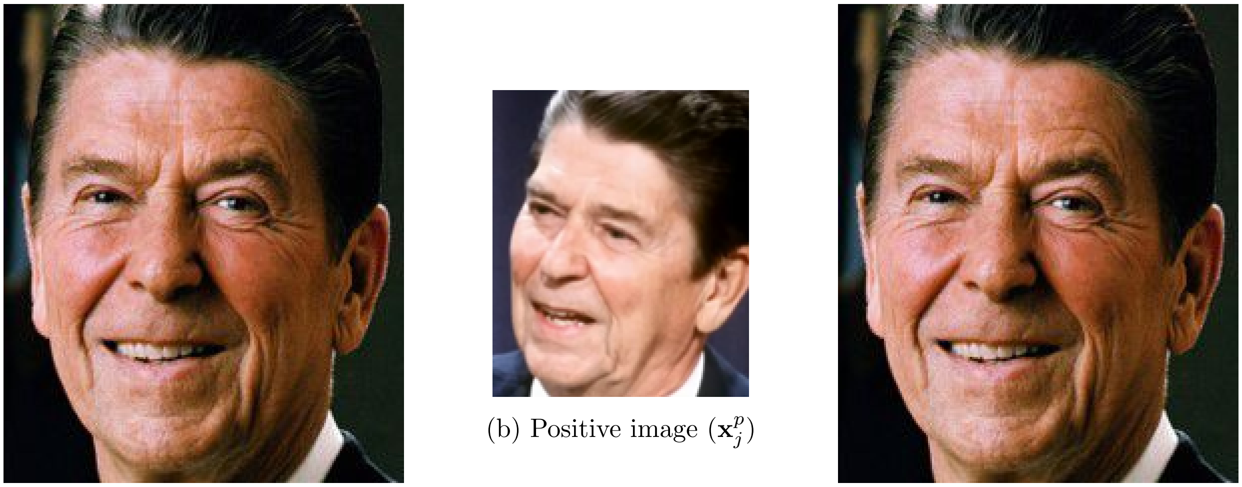
Training the network is achieved through the backpropagation algorithm and stochastic gradient descent. The author of `dlib` trained the network using a composite dataset of about 7000 images taken from several datasets, including WIDER FACE (Yang et al., 2016) and VGGFace (Parkhi, Vedaldi and Zisserman, 2015). Faces selected into the training set were chosen to reflect a wide variation in face poses in order to increase robustness in detection.

The next step in the face recognition system is face alignment, which takes an face image $\mathbf{x}$ as input and applies scaling, rotation, and translation matrices to the face so that the eyes are level and the face is centered and scaled to a specific size. The form of these matrices is determined by the locations of the facial landmarks estimated with a landmark detector. The `dlib` implementation for face alignment uses a 68-point facial landmark detector trained on the iBUG 300-W face landmark dataset (Sagonas et al., 2016). The detector takes a face image $\mathbf{x}$ as input and returns a shape vector $\mathbf{s} \in \mathbb{R}^{136}$ which contains the (x, y)-coordinates of all 68 landmarks. Estimation of the shape vector is done by a cascade of tree regressors, i.e. for an ensemble of K trees, the shape vector for image $\mathbf{x}$ is estimated as

$$\widehat{\mathbf{s}}_{\mathbf{x}}^{(K)} = \widehat{\mathbf{s}}_{\mathbf{x}}^{(0)} + \sum_{k=1}^{K} r_k(\mathbf{x}, \widehat{\mathbf{s}}_{\mathbf{x}}^{(k-1)}),$$

where $\widehat{\mathbf{s}}_{\mathbf{x}}^{(0)}$ is an initial estimate of the shape vector, and $r_k(\mathbf{x}, \widehat{\mathbf{s}}_{\mathbf{x}}^{(k-1)})$ is the kth regression tree which updates the previous shape estimate $\widehat{\mathbf{s}}_{\mathbf{x}}^{(k-1)}$. The core idea behind this algorithm is that each subsequent regression tree refines and improves the previous shape estimate. Each tree in the ensemble of regression trees is trained using the gradient tree boosting algorithm (see Kazemi and Sullivan, 2014, for details).

The FaceNet algorithm also uses a CNN. The goal of this network is to compute a vector representation of the given aligned face image that is close in Euclidean distance to the vector representation of other images pertaining to the same person, and far from images of other people. This is accomplished using a CNN with architecture similar to Google's Inception ResNet-34 (He et al., 2016) trained on a composite dataset consisting of about 3 million faces taken from the



(a) Anchor image ($\mathbf{x}_j^a$)

(b) Positive image ($\mathbf{x}_j^p$)

(c) Negative image ($\mathbf{x}_j^n$)

Figure S5.3: An Example of a Hard-to-classify Triplet Pair. The negative image (c) depicts Sam Brownback who was a republican candidate for the 2014 gubernatorial election in Kansas. The anchor image (a) is a picture of Ronald Reagan taken from Wikipedia, and the positive image (b) comes from a campaign ad for Democratic candidate Mary Burke in the 2014 gubernatorial election in Wisconsin.

VGGFace dataset (Parkhi, Vedaldi and Zisserman, 2015), the face scrub dataset (Ng and Winkler, 2014), and a collection of faces personally chosen and labeled by the `dlib` package author. The collection was chosen to have faces containing many variations in pose, emotional expression, illumination, and occlusion to allow for more robust face embeddings.

The network is trained to learn an embedding $f(\mathbf{x}) \in \mathbb{R}^{128}$ for a given face image $\mathbf{x}$. The training process uses a *triplet loss* function, which, for a given dataset of face images and identities $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, is defined as,

$$L_\beta(\mathcal{D}) = \sum_{j=1}^{N_{\text{trip}}} \max\left(0, ||f(\mathbf{x}_j^a) - f(\mathbf{x}_j^p)||^2 - ||f(\mathbf{x}_j^a) - f(\mathbf{x}_j^n)||^2 + \alpha\right),$$

where the sum is taken over all *triplet pairs* $\{(\mathbf{x}_j^a, \mathbf{x}_j^p, \mathbf{x}_j^n)\}_{j=1}^{N_{\text{trip}}}$ of images. For a given image $\mathbf{x}_j^a$ from the training dataset, called the anchor image, a single triplet pair is formed by selecting another positive image $\mathbf{x}_j^p$, which corresponds the same person shown in $\mathbf{x}_j^a$, and a third negative image $\mathbf{x}_j^n$, which corresponds to a different person. The parameter α corresponds to the margin between the distances and works to enforce a classification rule $\|f(\mathbf{x}_j^a) - f(\mathbf{x}_j^p)\|^2 + \alpha < \|f(\mathbf{x}_j^a) - f(\mathbf{x}_j^n)\|^2$ and is set to $\alpha = 0.2$.

Since the number of triplets formed from a dataset is large, with many triplets contributing little or nothing to the loss, only a subset of triplet pairs that are hard to classify are needed to update the network in each iteration of the algorithm. An example of a hard triplet pair is shown in Figure S5.3. Because Sam Brownback, who was the 2014 Republican candidate for the gubernatorial election in Kansas, has similar facial features to Ronald Reagan, the anchor-negative distance $\|f(\mathbf{x}_i^a) - f(\mathbf{x}_i^n)\|$ is relatively small, while the anchor-positive distance $\|f(\mathbf{x}_i^a) - f(\mathbf{x}_i^p)\|$

may be too large due to significant facial misalignment in the positive image. By using triplets such as these in each step of training, the network adjusts its parameters so that two Reagan pictures are closer together while Sam Brownback is further away, effectively learning how to distinguish between faces.

S6 Description of Audio Features

This section provides a more formal description of the audio features we used, which were taken from (Ren, Wu and Jang, 2015). Let $S \in \mathbb{R}^{N \times K}$ denote the spectrogram for an audio signal, with $S_{i,k}$ corresponding to the strength of frequency component k in frame i of the spectrogram. Table S6.2 provides a summary of each audio feature.

S6.1 Short-term Features

S6.1.1 Statistical Spectrum Descriptor (SSD)

Define the frequency associated with bin k as $f_k = k \cdot F_s / N_{FFT}$, where $F_s = 22050$ and $N_{FFT} = 1024$. The statistical spectrum descriptor for frame i is a combination of the spectral centroid, skewness, kurtosis, flux, and rolloff, which are defined as

$$\text{centroid}_i = \frac{\sum_{k=1}^{K} f_k S_{i,k}}{\sum_{k=1}^{K} S_{i,k}}, \qquad \text{skewness}_i = \frac{m_3(S_i)}{\widehat{\sigma}^3_{S_i}}, \qquad \text{kurtosis}_i = \frac{m_4(S_i)}{\widehat{\sigma}^4_{S_i}}$$

and

$$\text{flux}_i = \|S_i - S_{i-1}\|^2, \qquad \text{rolloff}_i = f_{k_*}, \text{ such that } \sum_{k=1}^{k_*} S^2_{i,k} = 0.85 \sum_{k=1}^{K} S^2_{i,k}$$

where $m_3(S_i)$ and $m_4(S_i)$ are the sample third and fourth central moment of the spectrum, respectively, and $\widehat{\sigma}_{S_i}$ is the sample standard deviation.

S6.1.2 Mel-frequency Cepstral Coefficients (MFCC)

The exact computation of the MFCC is beyond the scope of the paper, instead we defer the reader to (Lee et al., 2009) for a more formal description and give a high level overview here. The MFCC computes a low-dimensional representation of the pooled log energies contained in a set of logarithmically-spaced subbands designed to mimic the human auditory system[3] using a Fourier-like transform called the discrete cosine transform. This procedure aims to give a low-dimensional representation of the spectral envelope of the spectrum. We retain the first 20 coefficients of the MFCC and use the energy contained in the entire frame to form the feature vector.

S6.1.3 Octave Spectral Contrast (OSC)

The OSC characterizes differences in the peaks and valleys of the spectrum across 8 logarithmically-spaced subbands. Formally, for each frequency band a, we sort the bins in the subband in order of increasing magnitude. Defining the number of bins in band a as N_a and $P^i_{a,k}$ as the magnitude for the kth bin in band a in frame i, we take the smallest and largest 20% bins and compute the

[3] For example, a human would perceive a 500Hz and 1000Hz tone to be as different as 2000Hz and 3000Hz tone. The perceptual spacing between tones is nonlinear on the Hz scale.

Type	Name	Dimension	Description
Short-term Features	SSD	10	Statistical descriptor for shape of the spectrum, which is determined by the timbral qualities of the audio signal.
	MFCC	42	Low-dimensional representation of the spectrum shape based on a nonlinear frequency scale modeling human perception.
	OSC	32	Measure of the variation between peaks and valleys in eight logarithmically-spaced frequency subbands. This feature is a characterization of how noise-like or tone-like the audio signal is.
	SFM/SCM	32	Measure of frequency dispersion in eight logarithmically-spaced frequency subbands. This feature is also a characterization of how noise-like or tone-like the audio signal is.
Long-term Features	Modulation Feature	112	Septal-based spectral contrast feature derived from modulation feature spectrograms for the MFCC. This feature characterizes perceptual audio qualities like tempo and rhythm, which occur on a longer time scale than timbre.
	Joint-frequency Feature	224	Spectral contrast feature derived from the joint-frequency spectrogram, which characterizes the short-term and long-term frequency variations. This feature also characterizes rhythm and tempo.

Table S6.2: Music features used for mood classification and sentiment analysis. The short-term features measure the timbral qualities of the audio on a short time scale, while the long-term features characterize perceptual qualities like rhythm and tempo, which occur on a longer time scale.

peak and valley as

$$Peak_i(a) = \log\left(\frac{1}{\lceil 0.2N_a \rceil} \sum_{k=1}^{\lceil 0.2N_a \rceil} P_{a,k}\right), \qquad Valley_i(a) = \log\left(\frac{1}{\lceil 0.2N_a \rceil} \sum_{k=1}^{\lceil 0.2N_a \rceil} P_{a,N_a-k+1}\right).$$

Taking the difference between the peaks and valleys yields the contrast for band a in frame i. The OSC feature matrix is formed by concatenating the contrast and the values together.

S6.1.4 Spectral Flatness Measure/Spectral Crest Measure (SFM/SCM)

SFM/SCM quantify how noise-like or how tone-like the audio signal is. Like the OSC, these quantities are computed in 8 logarithmically-spaced subbands. Using the same definitions in the OSC section, the SFM/SCM is computed as

$$SFM_i(a) = \frac{\sqrt[N_a]{\prod_{k=1}^{N_a} P_{a,k}}}{\frac{1}{N_a}\sum_{k=1}^{N_a} P_{a,k}}, \qquad SCM_i(a) = \frac{\max_{k=1,\ldots,N_a} P_{a,k}}{\frac{1}{N_a}\sum_{i=k}^{N_a} P_{a,k}}.$$

An SFM of 1 implies the spectrum has relatively equal magnitudes at all frequencies, which corresponds to audio that sounds like white noise, such as radio static. Low spectral flatness suggests that the energy in the spectrum is concentrated around only a few frequency bands, corresponding to an audio signal that sounds like a mixture of tones. An SCM of 1 corresponds to a flat, noise-like spectrum, while and SCM much larger than 1 indicates the spectrum is "spiky".

S6.2 Long-term Features

S6.2.1 Modulation Feature Spectrogram

The modulation feature spectrogram is used to compute the long-term features, which aim to capture information regarding the rhythm, tempo, and beat of an audio signal. They are constructed from the short-term feature matrices. Formally, given a short-term feature matrix $F \in \mathbb{R}^{N \times D}$ and defining the dth column F as F_d, we compute the modulation spectrogram via the following steps:

1. For each feature d, compute the spectrogram corresponding to the feature signal F_d using a segment length $W = 256$ and an overlap factor of 1/2. This step produces a $T \times 129$ matrix M_d, where T depends on the number of frames N in the original spectrogram.

2. Collapse the matrix M_d to a single vector $m_d \in \mathbb{R}^{129}$ by taking the average over all rows. m_d characterizes the frequency content of feature d.

3. Row stack the vectors m_d to form the modulation feature spectrogram $M_F \in \mathbb{R}^{D \times 129}$.

As was the case for the short-term feature, the long-term feature is a characterization of the modulation feature spectrogram, and e use the septal-based spectral contrast as th. For a modulation feature spectrogram M_F, we partition this matrix along the columns into 7 logarithmically spaced subbands. Defining the number of bins in band a as N_a and $P^d_{a,k}$ as the magnitude for the kth modulation frequency bin in band a for feature dimension d, the valleys and peaks in are computed as

$$MPeak(d,a) = \max_{k \in a} P^d_{a,k} \qquad MValley(d,a) = \min_{k \in a} P^d_{a,k},$$

We form the contrast by taking the difference between the peaks and valleys, all of which are $D \times 7$ matrices. We take the mean and standard deviation along the rows and columns of the valley and contrast matrices, separately, and stack these together to produce a feature $f \in \mathbb{R}^{4D \times 28}$. We independently apply this procedure to the MFCC feature matrix, the OSC feature matrix, and the SFM/SCM feature matrix and concatenate these all together to form the long-term feature $LT \in \mathbb{R}^{296}$.

S6.2.2 Joint-Frequency Feature

The averaging step in computing the modulation feature spectrogram throws out some information regarding the temporal evolution of the modulation features. The purpose of the joint-frequency feature is to recover and characterize this lost information. After obtaining the spectrogram $S \in \mathbb{R}^{N \times 513}$ described at the beginning of Section 3.2.3, another one-sided FFT is performed along each column of S, followed by an absolute value operation, yielding a matrix $J \in \mathbb{R}^{N/2+1 \times 513}$. This matrix is then partitioned into a 7×8 grid using the same logarithmically-spaced subbands as the octal- and septal-based spectral contrast features described above. Each block of the grid is vectorized, a we compute the spectral contrast, spectral valley, spectral flatness measure, and spectral crest measure as defined in the short-term feature section. This process produces four matrices, each of dimension 7×8. We vectorize and stack the matrices together to form the joint-frequency feature $JF \in \mathbb{R}^{224}$.

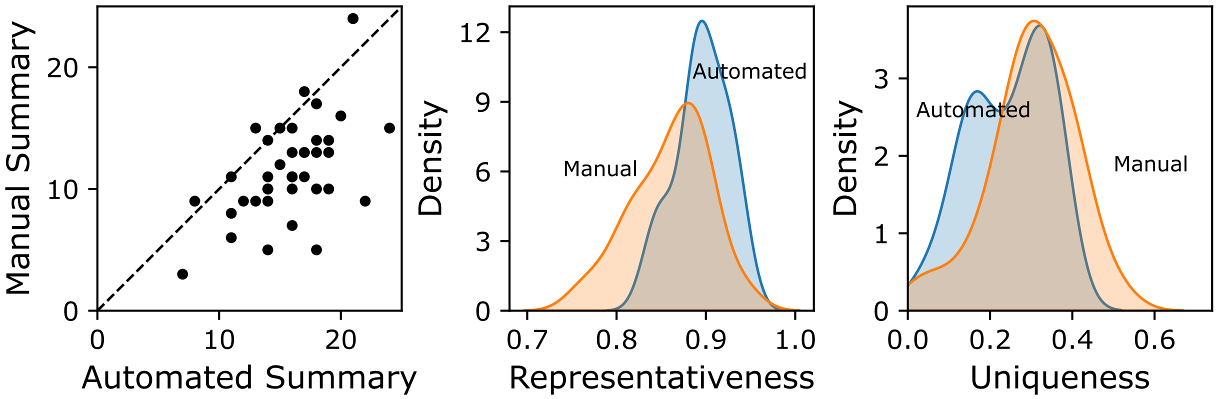


Figure S7.4: Results of the Validation Study for the Video Summarization Algorithm. The left plot shows the comparison between the number of frames selected into the summary for the auto- and manually-generated summaries. The middle and right plots show the comparison between the distributions of representativeness and uniqueness over automated (blue) and manual (orange) summaries, respectively.

S7 Video Summarization Validation Study

The results of our validation study are summarized in Figure S7.4. The left-most plot shows a comparison of the number of frames selected between the auto- and manually-generated summaries. Recall that the uniqueness quantifies how distinct frames are from one another in the summary, while the representativeness enforces the rule that for each frame in the original video, there should exist at least one frame in the summary that is visually similar. We see that for all videos, the summarization algorithm selected the same or more number of frame as the manual summary. Since the original video files contain roughly 720–960 frames, both summaries resulted in a roughly 97%–99% reduction in the number of frames. Furthermore, computation of the summary took approximately 20–40 seconds for each video. These results suggest summarization may yield significant efficiency gains over methods which analyze all frames in the video.

In the middle plot we show the distribution of the normalized summary representativeness. We find that the auto-generated summaries tend to be slightly more representative than the manually-generated summaries, suggesting the algorithm does well in representing the content of the video. The right-most plot shows the distribution of the normalized uniqueness measure. A high value of this statistic suggests there are few duplicates in the summary. We find that the auto-generated summaries tend to be less unique than the manually-generated summaries. This is not surprising since the algorithm tended to produce a greater number of frames than manual summarization.

Finally, visual inspection of auto-generated and manual summaries indicates that the algorithm produced summaries that contain at least one image for each shot in video in 75% of the sample. In all instances where an image is missing, the missing image either contains no meaningful information pertaining to the WMP variables examined in this study, or there is another image in summary that carries the same visual information. For example, if a candidate appeared multiple times in an ad in multiple, near-identical shots, the algorithm would select a representative frame in only some of these shots. Such issues, which were the most common cause of missingness in our study, would not lead to errors further downstream in our video processing pipeline. We emphasize that although, as shown below, video summarization does not appear to degrade our classification performance, it may have resulted in the loss of other important information.

S8 WMP Variable Encoding Instructions

Table S8.3 shows the instructions given to the human coders for the variables automated in our work.

Variable	Prompt	Choices	Note
Issue Mention (Political figures)	Are any of the following mentioned or pictured in the ad?	0 = No 1 = Yes, in a way to show approval or support 2 = Yes, in a way to show disapproval or opposition 3 = Yes, unclear whether in support or opposition	We treat all "Yes" answers as 1.
Issue Mention (Words/phrases)	Are any of these words or phrases *specifically* mentioned in the ad?	0 = No 1 = Yes	We exclude *Working Class*, *Middle Class*, and *Upper Class / Rich / Wealthy* from our analysis
Issue Mention (General)	Are any of the following issues mentioned in this ad?	0 = No 1 = Yes	We merge *abortion* and *women's health* into a single issue
Opponent Mention	Excluding the *oral approval*, is the opposing candidate mentioned by name in the ad?	0 = No 1 = Yes	
Candidate Picture	Excluding the *oral approval*, is the favored / opposing candidate pictured in the ad?	0 = No 1 = Yes	
Candidate Picture (Oral Approval)	Does the candidate physically appear on screen and speak to the audience during oral approval?	0 = No 1 = Yes	Static images accompanied by a voice-over do not count
Music Mood	If music is played during the ad, how would it best be described?	0 = No 1 = Yes	Asked for each of "ominous/tense", "uplifting", "sad/sorrowful"

Variable	Prompt	Choices	Note
Ad tone (WMP)	In your judgment, is the primary purpose of the ad to promote a specific candidate, attack a candidate, or contrast the candidates?	1 = Contrast 2 = Promote 3 = Attack	We ignore ads marked with “Contrast”

Table S8.3: WMP Encoding Instructions for Automated Variables.

S9 Keywords Used for Issue Detection

Among a total of 83 issues, we use the WMP issue names and the last names of political actors for 44 issues (e.g., “tax” for the *Tax* issue, and “Pelosi” as the *Pelosi* issue). For 16 issues, we also added synonyms and some words that share the same roots. For example, we add “Chinese” as another keyword for the *China* issue, and include the word root “agricult,” in addition to “farm” as a keyword for the “farming” issue. Third, for 21 issues, we have added relevant common expressions. For example, we include “climate change” as an additional keyword for the *global warming* issue. Another example is the addition of “second amendment,” “2nd amendment,” “NRA,” and “bear arms” for the *gun control* issue. Finally, for the *jobs/employment* and *abortion/women’s health* issues, we use a more extensive list of relevant words and phrases. Table S9.4 shows the list of keywords associated with each topic arranged in the order they are defined by the WMP and grouped together in broad categories.

Issue/Figure	Keywords	Frequency in WMP
Barack Obama	obama|the president|our president	603
George W. Bush	bush	21
Ronald Reagan	reagan	5
John Boehner	boehner	7
Nancy Pelosi	pelosi	52
Mitch McConnell	mcconnell	26
Harry Reid	reid	14
Congress	congress NOT in congress|for congress|to congress †	419
Democrats	democrats|and democrat|or democrat	229
Republicans	republicans|and republican|or republican|GOP	270
Tea Party	tea party	43
God	god NOT thank god †	8
Hope	hope NOT i hope|we hope †	21
Change	change	121
Experience	experience	50

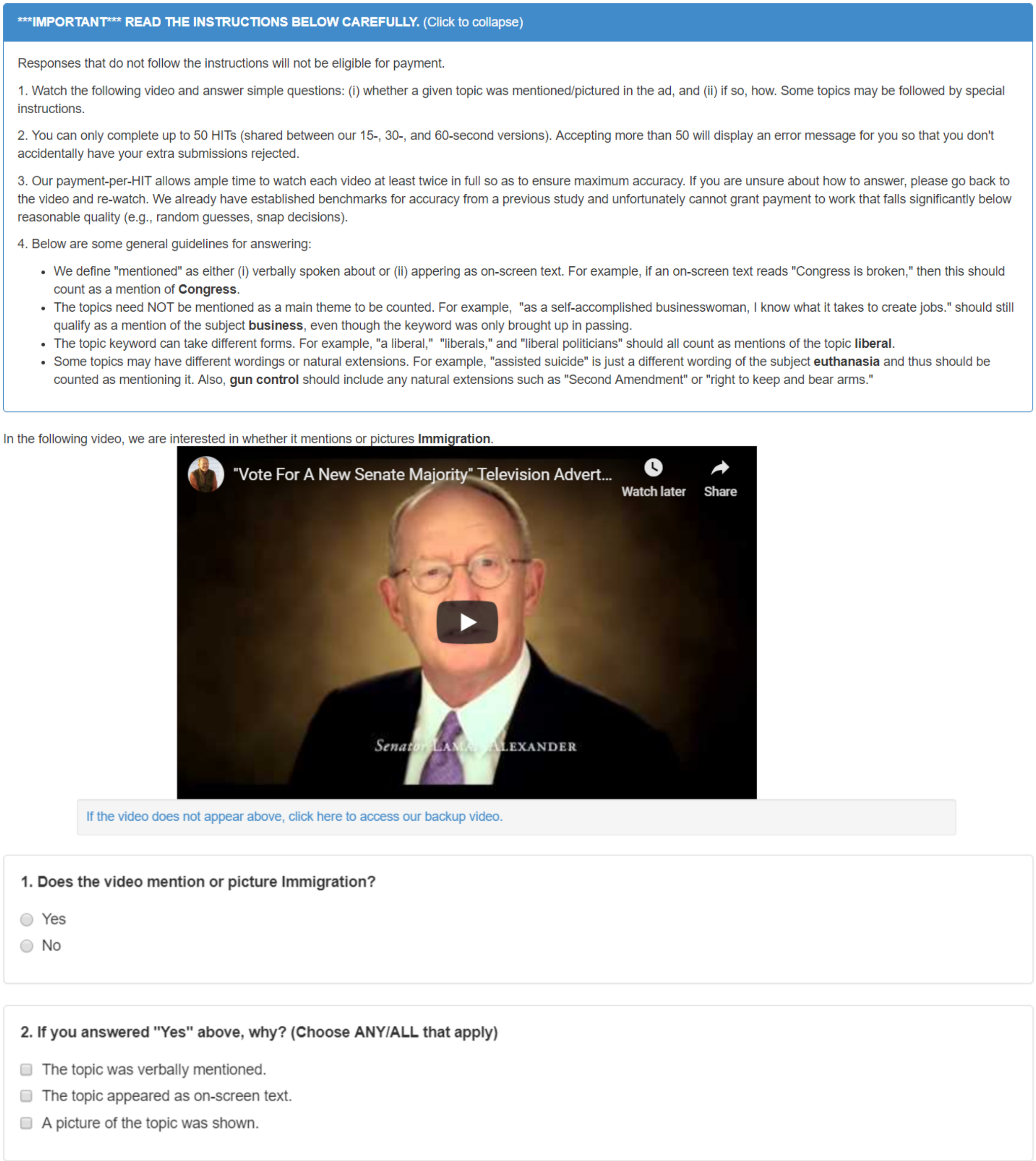
Issue/Figure	Keywords	Frequency in WMP
Liberal	liberal	76
Conservative	conservative	163
Special Interest	special interest	71
Negative Campaigning	negative campaign	18
Main Street	main street	6
Wall Street	wall street NOT wall street journal †	87
Big Government	big government	23
Tax	tax	769
Deficit/Budget/Debt	deficit|budget|debt	482
Government Spending	spending	309
Recession/Stimulus/Bailout	recession|stimulus|bailout	86
Minimum Wage	minimum wage	51
Farming	farm|agricult	62
Business	business NOT businessman †	351
Union	union	11
Jobs/Employment	jobs|outsourc|employment|unemploy| out of job|back to work + many forms of [create job], [lose job], [kill job]	1105
Poverty	poverty	14
Trade/Globalization	trade|globaliz|NAFTA NOT cap and trade †	113
Housing/Subprime Mortgage	housing|subprime|sub prime	38
Economy (Generic)	economy|economic prosperity	273
Inequality	unequal|inequal|equal pay	53
Abortion/Women's Health	abortion|pregnan|woman's right to choose| women's right to choose|reproductive right| pro choice|pro life|woman's health| women's health|birth control|contracept| planned parenthood + various forms of [Roe v. Wade]	213
Homosexuality/Gay/Lesbian	lgbt|gay|lesbian|transgender|same sex| marriage equality|traditional marriage| one man and one woman	16
Moral/Family/Religious Values	honesty|integrity|moral|family value	100
Tobacco	tobacco|cigarette	0
Affirmative Action	affirmative action	2
Gambling	gambling	0
Assisted Suicide/Euthanasia	euthanasia|assisted suicide	0
Gun Control	gun|second amendment|2nd amendment|	63

Issue/Figure	Keywords	Frequency in WMP
Civil Liberty/Privacy	bear arms|NRA civil libert|freedom of speech|free speech| freedom of religion|freedom of faith|privacy	12
Race Relations/Civil Rights	civil right	6
Crime	crime|criminals|violence|victim|predator	47
Narcotics/Drugs	drugs|drug addict|drug deal|drug lord|narcotic| marijuana|cocaine|heroin|opioid|opiate NOT prescription †	11
Capital Punishment/Death Penalty	death penalty|capital punishment	2
Supreme Court/Judiciary	judicia|supreme court|courts	4
Education/Schools	educat|schools|tuition|affordable college| college affordable|college more affordable	445
Lottery for Education	lottery for education|education lottery	4
Child Care	childcare|child care|daycare|day care| care for your child|care for our child| care for your kid|care for our kid	17
Healthcare	healthcare|health care|health insurance| medical insurance|obamacare|affordable care	428
Prescription Drug	prescription drug	22
Medicare	medicare	417
Social Security	social security	254
Welfare	welfare	26
Military	military|troops|armed force|servicemember| service member|in uniform|war in|wars in	145
Foreign Policy	foreign policy	19
Veterans	veteran|vet|VA|world war|vietnam war	244
Foreign Aid	foreign aid	4
Nuclear Proliferation	nuclear proliferation|nuclear weapon	3
China	china|chinese	68
Middle East	middle east	16
Afghanistan	afghan	26
September Eleven	911|september eleven|nine eleven|september 11	7
Terror/Terrorism/Terrorist	terror|war on terror	35
Iraq	iraq	54
Israel	israel	2
Iran	iran	4
Environment	environ|EPA|cap and trade	32
Global Warming	global warming|climate change	12
Energy	energy|power plant|keystone|pipeline|coal| petroleum|natural gas|solar|fracking	195

Issue/Figure	Keywords	Frequency in WMP
BP Oil Spill	oil spill	4
Campaign Finance Reform	campaign finance|citizens united	13
Government Ethics/Scandal	corrupt|government ethics|government scandal	144
Corporate Fraud	corporate fraud	22
Term Limit	term limit	6
Pledge of Allegiance	pledge of allegiance|pledge allegiance	0
Immigration	immigra|alien|border|dream act|amnesty	66
Local Issues	local	186
Government Regulation	regulation	118

Table S9.4: Comprehensive List of Keywords Used for Automated Detection of Issues/Figures. † Exclusion rules are applied.

S10 Scripts for Amazon Mechanical Turk Tasks



*****IMPORTANT*** READ THE INSTRUCTIONS BELOW CAREFULLY.** (Click to collapse)

Responses that do not follow the instructions will not be eligible for payment.

1. Watch the following video and answer simple questions: (i) whether a given topic was mentioned/pictured in the ad, and (ii) if so, how. Some topics may be followed by special instructions.

2. You can only complete up to 50 HITs (shared between our 15-, 30-, and 60-second versions). Accepting more than 50 will display an error message for you so that you don't accidentally have your extra submissions rejected.

3. Our payment-per-HIT allows ample time to watch each video at least twice in full so as to ensure maximum accuracy. If you are unsure about how to answer, please go back to the video and re-watch. We already have established benchmarks for accuracy from a previous study and unfortunately cannot grant payment to work that falls significantly below reasonable quality (e.g., random guesses, snap decisions).

4. Below are some general guidelines for answering:

- We define "mentioned" as either (i) verbally spoken about or (ii) appering as on-screen text. For example, if an on-screen text reads "Congress is broken," then this should count as a mention of **Congress**.
- The topics need NOT be mentioned as a main theme to be counted. For example, "as a self-accomplished businesswoman, I know what it takes to create jobs." should still qualify as a mention of the subject **business**, even though the keyword was only brought up in passing.
- The topic keyword can take different forms. For example, "a liberal," "liberals," and "liberal politicians" should all count as mentions of the topic **liberal**.
- Some topics may have different wordings or natural extensions. For example, "assisted suicide" is just a different wording of the subject **euthanasia** and thus should be counted as mentioning it. Also, **gun control** should include any natural extensions such as "Second Amendment" or "right to keep and bear arms."

In the following video, we are interested in whether it mentions or pictures **Immigration**.

If the video does not appear above, click here to access our backup video.

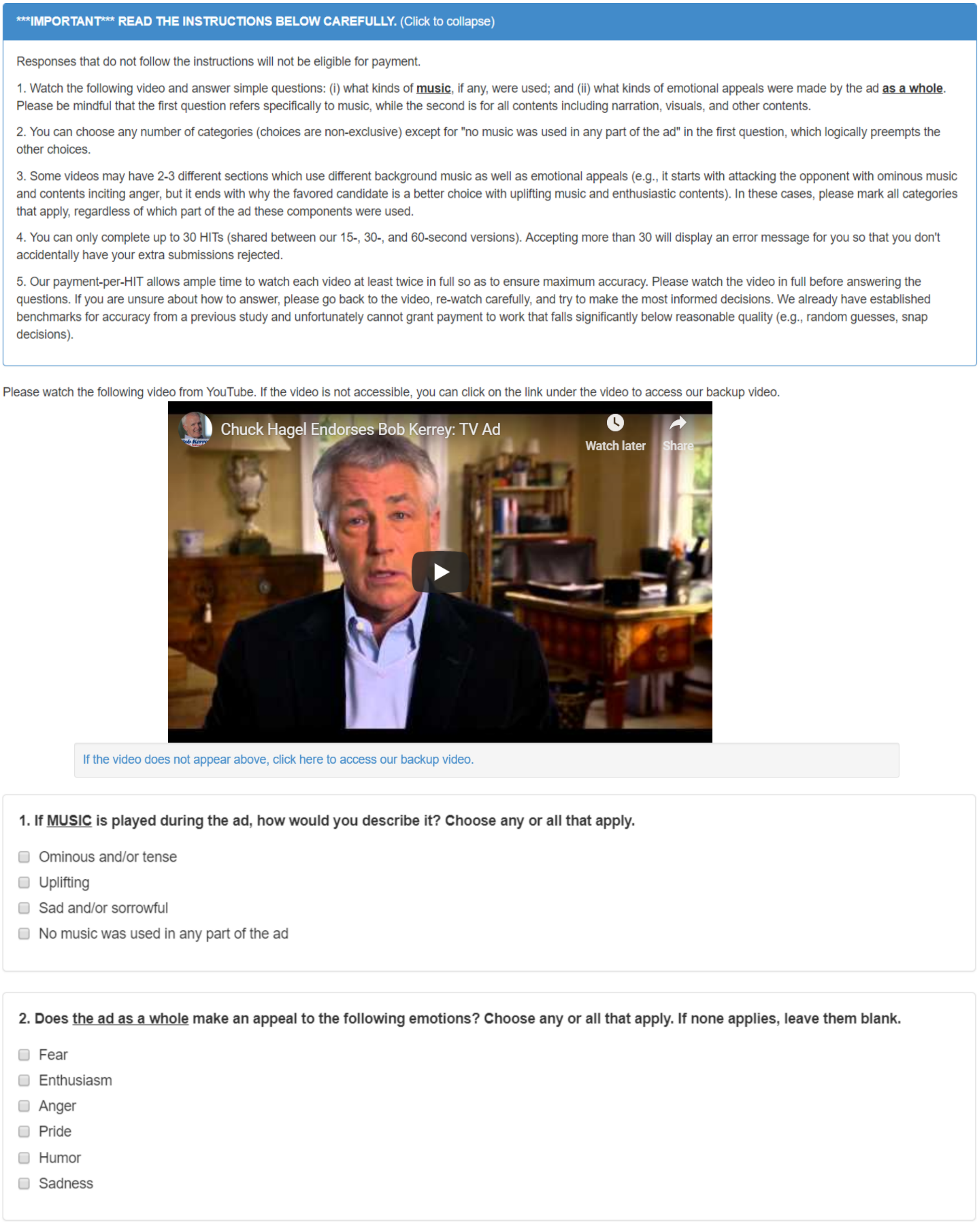
1. Does the video mention or picture Immigration?

○ Yes
○ No

2. If you answered "Yes" above, why? (Choose ANY/ALL that apply)

□ The topic was verbally mentioned.
□ The topic appeared as on-screen text.
□ A picture of the topic was shown.

Figure S10.5: An Example Script for Issue Detection Task as Seen by Amazon MTurkers.



*****IMPORTANT*** READ THE INSTRUCTIONS BELOW CAREFULLY.** (Click to collapse)

Responses that do not follow the instructions will not be eligible for payment.

1. Watch the following video and answer simple questions: (i) what kinds of **music**, if any, were used; and (ii) what kinds of emotional appeals were made by the ad **as a whole**. Please be mindful that the first question refers specifically to music, while the second is for all contents including narration, visuals, and other contents.

2. You can choose any number of categories (choices are non-exclusive) except for "no music was used in any part of the ad" in the first question, which logically preempts the other choices.

3. Some videos may have 2-3 different sections which use different background music as well as emotional appeals (e.g., it starts with attacking the opponent with ominous music and contents inciting anger, but it ends with why the favored candidate is a better choice with uplifting music and enthusiastic contents). In these cases, please mark all categories that apply, regardless of which part of the ad these components were used.

4. You can only complete up to 30 HITs (shared between our 15-, 30-, and 60-second versions). Accepting more than 30 will display an error message for you so that you don't accidentally have your extra submissions rejected.

5. Our payment-per-HIT allows ample time to watch each video at least twice in full so as to ensure maximum accuracy. Please watch the video in full before answering the questions. If you are unsure about how to answer, please go back to the video, re-watch carefully, and try to make the most informed decisions. We already have established benchmarks for accuracy from a previous study and unfortunately cannot grant payment to work that falls significantly below reasonable quality (e.g., random guesses, snap decisions).

Please watch the following video from YouTube. If the video is not accessible, you can click on the link under the video to access our backup video.

If the video does not appear above, click here to access our backup video.

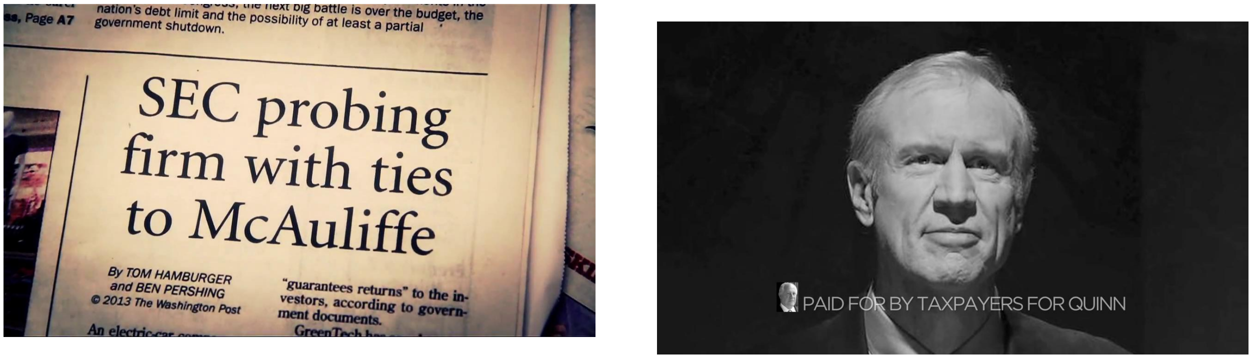
1. If MUSIC is played during the ad, how would you describe it? Choose any or all that apply.

□ Ominous and/or tense
□ Uplifting
□ Sad and/or sorrowful
□ No music was used in any part of the ad

2. Does the ad as a whole make an appeal to the following emotions? Choose any or all that apply. If none applies, leave them blank.

□ Fear
□ Enthusiasm
□ Anger
□ Pride
□ Humor
□ Sadness

Figure S10.6: An Example Script for Musical Mood Detection Task as Seen by Amazon MTurkers. The script also includes a second question on the emotional appeal of the ad, which was not used in the paper.


nation's debt limit and the possibility of at least a partial government shutdown.
Page A7
SEC probing firm with ties to McAuliffe
By TOM HAMBURGER and BEN PERSHING © 2013 The Washington Post
"guarantees returns" to the investors, according to government documents.
PAID FOR BY TAXPAYERS FOR QUINN

(a) Campaign ad by Republican candidate Ken Cuccinelli for the 2014 gubernatorial election in Virginia. The automated coding algorithm detected some of the smaller text from the newspaper clipping, incorrectly choosing the *deficit/budget/debt* issue.

(b) Campaign ad by Democratic candidate Pat Quinn for the 2014 gubernatorial election in Illinois. The automated coding algorithm detected the word "Taxpayer" from the disclaimer showing the campaign ad sponsor, incorrectly choosing the *tax* issue.

Figure S11.7: Examples Illustrating the Mistake of the Automated Coding for the Issue Mention Variable.

S11 Issue Mention Validation Results

We examine the conditions under which the automated procedure makes mistakes in coding the issue mention variable. To do this, we select a random subset of 120 issue-video pairs from the 300 disagreement cases used for the MTurk study (70 false positives and 50 false negatives) and carefully watch these videos to correct any mistakes for these pairs. We note that many of the samples in this analysis correspond to most difficult cases in which the "correct" coding is unclear due to ambiguity in WMP's instructions, and thus our corrected label set may contain bias.

We find that the automated coding made 22 mistakes out of 70 false positive evaluations, implying that the automated coding greatly outperformed the WMP coding in these cases. The most frequent mistake of our automated coding procedure is due to the fact that it ignores the context. For example, the use of the keyword "trade" in the context of the World Trade Center was considered as mentioning *trade/globalization.* The other reason for mistakes is our automated procedure detects irrelevant text in the background of ads. Figure S11.7a shows an example in which the automated coding algorithm detected on-screen text from a newspaper clipping, leading to a false positive declaration. The automated coding also incorrectly treated the names of institutions as mentioning certain issues. One such example is given in Figure S11.7b, where the coding algorithm detected the keyword "tax" in the name of an organization, "Taxpayers for Quinn," who sponsored the campaign ad.

We have also examined 50 false negative cases and found that the automated coding resulted in 31 mistakes, suggesting that the WMP had better performance in these cases. The most frequent reason for mistakes is missing keywords, which is relatively easy to correct by simply including an additional set of keywords. Examples include "college debt" and "teachers" for the *education* issue and "foreign competition"for the *trade/globalization* issue. The second most frequent reason is an indirect mention of issues. For example, an ad included the phrase "opposed buy American provisions for military weapon systems", as an indirect mention of *trade/globalization* issue; another ad used the phrase "asking the wealthy to pay more" in the context of the *tax* issue. These cases represent a challenge to keyword-based approaches such as ours, since accounting for all such variations in indirect references is impossible for most practical purposes.

S12 Opponent Mention Validation Results

Like in the previous section, we carefully watch all 90 videos in the disagreement conditions and correct any mistakes made by WMP coders. In our labeling procedure, we treat all on-screen text of the opponent's name as an instance of an opponent mention, though we note that WMP provides no guidelines on how this should be handled. Out of 67 false positives (i.e., the automated coding gives Yes while the WMP gives No), the automated coding makes only three mistakes, all of which were due to ignoring context. Two instances occurred in an election where both the opponent and candidate shared the same last name, and the final instance referenced the opponent during the approval segment, which violates the definition of the variable. For the 23 videos in the false negative condition (i.e., the automated coding gives No while the WMP gives Yes), the automated coding makes 16 mistakes.

The most frequent reason (13 out of 16) why the automated coding fails to detect the mention of opponents is the mis-transcription of the last names of the opposing candidates, especially when they are relatively uncommon. Examples include mishearing "Tisei" as "to say," "Lankford" as "Langford," and "Critz" to "Crits." Among the three remaining errors, one is because the opponent was mentioned by first name only, another because a sample labeled as an ad for a general election was actually for the primary election, so the wrong opponent name was used in our method, and the last one due to "Obamacare" being treated as a reference to the opponent, Barack Obama. Altogether, in cases of disagreement, our method is correct 71 out of 90 (79%) cases, which demonstrates that the performance of automated coding of the opponent mention variable exceeds that of WMP human coding.

S13 Face Recognition Validation Results

We first plot the ROC curves in Figure S13.8. The left plot shows the ROC curve using the original variable for opponent candidate appearance and the combined variable for favored candidate appearances, both of which exclude appearances during the oral approval segment. The right plot shows the ROC curve using recoded variables in which we corrected mistakes in the WMP coding and dropped the restriction that candidates appear outside the oral approval segment. The results in the left plot show that face recognition performs well on opponents and poorly on favored candidates. After recoding the variables, the performance greatly improves, suggesting that face recognition is very accurate in this application.

Next, we watch all videos corresponding to the 216 disagreement cases (148 for the favored candidates and 68 for the opponents) to identify the reasons why the WMP coders and the face recognition algorithm differ. For the cases of favored candidates, the primary reason for disagreement (97 cases or 67%) is that the algorithm has detected the images of favored candidates within the oral approval segment. This is expected as we did not add any filter regarding the restriction made by the WMP. The second most frequent reason (33 cases or 22%) is that the algorithm had a difficult time dealing with angled or occluded faces, or the quality of the video was poor. The remaining 18 disagreements (12%) are due to mislabeling by the WMP coders. For the opposing

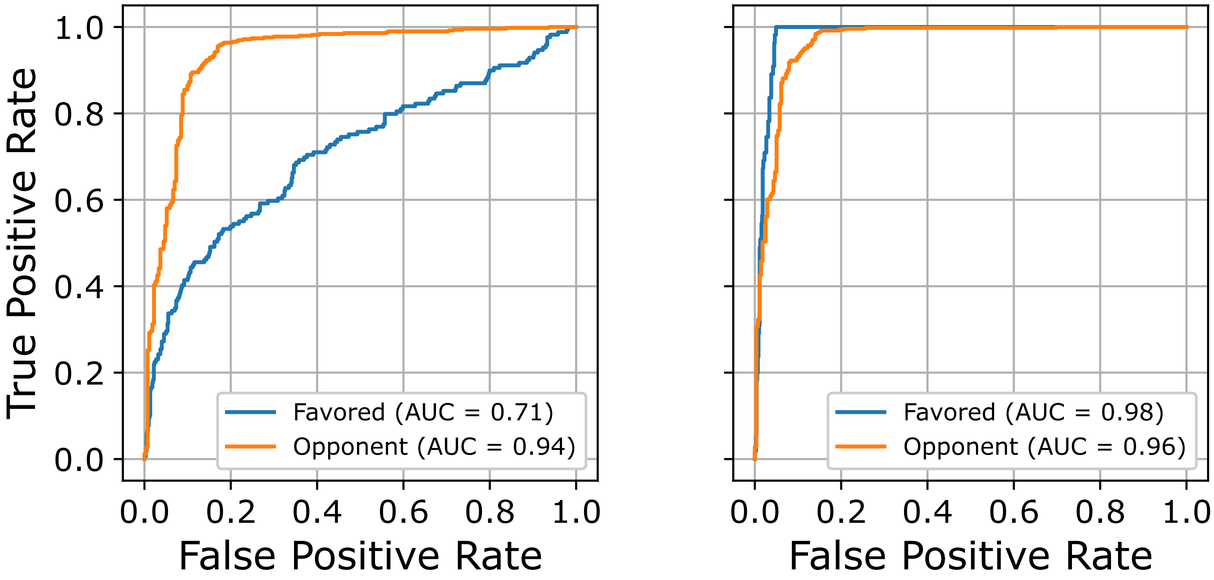


Figure S13.8: ROC Curves for the Face Recognition Algorithm. The left plot shows the face recognition performance using the original WMP data for the favored and opposing candidate mentions. The right plot shows performance using the corrected WMP variables.

candidates, 47 disagreements (69%) are due to image or face quality. The remaining 21 cases (31%) are due to mislabeling on the part of the WMP.

Finally, we evaluate the performance of the face recognition algorithm by removing the restriction that the favored candidates appear in the main segment of an ad alone. To do this, we manually recode the WMP variables for the disagreement cases so that the variable represents whether the favored candidate appears in any segment. We also correct the labeling mistakes made by the WMP coders. Note that this procedure assumes the agreements between the manual and automated codings indicate the accurate classification, so the results of this exercise will lead to inflated results in our favor, since we benefit from making the same mistakes as WMP. We compute the precision, recall, and accuracy of the face recognition algorithm using the corrected data and find that they are 0.99, 0.96, and 0.96 for the favored candidates and 0.97, 0.86, and 0.94 for the opposing candidates, respectively. These numbers represent an impressive performance of the face recognition algorithm.

S14 Additional Validation Results

S14.1 MTurk Analysis for Music Mood Variables

		Majority Opinion among MTurkers					
		Ominous/Tense		Uplifting		Sad/Sorrowful	
		No	Yes	No	Yes	No	Yes
WMP Coding	No	60.89%	7.33%	24.44%	4.67%	78.22%	7.11%
	Yes	7.78%	24.00%	9.78%	61.11%	7.11%	7.56%

Table S14.5: Comparison of the Musical Mood Variables between the WMP and MTurker Codings. MTurk coder responses are transformed to a binary variable based on the majority opinion. The value in each cell corresponds to the proportion of the four different combinations of results from the WMP and automated coding schemes. The three two-by-two matrices correspond to the three different moods of music used in the WMP data. The results shown here are from the test data set of size 450.

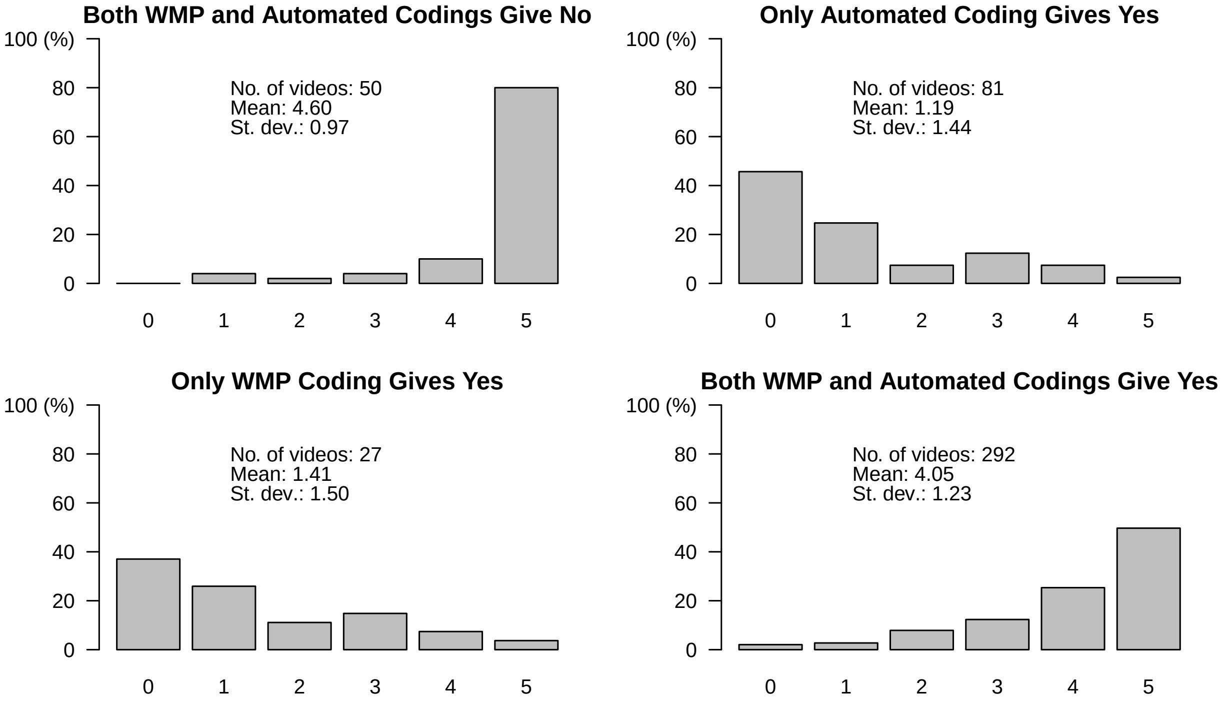


Figure S14.9: Number of MTurkers Who are in Agreement with Automated Coding For Uplifting Music. Four cases are based on the agreement between the WMP and automated codings. The total number of MTurkers for each task is five. The texts within each plot show the number of videos included in each sample as well as the mean and standard deviation of the number of MTurkers in agreement with the automated coding.

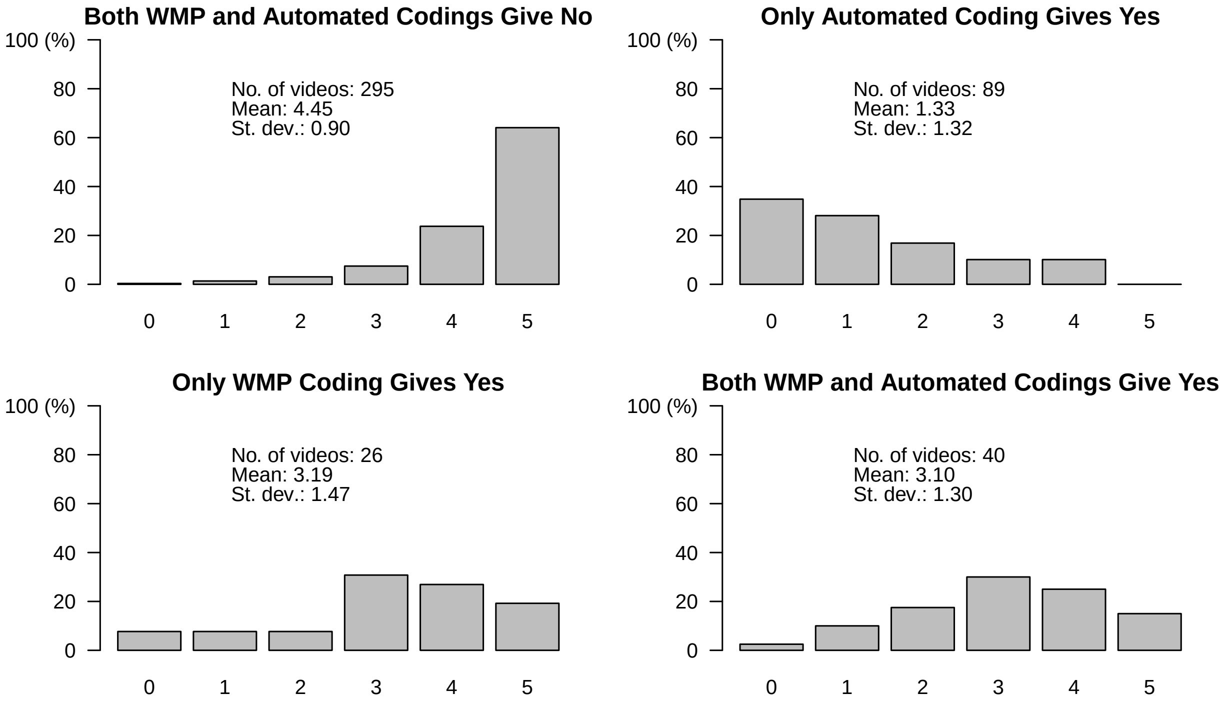


Figure S14.10: Number of MTurkers Who are in Agreement with Automated Coding For Sad/Sorrowful Music. See the caption of Figure S14.9 for details.

S14.2 Classification Performance for Ad Negativity Variable

				Linear SVM			
		Text Only		Music Only		Text and Music	
		No	Yes	No	Yes	No	Yes
WMP Coding	No	30.09%	10.66%	25.12%	15.64%	33.65%	7.11%
	Yes	5.70%	53.55%	8.29%	50.95%	7.82%	51.42%

				KNN			
		Text Only		Music Only		Text and Music	
		No	Yes	No	Yes	No	Yes
WMP Coding	No	21.33%	19.43%	17.76%	22.99%	14.69%	26.07%
	Yes	1.18%	58.06%	7.35%	51.90%	5.45%	53.79%

				Random Forest			
		Text Only		Music Only		Text and Music	
		No	Yes	No	Yes	No	Yes
WMP Coding	No	28.67%	12.09%	22.04%	18.72%	22.51%	18.25%
	Yes	5.21%	54.03%	9.00%	50.24%	5.21%	54.03%

				Naive Bayes			
		Text Only		Music Only		Text and Music	
		No	Yes	No	Yes	No	Yes
WMP Coding	No	31.28%	9.48%	31.75%	9.00%	31.52%	9.24%
	Yes	6.16%	53.08%	31.05%	28.20%	15.40%	43.84%

Table S14.6: Comparison of the Ad Negativity Variable between the WMP and Automated Codings Using Linear SVM, KNN, Random Forest, and Naive Bayes. The value in each cell corresponds to the proportion of the four different combinations of results from the WMP and automated coding schemes. The three two-by-two matrices in each row correspond to the three types of input data used to train the models. The results shown here are from the test data set of size 422.

S14.3 Ad Negativity Classification using LSD

		Automated Coding					
		LSD (Full)		LSD (Test)		Ours (Test)	
		No	Yes	No	Yes	No	Yes
WMP Coding	No	15.57%	24.17%	15.64%	25.12%	31.99%	8.77%
	Yes	9.74%	50.52%	11.14%	48.10%	6.87%	52.37%

Table S14.7: Comparison of the Ad Negativity Variable between the LSD and Automated Codings Using the Non-linear SVM. The value in each cell corresponds to the proportion of the four different combinations of results from the WMP and automated coding schemes. Full sample size is 2106 and test sample size is 422.

S14.4 Impact on Downstream Analysis

	Dependent variable:	
	Issue Convergence	
	WMP Coding	Automated Coding
Competitiveness	5.125	2.700
	(5.425)	(5.785)
Total Spending/VEP	−0.447	−0.255
	(0.464)	(0.465)
Diff. in Spending/VEP	−0.182	0.034
	(0.820)	(0.801)
Negative Ads	0.199*	0.222**
	(0.111)	(0.109)
VAP (logged)	−8.650**	−8.224**
	(3.419)	(3.332)
Year 2012	10.009**	6.884
	(4.482)	(4.348)
Consensual	12.704**	32.157***
	(6.440)	(7.974)
Owned	11.323**	10.663**
	(4.547)	(4.780)
Salience	0.662**	0.474*
	(0.260)	(0.283)
Constant	123.657**	120.178**
	(49.728)	(49.186)
Observations	292	290
Note:		*p<0.1; **p<0.05; ***p<0.01

Table S14.8: Application of Methods Used in Kaplan, Park and Ridout (2006) for 2012 & 2014 Using WMP and Automated Coding. The base data sets are prepared by either using the WMP-provided issue mention variables or replacing them with automated ones. Although standard errors tend to be slightly larger with the automatically coded data, the two analyses produce substantively similar results.

The analysis of this appendix examines the downstream impact of automatic coding on regression analysis. We use an existing analysis in political science literature (Kaplan, Park and Ridout, 2006). Using the WMP manually coded data, the authors produced a measure of issue convergence between Republican and Democratic campaign TV advertisements and employed a variety of campaign- and issue-specific independent variables to explain potential causes of the said measure (see the original article for details). The independent variables used for each random effects model were modeled after the original study, including which issues are defined as owned or consensual. The issue salience measure was only available for 2012 and had to be reused for 2014. Here, we simply compare the results of the same regression analysis between the automatically and manually coded data sets. The table indicates that although standard errors tend to be slightly larger with the automatically coded data, the two analyses produce substantively similar results.

References

Chakraborty, Shayok, Omesh Tickoo and Ravi Iyer. 2015. Adaptive keyframe selection for video summarization. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on.* IEEE pp. 702–709.

Dalal, Navneet and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05).* Vol. 1 IEEE Computer Society pp. 886–893.

Haitsma, Jaap and Ton Kalker. 2002. A highly robust audio fingerprinting system. In *Ismir.* Vol. 2002 pp. 107–115.

Harris, Fredric J. 1978. "On the use of windows for harmonic analysis with the discrete Fourier transform." *Proceedings of the IEEE* 66:51–83.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* pp. 770–778.

Kaplan, Noah, David K Park and Travis N. Ridout. 2006. "Dialogue in American Political Campaigns? An Examination of Issue Convergence in Candidate Television Advertising." *American Journal of Political Science* 50:724–736.

Kazemi, Vahid and Josephine Sullivan. 2014. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* pp. 1867–1874.

King, Davis E. 2015. "Max-margin object detection." *arXiv preprint arXiv:1502.00046.*

Lee, Chang-Hsing, Jau-Ling Shih, Kun-Ming Yu and Hwai-San Lin. 2009. "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features." *IEEE Transactions on Multimedia* 11:670–682.

Ng, Hong-Wei and Stefan Winkler. 2014. A data-driven approach to cleaning large face datasets. In *2014 IEEE international conference on image processing (ICIP).* IEEE pp. 343–347.

Oppenheim, Alan V, Alan S Willsky and S Nawab. 1996. *Signals and Systems (Prentice-Hall signal processing series).* Prentice Hall Englewood Cliffs, New Jersey.

Parkhi, Omkar M, Andrea Vedaldi and Andrew Zisserman. 2015. "Deep face recognition.".

Ren, Jia-Min, Ming-Ju Wu and Jyh-Shing Roger Jang. 2015. "Automatic music mood classification based on timbre and modulation features." *IEEE Transactions on Affective Computing* 6:236–246.

Sagonas, Christos, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou and Maja Pantic. 2016. "300 faces in-the-wild challenge: Database and results." *Image and vision computing* 47:3–18.

Yang, Shuo, Ping Luo, Chen-Change Loy and Xiaoou Tang. 2016. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* pp. 5525–5533.