Aligning Movies with Scripts by Exploiting Temporal Ordering Constraints

Iftekhar Naim, Abdullah Al Mamun, Young Chol Song, Jiebo Luo, Henry Kautz, Daniel Gildea
Department of Computer Science, University of Rochester, Rochester, NY

***Abstract*—Scripts provide rich textual annotation of movies, including dialogs, character names, and other situational descriptions. Exploiting such rich annotations requires aligning the sentences in the scripts with the corresponding video frames. Previous work on aligning movies with scripts predominantly relies on time-aligned closed-captions or subtitles, which are not always available. In this paper, we focus on automatically aligning faces in movies with their corresponding character names in scripts without requiring closed-captions/subtitles. We utilize the intuition that faces in a movie generally appear in the same sequential order as their names are mentioned in the script. We first apply standard techniques for face detection and tracking, and cluster similar face tracks together. Next, we apply a generative Hidden Markov Model (HMM) and a discriminative Latent Conditional Random Field (LCRF) to align the clusters of face tracks with the corresponding character names. Our alignment models (especially LCRF) significantly outperform the previous state-of-the-art on two different movie datasets and for a wide range of face clustering algorithms.**

I. Introduction

Movie scripts provide rich textual descriptions of movie scenes and can be easily downloaded from the internet [1]. However, in order to exploit such a rich source of annotation, we need to know the alignment between the sentences in the scripts and their corresponding video frames. Aligning movies with scripts can be useful for movie search and retrieval [2], rapid scene browsing [3], and movie summarization [4]. Furthermore, it can help us acquire large-scale weakly-annotated video datasets for training supervised and semi-supervised computer vision models [5]. Manually segmenting movie scenes and aligning them with scripts is tedious, especially for large collections of movies, and therefore the need for automated alignment methods has become crucial. Existing methods for movie-to-script alignment predominantly rely on closed-captions or subtitles with precise time-stamps [1], [5], [3], [6]. Since there exists a strong correspondence between the scripts and the subtitles (as they both include the dialogs), aligning in the presence of closed-captions/subtitles is a significantly easier task, and has been shown to achieve high alignment accuracy. However, closed-captions/subtitles are not always available, especially for foreign language movies and TV shows [7]. In this paper, we focus on aligning movies with scripts in the absence of closed-captions/subtitles.

Existing methods for subtitle-free movie-to-script alignment usually apply unsupervised clustering techniques to automatically group similar face tracks in the movies, and match face clusters with character names by exploiting the correlation in their co-occurrence pattern [8], [9]. These methods are based on the following intuition: if a group of character names frequently appear together in the script, their faces also frequently appear together in the movie. However, these methods do not consider the temporal order in which the faces and names appear, and instead focus on the aggregated co-occurrence statistics only. We propose several methods for aligning faces in movies with their corresponding character names in scripts by exploiting the temporal ordering constraints between these two modalities.

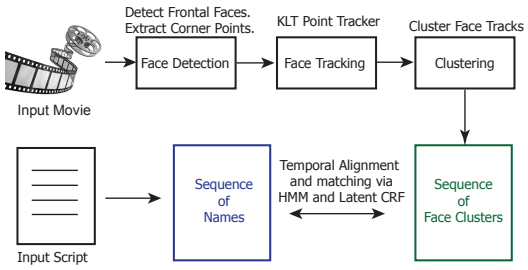


Fig. 1. An overview of our pipeline for aligning face tracks in movies with character names in scripts.

Our work is similar to the state-of-the-art work by Zhang et al. [10], which also exploits temporal cues for aligning a sequence of face clusters extracted from a movie with the sequence of names extracted from the associated script. Their method divides each movie and script into an equal number of temporal bins, and aligns each face cluster with a name based on the symmetric Kullback-Liebler (KL) Divergence [11] of their temporal distributions over those bins. The alignment accuracy of this method, however, is highly sensitive to the number of temporal bins, and it does not work well for less frequent movie characters. We extend the state-of-the-art by applying probabilistic latent variable models that align each movie segment with the corresponding script segments, and simultaneously align each face cluster within a movie segment with the corresponding character name in the script segment. While the previous methods focus on identifying the major characters only, we include all the face tracks in our evaluation.

The proposed alignment models are based on our prior work [12], [13], which aligns a small number of wetlab videos recorded in a controlled environment (12 videos, average duration of 5 minutes) with the corresponding natural language instructions in text protocols. In this study, we deal with movies which have significantly longer duration and contain more diverse objects and actions. The input to our system is a

collection of movies, each paired with a script (Figure I). We first detect frontal faces from each video frame, and track them using a Kanade, Lucas, and Tomasi (KLT) point tracker [14]. Next, we apply face clustering to group similar faces together and obtain a sequence of face clusters from the movie. We experiment with a wide variety of clustering methods and feature spaces, including the state-of-the-art Convolutional Neural Networks (CNN) based *FaceNet* embedding [15]. On the text side, we extract a sequence of character names from each script. Finally, we apply a generative Hidden Markov Model (HMM) and a discriminative Latent Variable Conditional Random Field (LCRF) to align the sequence of face clusters with the corresponding sequence of character names.

The primary contributions of this paper are as follows:

- We apply probabilistic latent variable alignment models, both generative (HMM) and discriminative (LCRF), for automatically aligning faces in movies and the character names in scripts without closed-captions or subtitles.
- We compare our methods with the state-of-the-art method for matching movie faces with names [10], which is significantly outperformed by our alignment methods (especially LCRF).
- We incorporate gender-based features in the LCRF model, and show the effectiveness of such features for improving face-to-name alignment.
- Finally, we experiment with several different face clustering methods (different features, distance metrics, and algorithms), and show comprehensive results regarding the impact of clustering accuracy on the accuracy of face-to-name alignment. We show that the convolutional neural network based FaceNet embedding and Landmark SIFT features provide best alignment accuracies.

II. Related Work

A. Closed-Caption or Subtitle-based Alignment

Previous papers on automatically associating faces in videos with the corresponding names in text primarily relied on closed-captions/subtitles with precise timestamps. Everingham et al. [1] aligned movie subtitles with scripts using dynamic time warping, and then used computer vision features to exploit similarities in faces and clothes. Cour et al. [3] proposed a generative model to jointly segment and parse a video into a hierarchy of shots and scenes, using both closed-captions and script. Ramanathan et al. [6] extended [3] by applying coreference resolution to resolve pronouns and other ambiguous mentions in the script. All these methods assume nearly perfect knowledge of the true alignment between script and movie, obtained through closed-captions/sub-titles. However, movie to script alignment in the absence of closed-captions/subtitles remains an extremely challenging task, and has not been explored much.

B. Alignment without Closed-caption/Subtitles

A few previous papers attempted the challenging task of movie-to-script alignment, without any subtitles/closed-captions. Sankar et al. [7] trained supervised classifiers to recognize the faces of all the major characters and applied the classifiers for aligning movies with scripts in the absence of subtitles. Training supervised classifiers, however, requires manually labeled face images for each of the main characters. Unsupervised methods for aligning movie faces with their names typically rely on face clustering and graph matching [8], [9]. These methods first cluster similar face-tracks together and construct two separate weighted graphs G_{faces} and G_{names} representing the co-occurrence structure among the face-clusters and the script names respectively. The correspondences between the vertices between the two graphs were learned using bipartite graph matching. While these approaches considered the co-occurrence patterns between names and faces, they ignored the global temporal ordering constraints between movies and scripts.

In the literature, we found only two methods that considered the global temporal ordering constraints for movie-to-script alignment [16], [10]. Liang et al. [16] proposed a generative Hidden Semi-Markov model, *TVParser*, for automatically grouping movie-shots into scenes, and jointly aligning these scenes to their corresponding script segments. However, finer grained segmentations and alignment between movie frames and script sentences have not been considered. Zhang et al. [10] proposed a movie-to-script alignment model that explicitly considered the temporal ordering constraints between movie and script. They represented each input movie as a sequence of face clusters and the corresponding script as a sequence of character names. Each face cluster sequence and character name sequence were divided into an equal number of temporal bins, and a temporal distribution has been estimated over these bins for each face cluster and character name. Finally, the name sequence was aligned with the corresponding face sequence based on the symmetric Kullback-Leibler (KL) divergence [11] between the temporal distribution of individual face clusters and character names. Zhang et al. used fixed duration bins for estimating the temporal distributions for names and faces. Such arbitrary binning, however, is likely to introduce errors in alignment, especially for minor characters who do not appear frequently in the script.

We extend the state-of-the-art alignment approach [10] by applying rich probabilistic latent variable models, which explicitly learn a probability distribution over the alignments between names and faces using IBM Model 1 and a Hidden Markov Model (HMM). Furthermore, we apply a discriminative latent CRF model with many informative features.

C. Aligning Videos with Text

Recently, there has been a growing interest in automatically aligning videos with text documents without direct human supervision [12], [13], [17], [18]. Our algorithms are based on our prior work on aligning videos of biological experiments in wetlabs with the corresponding sentences in a text protocol [12], [13]. The wetlab videos were fairly short, contains activities by a single agent, recorded in a fairly controlled environment (similar lighting and background), and objects were distinguished by color. Movies and TV show episodes

Fig. 2. Examples of faces detected in our dataset.

are usually more complex and diverse and tend to have longer duration. Furthermore, the wetlab videos were recorded via Kinect, which are not available for movies.

D. Face Clustering

A key bottleneck of our method is performing reasonably accurate face clustering, which aims to assign the faces of the same person to the same cluster and the faces of different people to different clusters. Face clustering is an extremely challenging task due to variations in view points, pose, facial expression, illumination, scale, and occlusions (Figure 2). As a result, faces of the same person often have more appearance variations than the faces of different people.

Early works on face clustering [19], [20] focus on learning robust distance metrics invariant to translation, rotation, and other affine transformations. These methods did not exploit the inherent temporal constraints arising in videos: (1) faces that belong to the same track must group together in the same cluster (i.e., *must-link*) and (2) faces belonging to two different tracks that overlap in time must go to different clusters (i.e., *cannot-link*). Both these constraints were exploited by the more recent works [21], [22].

Over the last few years, several Convolutional Neural Network (CNN) based face recognition and verification algorithms have achieved near-human performance on standard benchmark datasets [23], [15]. Schroff et al. [15] proposed *FaceNet* – a CNN which embeds face images to a 128 dimensional Euclidian space such that squared L2 distance between two faces in the embedding space directly corresponds to the dissimilarity between the faces. We use OpenFaceNet [24], which is an open-source implementation of *FaceNet*.

The existing face clustering methods typically exclude all the faces belonging to the minor characters, and apply clustering on the faces of major characters only. However, removing the faces of minor characters requires the knowledge of their identity, which is not usually available for real-world unsupervised learning tasks. We evaluate in a more realistic setting and consider all the face tracks in our experiments without relying on their ground truth identities.

III. Data Preprocessing Pipeline

First, we detect frontal faces from every video frame, and use these faces to initialize a KLT point tracker, which extracts a collection of face tracks extracted from the entire video. Next, we apply several different clustering methods to group similar face tracks together. Finally, we extract a sequence of face clusters in the same order as they appear in the movies and a sequence of character names in the order in which they appear in the scripts, and apply our alignment algorithms to align these two heterogeneous sequences.

A. Frontal Face Detection

We apply the standard Viola-Jones cascade face detector [25] to detect frontal faces in every video frame. To avoid false detections, we set a high merging threshold and apply a threshold on the minimum bounding box size of the face.

B. Face Tracking

We detect corner points from each of the detected frontal faces using the Minimum Eigenvalue algorithm [26], and track them using a KLT point tracker. We also keep track of the amount of overlap of the bounding box of the tracked feature points with any of the previously detected faces. If a detected face has more than 50% overlap with the bounding box containing the tracked feature points, then we decide that face to be already a part of that track, and re-estimate the corner points from that face. Since tracks initiated from false positive face detections typically do not have enough temporal support, we filter out many spurious tracks by applying a threshold on the minimum track length (set to 20 in our experiments). It also allows us to filter out the face tracks detected from the title/cast segment of movies, because the title segments show each of the main characters for a very short duration.

C. Face Track Clustering

We apply constrained spectral clustering to group face tracks belonging to the same character. A key challenge is to find a distance measure or a feature space invariant to undesired transformations (e.g., view point, lighting, and pose variation). We experiment with standard features including SIFT, Sparse Coding, and pixel-level Hue and Saturation color features. Furthermore, we experiment with the state-of-the-art Convolutional Neural Network (CNN) face embedding system – *FaceNet* [15]. We use the pre-trained models from the open-source implementation by Amos et al. [24]. Here is a list of different distance measures that we tried:

FaceNet Distance: Given two face images I_1 and I_2, we estimate two 128-dimensional feature vectors $\mathbf{v}_{I_1}^{FaceNet}$ and $\mathbf{v}_{I_2}^{FaceNet}$ for I_1 and I_2 respectively, by applying the pre-trained FaceNet models. The FaceNet distance $d_{FaceNet}(I_1, I_2)$:

$$d_{FaceNet}(I_1, I_2) = \|\mathbf{v}_{I_1}^{FaceNet} - \mathbf{v}_{I_2}^{FaceNet}\|_2^2 \quad (1)$$

Landmark SIFT Distance: We extract SIFT features from 9 landmark points detected from the faces in each track [1]. Let $\mathbf{v}_I^{sift}$ be the SIFT feature descriptor for the image I. The Landmark SIFT distance $d_{sift}(I_1, I_2)$ is estimated as:

$$d_{sift}(I_1, I_2) = \|\mathbf{v}_{I_1}^{sift} - \mathbf{v}_{I_2}^{sift}\|_2^2 \quad (2)$$

Hue-Saturation Distance: The SIFT features were sensitive to illumination variations. Recent work has shown that the hue and saturation components in the HSV color space are relatively robust to illumination variation [27]. This motivated us to experiment with a distance measure $d_{hs}(I_1, I_2)$:

$$d_{hs}(I_1, I_2) = \|I_1^h - I_2^h\|_2^2 + \|I_1^s - I_2^s\|_2^2 \quad (3)$$

Mirrored Hue-Saturation Distance: To address the challenges of pose variations, we compute distance between both the original image pairs and their mirrored combinations, and choose the minimum distance:

$$d_{mir-hs}(I_1, I_2) = \min\Big\{ d_{hs}(I_1, I_2), d_{hs}(I_1^{mir}, I_2), d_{hs}(I_1, I_2^{mir}), d_{hs}(I_1^{mir}, I_2^{mir}) \Big\} \quad (4)$$

Low-dimensional Projection Distance: We also train a low-dimensional embedding space for face images using Sparse Coding [28], and estimate the distance $d_{sc}(I_1, I_2)$ between the sparse coefficient vectors.

We also incorporate the cannot-link and must-link constraints in the affinity matrix of spectral clustering. We have noticed many cases where the faces of male and female characters were assigned to the same cluster. To avoid this obvious error, we add another cannot-link constraints indicating that two face tracks can not link if they are detected as different genders. To detect the genders of individual tracks, we sample 5 face images from each track, detect the gender for each image using the SHORE^{TM} framework [29], and take a majority vote. Due to the gender based cannot-link constraints, the constrained spectral clustering performed well in discriminating between the male and female faces, and each of the clusters consisted of predominantly either male or female faces, but not both. To address the variability due to head pose, we normalize each face via an affine transformation using the DLIB library (http://dlib.net/). In addition to the variants of spectral clustering, we also apply the HMRF clustering method [21] [1] and perform an extensive comparison.

D. Extracting Character Names from Scripts

In our scripts, each name starts with a capital letter, followed by a colon character ':'. We apply a simple text pattern search to extract all the character names from each script, and build a sequence of names in the order they appear in the script. Detailed parsing of the script sentences and dialogues [3], [6] is left as future work.

IV. Aligning Character Name Sequences with Face Cluster Sequences

The input to our system is a dataset containing N pairs of observations $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, where $\mathbf{x}_i$ represents the i^{th} script in our datasest, and $\mathbf{y}_i$ represents the corresponding movie. We merge every two consecutive lines in a script to create a chunk and extract the character names mentioned in those lines. Each input script is represent as $\mathbf{x}_i = \{X_{i,1}, X_{i,2}, \ldots, X_{i,m_i}\}$, where $X_{i,m}$ is the set of character names mentioned in the m^{th} script chunk in $\mathbf{x}_i$. We also divide each input movie into 2-second long chunks, and extract the face clusters present in the frames in those chunks. Any movie chunk that does not overlap with any of the face tracks is ignored by our system. Each video is represented as $\mathbf{y}_i = \{Y_{i,1}, Y_{i,2}, \ldots, Y_{i,n_i}\}$, where $Y_{i,n}$ is the set of face clusters present in the n^{th} movie chunk in $\mathbf{y}_i$. Let m_i be the number of script chunks in $\mathbf{x}_i$ and n_i be the number of movie chunks in $\mathbf{y}_i$. We aim to align each movie chunk $Y_{i,n}$ to one of the script chunks $X_{i,m}$ and simultaneous align each face $y \in Y_{i,n}$ with the corresponding character name $x \in X_{i,m}$.

[1] Code downloaded from https://sites.google.com/site/baoyuanwu2015/

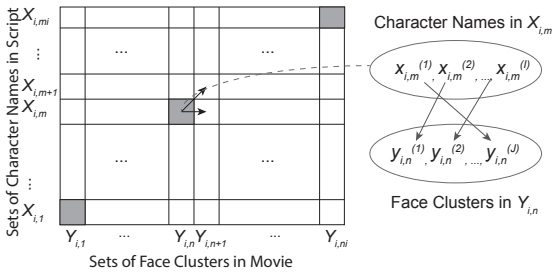


Fig. 3. An illustration of the proposed alignment models. We align each movie chunk to one of the script chunks. Furthermore, we align each face cluster in a movie chunk to a character name in the corresponding script chunk.

Let $\mathbf{h}_i$ be the latent variable representing the alignment between the script chunks in $\mathbf{x}_i$ and the movie chunks in $\mathbf{y}_i$. Formally, $\mathbf{h}_{i,n} \in \{1, \ldots, m_i\}$, for $1 \leq n \leq n_i$, where $\mathbf{h}_{i,n} = m$ indicates that the movie chunk $Y_{i,n}$ is aligned to the script chunk $X_{i,m}$. Let V_X and V_Y be the vocabulary of character names and face clusters respectively. Our models contain additional boolean latent variables representing the mapping of each face cluster $y \in V_Y$ to one of character names $x \in V_X$. Our goal is to learn the overall alignment $\mathbf{h}_i$ between $\mathbf{x}_i$ and $\mathbf{y}_i$ and the global mapping variables between every face cluster and character name.

A. Hidden Markov Model (HMM)

We first apply a hierarchical generative model [12], which assumes that each movie chunk $Y_{i,n}$ is generated from one of the script chunks $X_{i,m}$ according to a Hidden Markov Model (HMM) and each face cluster $y \in Y_{i,n}$ is generated from one of the character names $x \in X_{i,m}$ according to IBM Model 1 (Figure 3). The model parameters include a matching probability table $T = \{p(y|x)\}$ for each $x \in V_X$ and $y \in V_Y$, representing the probability of observing the face cluster y given the name x. The probability of generating a set of blobs $Y_{i,n} = \{y_{i,n}^{(1)}, \ldots, y_{i,n}^{(J)}\}$ from the set of nouns $X_{i,m} = \{x_{i,m}^{(1)}, \ldots, x_{i,m}^{(L)}\}$ according to IBM Model 1 is:

$$\mathrm{P}\left(Y_{i,n}|X_{i,m}\right) = \frac{\epsilon}{(L)^J} \prod_{j=1}^{J} \sum_{l=1}^{L} p(y_{i,n}^{(j)}|x_{i,m}^{(l)}), \quad (5)$$

which becomes the emission probability of our HMM at the alignment state $\mathbf{h}_{i,n} = m$. Following the Markov assumption, the alignment state $\mathbf{h}_{i,n} = m$ depends on the alignment state for the previous video segment $\mathbf{h}_{i,n-1} = m'$. The transition probability $\mathrm{P}(\mathbf{h}_{i,n} = m|\mathbf{h}_{i,n-1} = m')$ is parameterized by the jump size $(m - m')$ between adjacent alignment points:

$$\mathrm{P}(\mathbf{h}_{i,n} = m|\mathbf{h}_{i,n-1} = m') = c(m - m') \quad (6)$$

where $c(k)$ represents the probability of jumps of distance k. We only allow monotonic transitions such that if $\mathbf{h}_{i,n} = m$, then we allow $\mathbf{h}_{i,n+1} \in \{m, m + 1\}$.

TABLE I

THE AVERAGE ACCURACY (% OF FACE TRACKS MATCHED TO THE CORRECT CHARACTER NAMES) OF ALIGNING FACE TRACKS WITH CHARACTER NAMES FOR THE TWO DATASETS: TBBT AND FRIENDS. FOR EACH CLUSTERING METHOD, WE REPORT THE AVERAGE ACCURACY (AND STANDARD DEVIATION) FOR THE BEST SETTING OF K. THE ACCURACY OF DIAGONAL ALIGNMENT IS DETERMINISTIC AND DOES NOT DEPEND ON CLUSTERING.

Dataset	System	Alignment Accuracy (%)					
		FaceNet	Mir Hue/Sat	Hue/Sat	SIFT	SC	HMRF
TBBT	Diagonal	30.1 (0.0)	30.1 (0.0)	30.1 (0.0)	30.1 (0.0)	30.1 (0.0)	30.1 (0.0)
	KL-Div [10]	39.7 (0.7)	38.4 (2.4)	35.7 (2.6)	44.5 (1.1)	32.8 (0.0)	37.3 (2.2)
	HMM	54.7 (1.8)	46.2 (0.1)	42.9 (2.3)	64.6 (1.6)	46.5 (2.2)	54.3 (2.5)
	LCRF	**55.9** (2.3)	**50.1** (2.5)	**50.3** (0.2)	**65.1** (1.7)	**49.6** (1.8)	**56.5** (2.6)
Friends	Diagonal	23.0 (0.0)	23.0 (0.0)	23.0 (0.0)	23.0 (0.0)	23.0 (0.0)	23.0 (0.0)
	KL-Div [10]	32.0 (1.9)	36.4 (0.0)	28.6 (0.0)	29.6 (1.2)	27.8 (0.6)	27.7 (1.5)
	HMM	37.5 (2.1)	33.1 (0.0)	33.2 (0.0)	33.6 (0.0)	30.6 (2.2)	22.6 (1.6)
	LCRF	**43.4** (1.3)	**41.5** (0.3)	**40.8** (0.7)	**37.8** (0.2)	**38.8** (0.2)	**27.8** (2.2)

The matching probability table $T = \{p(y|x)\}$ is initialized uniformly, i.e., we set $p(y|x) = 1/|V_Y|$ for all $y \in V_Y$ and $x \in V_X$. We also initialize the jump probabilities $c(k)$ uniformly. The matching probabilities T and the jump probabilities c are learned from the input dataset via the Expectation Maximization (EM) algorithm. The best alignment for $(\mathbf{x}_i, \mathbf{y}_i)$ is inferred using Viterbi-like dynamic programming.

B. Latent Conditional Random Field (LCRF)

Next, we apply a Latent Conditional Random Field (LCRF) [13] for aligning movies with scripts. We assume the script $\mathbf{x}_i$ is the observed input, and we aim to predict the movie $\mathbf{y}_i$. Similar to HMM, the alignment $\mathbf{h}_i$ is treated as latent variables. The feature function $\mathbf{\Phi}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}_i)$ maps the input observation $(\mathbf{x}_i, \mathbf{y}_i)$, and their latent alignment vector $\mathbf{h}_i$ to a d-dimensional feature vector. Our goal is to learn the weights $\mathbf{w} \in \mathbb{R}^d$ for these features.

Given the observed script $\mathbf{x}_i$ and a fixed video length n_i, the conditional probability of the output variable $\mathbf{y}_i$ is:

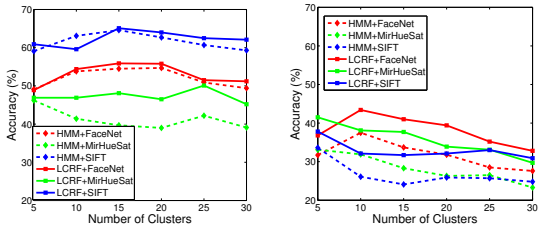
$$p(\mathbf{y}_i|\mathbf{x}_i, n_i) = \sum_{\mathbf{h}_i} p(\mathbf{y}_i, \mathbf{h}_i|\mathbf{x}_i, n_i) \tag{7}$$

The conditional probability distribution $p(\mathbf{y}_i, \mathbf{h}_i|\mathbf{x}_i, n_i)$ is parameterized using a log-linear model:

$$p(\mathbf{y}_i, \mathbf{h}_i|\mathbf{x}_i, n_i) = \frac{\exp \mathbf{w}^T \mathbf{\Phi}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}_i)}{Z(\mathbf{x}_i, n_i)}, \tag{8}$$

where $Z(\mathbf{x}_i, n_i) = \sum_{\mathbf{y}} \sum_{\mathbf{h}} \exp \mathbf{w}^T \mathbf{\Phi}(\mathbf{x}_i, \mathbf{y}, \mathbf{h})$. The optimal values for the feature weights $\mathbf{w}$ are learned via stochastic gradient descent optimization.

We add a co-occurrence feature for every pair of name x and face cluster y, and automatically learn the weight for matching x with y. We also incorporate jump-size and diagonal alignment features [13]. Finally, we incorporate features based on the gender of the face clusters and character names to encourage matching a face cluster with a name belonging to the same gender. The gender label for a face cluster is decided by a majority voting over the detected genders for all the face tracks in that cluster. The gender of each script character is determined using a publicly accessible web-application *Genderize.io* (https://genderize.io/), which takes a first name as an input, and returns its gender label (male or female) or a null string when the query name is not in their database. We add 4 gender-based features: $(\text{Name}_{\text{male}}, \text{Face}_{\text{male}})$, $(\text{Name}_{\text{male}}, \text{Face}_{\text{female}})$, $(\text{Name}_{\text{female}}, \text{Face}_{\text{male}})$, and $(\text{Name}_{\text{female}}, \text{Face}_{\text{female}})$. We initialize the feature weights for the correct gender matching (male name to male face and female name to female face) to a small positive value $(+1.0)$, and incorrect gender matching to a small negative value (-1.0). The co-occurrence and jump size features are initialized to the log-probabilities $P(y|x)$ estimated by 10 iterations of HMM.



(a) The Big Bang Theory (b) Friends

Fig. 4. The impact of the number of clusters (K) on the alignment accuracies. The mid range values $K = 10$ or 15 appears to perform best.

V. EXPERIMENTAL RESULTS

Datasets: We experiment with two datasets: (1) TBBT and (2) Friends. The TBBT dataset consists of the first 3 episodes of the sixth season of the TV show *The Big Bang Theory* and the Friends dataset consists of the first 3 episodes of the first season of the TV show *Friends*. Approximately, each TBBT episode contains $30,000$ frames, whereas each Friends episode contains $40,000$ frames. The face detection module detected 20,819 faces from the 3 TBBT episodes and 30,515 faces from the 3 Friends episodes. We apply a KLT point tracker and extract 544 and 819 face tracks respectively.

Experiments: We cluster the detected face tracks and perform alignment using the HMM and LCRF methods. We apply the constrained spectral clustering algorithm with five different distance measures: (1) FaceNet distance, (2) Hue/Sat distance, (3) Mirrored Hue/Sat distance, (4) Sparse Coding (SC) distance, and (5) SIFT distance. Furthermore, we experimented with the state-of-the-art HMRF face clustering method. The number of clusters K is a tunable parameter, and we experiment with 6 different values of $K = [5, 10, 15, 20, 25, 30]$.

Spectral clustering may get trapped in local optima as it employs non-convex K-means clustering. For each K-means clustering, we perform 10 repeated runs with random initializations, and choose the best one from those 10 solutions. Furthermore, we run the spectral clustering algorithm 10 times (each with 10 K-means repetitions), and report the average alignment accuracy and standard deviation.

Evaluation: We evaluate the accuracy of alignment by estimating the percentage of face tracks that were aligned to the correct character names. Since we divide the video into 2-second long chunks, one track may appear in many different chunks, and therefore can potentially be aligned to different character names in the script. For each track, we assign it the character name that it has been aligned to the majority number of times. Table I reports the average alignment accuracy and standard deviation for the best choice of K on the two datasets. We compare our HMM and LCRF models with the state-of-the-art KL-divergence based alignment [10] and a simple diagonal alignment baseline. For the KL-divergence baseline, we tried different number of temporal bins in the range 10 to 100 with an increment of 10, and chose 20 as it provided the best accuracy. The diagonal alignment assumes that each script line aligns to equal number of movie chunks.

Discussion: Our HMM and LCRF methods significantly outperform the baseline methods (Table I). LCRF achieves the best accuracy for all the clustering methods on both datasets, presumably due to the effectiveness of the gender features.

The best alignment accuracy for TBBT is 65.1% (using SIFT features), which is significantly higher than that for Friends (43.4% using FaceNet). This is partly because the faces of the main characters in TBBT are quite distinct from each other, which is not the case for Friends. Furthermore, the videos in TBBT have relatively better lighting and fewer minor characters. For similar reasons, SIFT and HMRF worked well for TBBT, whereas the more advanced FaceNet embedding did better for Friends. The Mirrored-Hue/Sat distance performed slightly better than the Hue/Sat distance as it was robust to pose variation.

VI. Conclusion and Future Work

We apply unsupervised alignment algorithms for automatically aligning faces in complex movie scenes with their corresponding character names in scripts. Our algorithms significantly outperform the previous state-of-the-art. As a future direction, we would like to perform joint alignment and clustering and incorporate knowledge based features from online movie databases. The proposed methods can be applied for aligning movies with the story-books that they are adapted from [30], [31], when subtitles are not available.

Acknowledgment

Funded by Intel ISTC-PC, NSF-1319378, NYS CoE DS, and NSF IIS-1446996.

References

[1] M. Everingham, J. Sivic, and A. Zisserman, ""Hello! My name is... Buffy"–automatic naming of characters in TV video." in *BMVC*, vol. 2, no. 4, 2006, p. 6.

[2] A. Gaidon, M. Marszalek, and C. Schmid, "Mining visual actions from movies," in *BMVC*, 2009, pp. 125–1.

[3] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar, "Movie/script: Alignment and parsing of video and text transcription," in *ECCV*, 2008, pp. 158–171.

[4] M. Aparício, P. Figueiredo, F. Raposo, D. M. de Matos, and R. Ribeiro, "Summarization of films and documentaries based on subtitles and scripts," *CoRR*, vol. abs/1506.01273, 2015.

[5] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE CVPR*, 2008, pp. 1–8.

[6] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei, "Linking people in videos with "their" names using coreference resolution," in *ECCV*. Springer, 2014, pp. 95–110.

[7] P. Sankar, C. V. Jawahar, and A. Zisserman, "Subtitle-free movie to script alignment," in *BMVC 2009*, 2009, pp. 121.1–121.11.

[8] Y. Zhang, C. Xu, H. Lu, and Y.-M. Huang, "Character identification in feature-length films using global face-name matching," *IEEE Transactions on Multimedia*, vol. 11, no. 7, pp. 1276–1288, Nov. 2009.

[9] J. Sang, C. Liang, C. Xu, and J. Cheng, "Robust movie character identification and the sensitivity analysis," in *IEEE ICME 2011*, July 2011, pp. 1–6.

[10] Y. Zhang, Z. Tang, C. Zhang, J. Liu, and H. Lu, "Automatic face annotation in TV series by video/script alignment," *Neurocomputing*, vol. 152, pp. 316 – 321, 2015.

[11] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, pp. 79–86, 1951.

[12] I. Naim, Y. C. Song, Q. Liu, H. Kautz, J. Luo, and D. Gildea, "Unsupervised alignment of natural language instructions with video segments," in *AAAI*, 2014.

[13] I. Naim, Y. C. Song, Q. Liu, L. Huang, H. Kautz, J. Luo, and D. Gildea, "Discriminative unsupervised alignment of natural language instructions with corresponding video segments," in *NAACL*, 2015.

[14] C. Tomasi and T. Kanade, "Detection and tracking of point features," *IJCV*, 1991.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE CVPR*, June 2015.

[16] C. Liang, C. Xu, J. Cheng, and H. Lu, "TV-Parser: An automatic TV video parsing method," in *IEEE CVPR*, 2011, pp. 3377–3384.

[17] J. Malmaud *et al.*, "What's cookin'? interpreting cooking videos using text, speech and vision," in *NAACL HLT*, 2015, pp. 143–152.

[18] P. Bojanowski *et al.*, "Weakly-supervised alignment of video with text," in *IEEE ICCV*, 2015.

[19] A. Fitzgibbon and A. Zisserman, "On affine invariant clustering and automatic cast listing in movies," in *ECCV*, 2002, pp. 304–320.

[20] A. W. Fitzgibbon and A. Zisserman, "Joint manifold distance: a new approach to appearance based clustering," in *IEEE CVPR*, vol. 1, 2003, pp. 1–26.

[21] B. Wu, Y. Zhang, B.-G. Hu, and Q. Ji, "Constrained clustering and its application to face clustering in videos," in *IEEE CVPR*, June 2013, pp. 3507–3514.

[22] X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh, "Constrained multi-view video face clustering," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4381–4393, Nov 2015.

[23] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *IEEE CVPR*, 2014, pp. 1701–1708.

[24] B. Amos *et al.*, "OpenFace 0.1.1: Face recognition with Google's FaceNet deep neural network." October 2015.

[25] P. Viola and M. J. Jones, "Robust real-time face detection," *IJCV*, vol. 57, no. 2, pp. 137–154, 2004.

[26] J. Shi and C. Tomasi, "Good features to track," in *IEEE CVPR*, 1994, pp. 593 – 600.

[27] N. Vretos, V. Solachidis, and I. Pitas, "A mutual information based face clustering algorithm for movie content analysis," *Image and Vision Computing*, vol. 29, no. 10, pp. 693–705, 2011.

[28] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *NIPS 2006*, 2006, pp. 801–808.

[29] B. Froba and A. Ernst, "Face detection with the modified census transform," in *IEEE FG*, 2004, pp. 91–96.

[30] M. Tapaswi, M. Bäuml, and R. Stiefelhagen, "Book2movie: Aligning video scenes with book chapters," in *IEEE CVPR*, 2015, pp. 1827–1835.

[31] Y. Zhu *et al.*, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *IEEE ICCV*, 2015.