Naming TV Characters by Watching and Analyzing Dialogs

Monica-Laura Haurilet Makarand Tapaswi Ziad Al-Halah Rainer Stiefelhagen
Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
{haurilet, tapaswi, ziad.al-halah, rainer.stiefelhagen}@kit.edu

Abstract

Person identification in TV series has been a popular research topic over the last decade. In this area, most approaches either use manually annotated data or extract character supervision from a combination of subtitles and transcripts. However, both approaches have key drawbacks that hinder application of these methods at a large scale – manual annotation is expensive and transcripts are often hard to obtain. We investigate the topic of automatically labeling all character appearances in TV series using information obtained solely from subtitles. This task is extremely difficult as the dialogs between characters provide very sparse and weakly supervised data.

We address these challenges by exploiting recent advances in face descriptors and Multiple Instance Learning methods. We propose methods to create MIL bags and evaluate and discuss several MIL techniques. The best combination achieves an average precision over 80% on three diverse TV series. We demonstrate that only using subtitles provides good results on identifying characters in TV series and wish to encourage the community towards this problem.

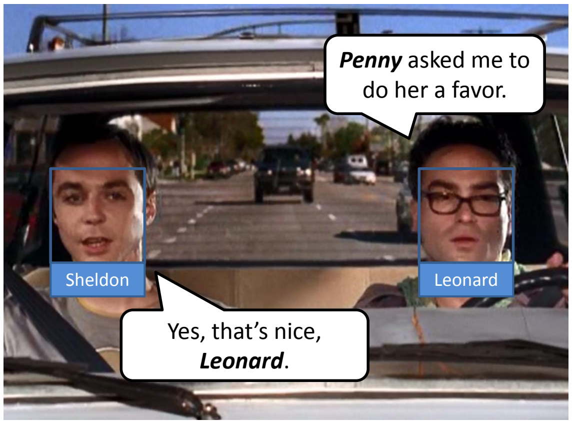
Penny asked me to do her a favor.
Sheldon
Leonard
Yes, that's nice, ***Leonard***.

Figure 1: Example of a dialog in a TV series episode, where we are able to learn the name of one of the characters. Based on the 3^{rd} person reference to *Penny*, we also learn that *Penny* is not among the two visible characters.

1. Introduction

Identifying characters in TV series has seen over a decade of research. Most prominently, since the work by Everingham *et al.* [10], the area has seen a paradigm shift towards using subtitles and transcripts to perform fully automatic identification. However, online transcripts are often difficult to find and come in various forms.

When we humans watch a (new) TV series, we are very good at latching onto the names of characters based simply on their interaction with each other. By listening to dialogs such as "Yes, that's nice, Leonard." we infer that the person who is seen but is not talking during this shot is probably Leonard (see Figure 1). Among several works in the area of identifying TV / movie characters, we know of only one, [9], which uses supervision only in the form of dialogs. From hereon, we refer to dialogs as subtitles and use them as a proxy for perfect automatic speech recognition on the spoken dialogs.

Transcripts have seen use in various applications: person identification [3, 10, 25], action recognition [4, 16], joint coreference resolution and person-id [21], and even to analyze the structure of videos [7]. However, transcripts come in various forms and styles and can sometimes be difficult to obtain for complete seasons. Sometimes, transcripts may only contain dialogs, and the names of characters that are critical to obtain weak labels (as in [10]) are missing[1].

Recently, large breakthroughs have been achieved in computer vision through the use of deep (convolutional) neural networks. Face descriptors [19, 23, 26] tapped at the last layer of CNNs have shown close-to-human performance on classical face image and video verification [13, 30] tasks. We believe that the community is well poised to drop transcripts as the form of supervision used to obtain weak labels to annotate all faces. Doing so will make automatic person identification methods truly appli-

[1] Compare original production scripts obtained from https://sites.google.com/site/tvwriting/ against fan transcripts http://www.buffyworld.com/buffy/transcripts/079_tran.html or https://bigbangtrans.wordpress.com/.

cable to all kinds of films and TV series data. Another advantage of dropping transcripts is applying the identification approaches to videos produced in other languages, making them truly widely applicable.

In this paper, we revisit the problem of person identification in TV series by using supervision only in the form of subtitles. We wish to promote this problem as a challenge to all people working in this area. To this end, we evaluate our proposed method on a new data set consisting of 8 episodes of the TV series "Lost" in addition to the KIT TV data set [3]. While "Lost" has been used previously by [8, 9], the data does not come with face tracks, which makes it difficult to fully exploit. As part of technical contributions, we present a framework to use Multiple Instance Learning (MIL) techniques on our challenging problem. We introduce methods to use subtitles and create MIL bags and evaluate various iterative or non-iterative MIL techniques on the three TV series.

2. Related work

Character identification/retrieval has come a long way since 2005. One of the earliest work in this area is [24] where the authors propose the concept of matching sets of faces (tracks) to perform retrieval. Soon after, Everingham *et al*. [10] show that using subtitles and transcripts allows to obtain exemplar faces from which character models can be trained. We present a brief overview of the related work in this area based on the type of supervision used to train character models.

Transcripts and subtitles. Since [10], many works (*e.g*. [3, 8, 15, 25]) rely on subtitle-transcript matching to obtain weak labels for characters of a TV show. [25] extends [10] to include multi-pose face tracks and uses Multiple Kernel Learning for classification. [8] proposes to treat the problem as an ambiguous labeling task where each face track is potentially labeled by multiple character names. [15] proposes to use semi-supervised multiple instance learning while [3] argues that all face tracks (weakly labeled or not) can be used to improve classifiers and proposes to jointly model supervised and unsupervised data along with constraints. [4] uses movie scripts to not only identify characters, but jointly model character actions and identities. Recently, [29] analyzes and improves upon the weak label association of [10]; and [6] uses Minimum Spanning Trees to analyze and associate tracks.

As is evident, using subtitles and transcripts jointly is very popular for identifying characters and has provided steady progress through different machine learning approaches and evolving vision methods (face detection, alignment, descriptors). However, we argue that using supervision from transcripts is unnatural (not the way humans identify characters) and transcripts are often difficult to find for not-so-popular TV series.

Other supervision. There are some scenarios where only transcripts are used as the form of supervision. [22] proposes a method to align scripts to videos in the absence of subtitles. [21] jointly model co-reference resolution and character identification using only scripts.

A few instances also use manual supervision as a means to obtain training data. [20] performs track clustering and argue that once clustered, manually labeling these clusters is a much simpler task. [27] sets aside few episodes as training data and uses manually labeled tracks from them to create models. Both of these are hard to scale to larger data sets.

Only subtitles. While transcripts gained large popularity, we know of only one paper in literature [9], that uses only subtitles to identify characters. Subtitles can be thought of as the output of a perfect automatic speech recognition model and are thus closest to how humans identify and follow characters. Most related to our work, Cour *et al*. [9] use a mixture of multiple cues – appearance, dialog person reference, video-editing, mouth movement – to identify characters. As a first step, face tracks are clustered using Conditional Random Fields. Groups of face tracks are identified using a linear combination of convex losses which capture the supervision from dialogs.

In this work, we show that the advances in face descriptors and multiple instance learning allow to directly leverage the sparse and indirect supervision obtained from subtitles. As opposed to hand-crafted loss functions, we model the problem via the MIL framework. We propose a novel approach to create MIL bags containing face tracks and evaluate several MIL techniques to generate training data. Finally, we train character-specific SVM classifiers to name all tracks in the series.

3. MIL: Bag creation and model learning

In this section we present novel techniques to obtain annotated bags for training character models. We then present and discuss the MIL techniques that can be applied on such bags (see Figure 2). Finally, the annotated instances are used for classifying all tracks using an SVM.

3.1. Resolving name references in subtitles

We leverage name mentions in subtitles to obtain cues about the presence or absence of characters in a certain time period. These name mentions are classified into three groups: 1$^{\text{st}}$, 2$^{\text{nd}}$ and 3$^{\text{rd}}$ person references. 1$^{\text{st}}$ and 2$^{\text{nd}}$ person name references indicate that the character appears in a short temporal neighborhood of the utterance. On the other hand, 3$^{\text{rd}}$ person references suggest that the character being talked about may not be in the scene.

Classifying name references into these three groups is not trivial. Unlike novels, dialogs are usually short and often end abruptly. Some of the dialogs do not even contain a verb.

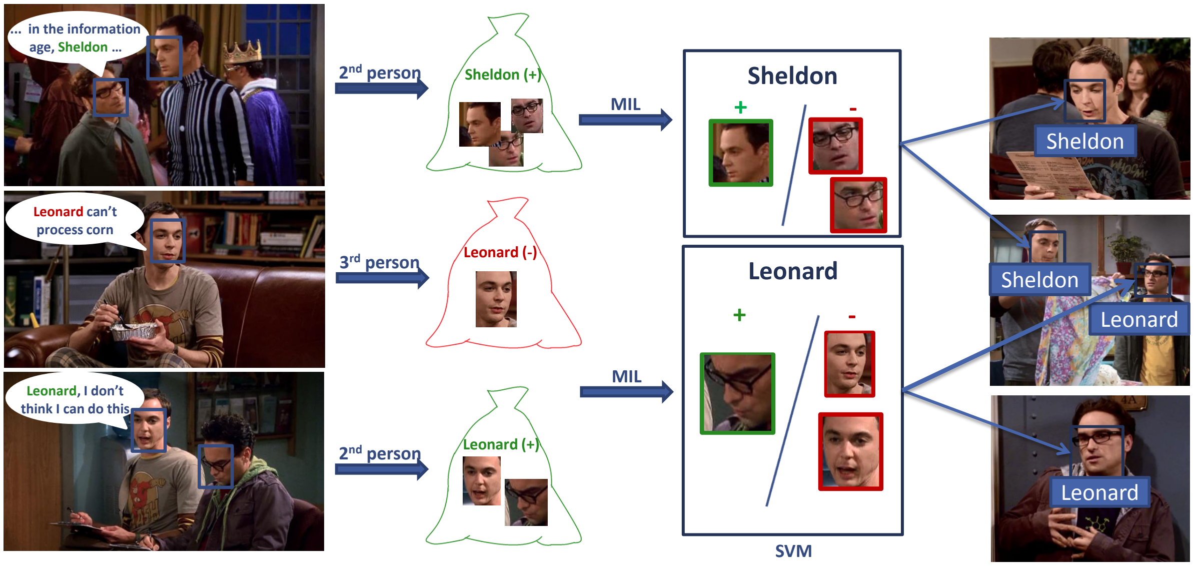

Figure 2: Overview of our approach. On the left we show the process of creating positive and negative bags consisting of face tracks surrounding 2^{nd} (3^{rd}) person name references in dialogs. In the center, MIL resolves the bags to provide positive and negative instances. Finally, we train character-specific SVM models on these instances to label all tracks.

To classify name mentions, we define a set of simple grammar rules to cope with such impediments:

R1-Templates We propose a set of 4 templates; namely ("I am *name*", "I'm *name*") for 1^{st} person classification, and ("You are *name*", "You're *name*") for 2^{nd} person.

R2-Addressing A characteristic of spoken language is that long dependent clauses are rare. As a result, almost all names which are at the beginning or the end of sentences and are separated by a comma can be classified as 2^{nd} person references. For example, "*Sheldon*, this was your idea." and "Save it for your blog, *Howard*". Furthermore, we notice that there are many one word sentences that contain a name mention like "*Leonard*". We notice that in most cases these name mentions are of the 2^{nd} person type, and hence we label them as such.

R3-Verb form To obtain 3^{rd} person references, we consider the form of the verb which appears near the name. If the verb associated to the name is in third person, we classify that name as a 3^{rd} person reference. Here, we apply the Standford CoreNLP toolkit [17] to obtain the verbs that are in third person. Due to lexical ambiguities, we are able to use verbs in the present tense and singular form (*e.g*. he *sees*). However, others in plural form or other tenses (*e.g*. they *saw*) are ignored.

R4-Preposition We also consider prepositions to find more 3^{rd} person references. A name that follows a preposition is almost always in 3^{rd} person, *e.g*. "I'm talking to *Penny* here". Prepositions are easily located using CoreNLP.

R5-Enumeration Sometimes, the names of characters are part of an enumeration, *e.g*. "Hey, I don't know if you heard about what happened with *Leonard* and *Sheldon*". In this case, following R4, *Leonard* would be correctly labeled as 3^{rd} person. However, the name mention *Sheldon* does not come under any of the previous rules, and is hence kept unlabeled. Therefore, if no grammar rule can be applied to a certain name reference and if the name is part of an enumeration, we propagate the label of the first name in the list to the subsequent name appearances.

Finally, name references which are not part of any of the above rules are silently discarded. In our data, this forms about 7.6% of all name utterances.

3.2. Multiple Instance Learning framework

The previous step provides us with several cues about the presence of characters in the video. Specifically, we use 1^{st} and 2^{nd} person references as positive cues (the named person appears in a temporal vicinity), and 3^{rd} person references as negative cues (the named person does not appear in the neighborhood).

Establishing a direct link between a name in the subtitle and a face track is tricky since it requires identifying the speaker and potentially the character who is being spoken to. We circumvent this by assigning the name from the subtitle to a group of face tracks. Our problem is well formulated in a Multiple Instance Learning (MIL) framework.

We denote the set of all tracks as $\mathcal{T}$ and characters in the TV series by C. X is a bag that contains a set of face tracks $\{x_i\} \subset \mathcal{T}$. A bag X^{c+} is positive for a certain character $c \in C$, if it contains at least one positive instance (face track) of that character, and negative X^{c-} if all tracks do not belong to that character. We define the set of bags (both positive and negative) for character c as

$$B^c = \{X_j^c : j \in \{1, \dots, N_B^c\}\}\,, \tag{1}$$

where $N_B^c = |B^c|$ is the total number of bags.

Creating bags based on name references. As motivated before, we leverage the cues obtained from the subtitles to create sets of positive and negative bags. We tried a variety of different methods to create bags based on the duration around the utterance of the name from which tracks are selected. The chosen approach for creating bags is to collect tracks that appear in the same scene. Here, within-episode scene boundaries are computed using [28] yielding on average one scene per minute.

Positive scene-level bags. Name mentions of the 1st and 2nd person are employed to create positive bags. For each person reference of character c, we create a positive bag X^{c+} that contains all face tracks from the corresponding scene. This ensures that the character appears in at least one of the tracks.

Negative scene-level bags. On the flip side, negative bags X^{c-} contain all face tracks from the scene for which a 3rd person name reference was found for character c.

While the density of positive instances in positive bags created by picking all face tracks at the scene-level is low, we have a very high chance of each bag having at least one positive instance, thus fulfilling the MIL criteria.

When bags are created by selecting face tracks from neighboring *shots*, we see a different property. Shot-level positive bags contain only a few tracks, but are prone to being incorrect as the mentioned character might not be present among the few selected tracks. We consider a small temporal neighborhood of ± 1 shot around the name mention when creating shot-level bags, typically yielding three bags (for the previous, current and next shot). Similar to scene-level bags, the label assignment for shot-level bags is based on the person reference (1st, 2nd for positive, and 3rd for negative) of the name mention.

MIL techniques. In the second phase of our approach we apply MIL methods to label the instances (face tracks) contained in the bags. To facilitate the use of standard MIL methods, we first need to simplify our bags so that they only contain binary labels. We treat each character c and his/her set of bags B^c independent from other characters, and learn $|C|$ MIL models to label instances.

We now present a brief overview about various MIL methods. To aid in understanding, we group MIL methods into iterative and non-iterative.

Non-iterative techniques. The Normalized Set Kernel (NSK) [12] is a non-iterative MIL method, with a loss function similar to the linear SVM in the primal form. Like other methods, it encourages instances in the negative bags to be labeled as negative and prefers to label positive bags so as to obtain a high density of positive instances. In contrast to typical iterative MIL techniques (as we will discuss further), an advantage of NSK is its ability to cope with bags that may be incorrect, *i.e.* bags labeled as positive, but not containing a positive instance.

An opposing approach to NSK is to support positive bags which have a low density (as seems to be the case for scene-level bags). In particular, sparse MIL (sMIL) [5] is a good technique which has the potential to work with sparse positive bags that contain a small density of positive instances. The sMIL loss is penalized when the number of positive instances is lower than a parameter that depends on the bag size. A key difference to NSK is that the parameter is not influenced by the bag size.

Iterative techniques. We now present MIL approaches that improve the quality of labeled instances iteratively. One such method is Multiple Instance SVM (miSVM) [2]. miSVM initializes the instance labels by inheriting the bag labels. At every iteration, the current instance labels are used to train an SVM classifier. This SVM is used subsequently to update the instance labels. When a positive bag is labeled incorrectly, miSVM tends to erroneously classify the largest category of negative instances (in the positive bag) as positive.

Another example of an iterative method is Sparse Transductive MIL (stMIL) [5]. This method initializes instance labels using sparse MIL (discussed above) and iteratively updates the decision boundary to move towards regions of low data density.

Inspired by miSVM, we explore an alternative form of an iterative MIL algorithm, that we call Single Instance Single Label-SVM (SISL-SVM). Similar to miSVM, this approach trains an SVM classifier at every iteration to update instance labels. While miSVM labels instances as positive (or negative) depending on the classification score, in SISL-SVM, k top scoring instances are labeled positive, while the rest are negative.

All MIL methods discussed above learn to resolve characters independent of each other. We additionally investigate one-vs-all MIL, that only considers positive bags for all characters. We initialize the positive instances similar to miSVM, while negative instances for character c are obtained from positive bags of all other characters.

3.3. Labeling face tracks in video

The above MIL methods provide for each character c, an associated set of instances and their corresponding confidence score

$$D^c = \{(x_i, p_i), i \in \{1, \dots, N^c\}\}\,, \tag{2}$$

where N^c is the total number of tracks from all training bags B^c.

Episode	05	06	07	08	09	10	11	12
#tracks	662	879	705	776	711	794	755	745
#mentions	28	13	79	53	42	61	71	15
#unknowns	24	84	69	17	69	47	17	49

Table 1: Number of face tracks, name mentions and unkowns tracks found in 8 episodes of *Lost*.

To label all character appearances in the TV series, we train an SVM based on samples in D^c. Here, an instance $x_i \in D^c$ is considered positive if the prediction score p_i is higher than a threshold θ and negative otherwise. We adapt the threshold θ and other MIL parameters (*e.g.* the weight λ in NSK, or SVM slack parameter of iterative approaches) using one episode for each TV series. We thus obtain models for all characters of interest. Finally, all face tracks in the TV series are labeled by the name of the character whose model has scores highest.

Note that we do not currently support "unknown" characters. Consequently, each face track is assigned a character name and background characters whose names are never uttered are always misclassified.

4. Evaluation

We present an evaluation of methods described in the previous section. We first discuss the experimental setup, followed by an analysis of name mention classification and bag creation. Finally, we present the results of labeling all tracks in the TV series.

4.1. Experimental setup

We evaluate our proposed approach on three TV series: (i) *The Big Bang Theory* (BBT); (ii) *Buffy the Vampire Slayer* (Buffy); and (iii) *Lost*. We use the face tracks provided by the KIT TV data set [3] for 6 episodes each from BBT and Buffy. To facilitate comparing our methods against [9], we also evaluate on the season 1 episodes 5-12 of the TV series *Lost*.

Face tracks. We use the provided face tracks for BBT and Buffy. For Lost, we consider eight episodes from the first season similar to [9]. Faces are detected in the video using a cascade classifier [11] and are tracked via the tracking-by-detection concept using a particle filter [14]. To reduce the variance of faces with respect to deformations such as head pose and translation, we align the faces using three facial landmarks (eyes and mouth). Landmark points are detected using the Supervised Descent Method [31]. In total, we obtain 6,027 face tracks (see Table 1), annotated among 30 characters and an additional class representing the unknown background characters. Note that the ground truth annotations of face tracks are used only for evaluation at test time, and not for training the models.

	BBT		Buffy		Lost	
	acc %	# refs	acc %	# refs	acc %	# refs
1st person	16.6	12	37.5	8	15.3	13
2nd person	98.2	116	91.4	235	96.8	219
3rd person	90.9	111	82.1	241	72.3	130

Table 2: Types of name mentions in the three TV series along with their ground truth counts and classification accuracy.

Ground truth name mentions. In order to evaluate our name mention classification, we manually annotated the name utterances along with their person types in the subtitles of all episodes of the three TV series.

Face track descriptors. We consider two recent and successful face track descriptors: (i) VGG face [19], are extracted using a pre-trained very deep CNN model; and (ii) VF^2 (Video Fisher Vector Faces) [18] are state-of-the-art non-deep features.

Note that the VF^2 descriptor aggregates face image representation of the track before applying Fisher encoding. For the VGG face CNN model the track representation is obtained by mean pooling features of every frame.

Characters, named and unknown. We call tracks for characters whose name never appears in the story as *unknown*. The number of such characters is an important factor, as we are unable to train a model for them which results in all of them being misclassified. Lost and BBT have a higher fraction of unknown character face tracks at 9% (*c.f.* Table 1) and 10.6% respectively. In Buffy, about 6.5% of all face tracks are unknown.

We obtain the set of named characters based on the cast list (*e.g.* obtained from IMDb [1]). These names are used to find name mentions in the subtitles.

4.2. Name mention classification

The quality of bag labels hinges on first detecting and classifying the name utterances in the dialogs. In Table 2, we present the classification accuracy of the name mentions based on the grammar rules introduced in Section 3.1.

A 1st person reference can be thought of as most useful since the speaker is associated with the particular name. However, they occur rarely and are often misclassified as second person (*e.g.* "Howard Wolowitz, CalTech Department of Applied Physics."), especially when seen out of context. Nevertheless, since our method uses both 1st and 2nd person references to collect positive bags, the confusion does not critically hurt performance. As is evident from Table 2, the 2nd and 3rd person name mentions are not only more frequent, but also easier to identify leading to a higher classification performance.

Name mentions that do not match any grammar rule are discarded from further processing. Overall, 7.6% of all name mentions are discarded and 5.8% of those which are

	Series	# bags	% correct	# tracks	ρ^+tracks
	Scene level bags				
Positive	BBT$^+$	125	96.0	7240	0.23
	Buffy$^+$	237	89.5	10273	0.24
	Lost$^+$	178	84.8	4093	0.32
Negative	BBT$^-$	73	46.5	4461	0.11
	Buffy$^-$	144	44.4	6395	0.12
	Lost$^-$	86	70.9	2157	0.06
	Shot level bags				
Positive	BBT$^+$	335	59.5	858	0.29
	Buffy$^+$	584	52.9	1575	0.29
	Lost$^+$	467	55.5	880	0.43
Negative	BBT$^-$	297	92.3	732	0.07
	Buffy$^-$	578	83.4	1524	0.09
	Lost$^-$	514	92.9	514	0.06

Table 3: The quality of the created bags using all episodes in our data set. The columns show the number of bags, the percentage of correctly labeled bags, the total number of instances in all bags and the average density of positive instances in the bags.

assigned a type are misclassified.

4.3. Analysis of MIL bags

We now evaluate the quality of bags created by collecting face tracks surrounding the name mention in the dialog. Table 3 shows the results considering four key properties: (i) the number of bags; (ii) the percentage of correctly labeled bags (recall that a positive bag is correct when it contains at least one positive instance, a negative bag is correct when all instances are negative); (iii) the total number of tracks within bags; and (iv) the average density of positive tracks in the bags.

Our approach using *scenes* to create bags is able to obtain a high number of correctly classified positive bags (85% to 96%). While the percentage of correct negative bags is lower, for most MIL algorithms it is sufficient to have the density of positive instances in the positive bags higher than in the negative bags.

Furthermore, the high number of instances contained in the scene-level bags makes it easier to train MIL models. Note that bags for different characters may have overlapping tracks since all tracks within a scene are collected in each bag.

In comparison to scene-bags, the *shot* bags have a slightly higher density of positive instances in the positive bags and a lower density in negative bags. The shot level bags not only have fewer tracks (about 20% compared to scene-level bags) but merely 55% of the positive bags are correct. As we will see later, the shot bags perform worse than scene bags at identifying all tracks.

	Non-iterative		Iterative			
type	NSK	sMIL	stMIL	miSVM	SISL	one-vs-all
BBT (VF^2)	**62.2**	47.9	34.5	54.8	23.7	50.8
BBT (VGG)	**67.4**	56.6	54.6	54.5	61.4	47.9
Buffy (VF^2)	33.9	**34.0**	14.7	31.8	24.8	22.5
Buffy (VGG)	**49.8**	46.9	20.6	23.9	27.6	35.0
Lost (VF^2)	34.3	**35.1**	12.3	26.2	25.8	27.5
Lost (VGG)	40.5	45.9	**46.4**	37.9	34.2	26.3

Table 4: Comparison of the accuracy of labeling all face tracks using different MIL-methods and features. Best results are in bold while the second best are underlined.

4.4. Face tracks labeling

Similar to the subtitle-transcript paradigm [3], we use the instances labeled via MIL to train character SVM models and evaluate the character identification accuracy on all tracks. Table 4 presents the accuracies obtained by different MIL-algorithms from Section 3.2. We see that non-iterative approaches, NSK and sMIL, perform well on all three TV series since they do not assume that the bags are labeled correctly, and the positive instance density is higher in positive bags than in negative bags. This makes them suitable for the type of bags that we create.

Furthermore, we observe in Table 3 (scene-bags) that in comparison to BBT and Buffy, Lost has a higher positive instance density, but a lower percentage of correctly labeled positive bags. Most positive bags in Lost are either correct and contain a very high density of positive instances or do not contain any positive instances (*i.e.* are labeled incorrectly). This feature explains why sMIL outperforms NSK. sMIL learns better classifiers since it does not encourage incorrectly labeled positive bags to contain a high number of positive instances and meanwhile allows correctly labeled bags to have a high number of positive instances.

Id performance using shot-bags. Shot bags produce exhibit lower performance as compared to scene-bags – BBT: 40.8%, Buffy: 33.4% and Lost 38.9% using NSK and VF^2. The worse performance on BBT and Buffy of the shot-level bags is due to the small number of tracks and the lower quality of the positive bags (see Table 3). On Lost, the shot-level bags, despite the small number of tracks, are able to improve the accuracy by 4% using NSK. This improvement may be explained by the fact that NSK works well with positive bags with a high density of positive instances.

The positive shot-level bags have a much higher positive instance density than the scene-level bags (see Table 3). Nevertheless, due to the larger number of tracks and higher overall performance, we believe that scene-level bags are a promising direction to pursue.

Precision-recall curves. In the spirit of the "refusal to pre-

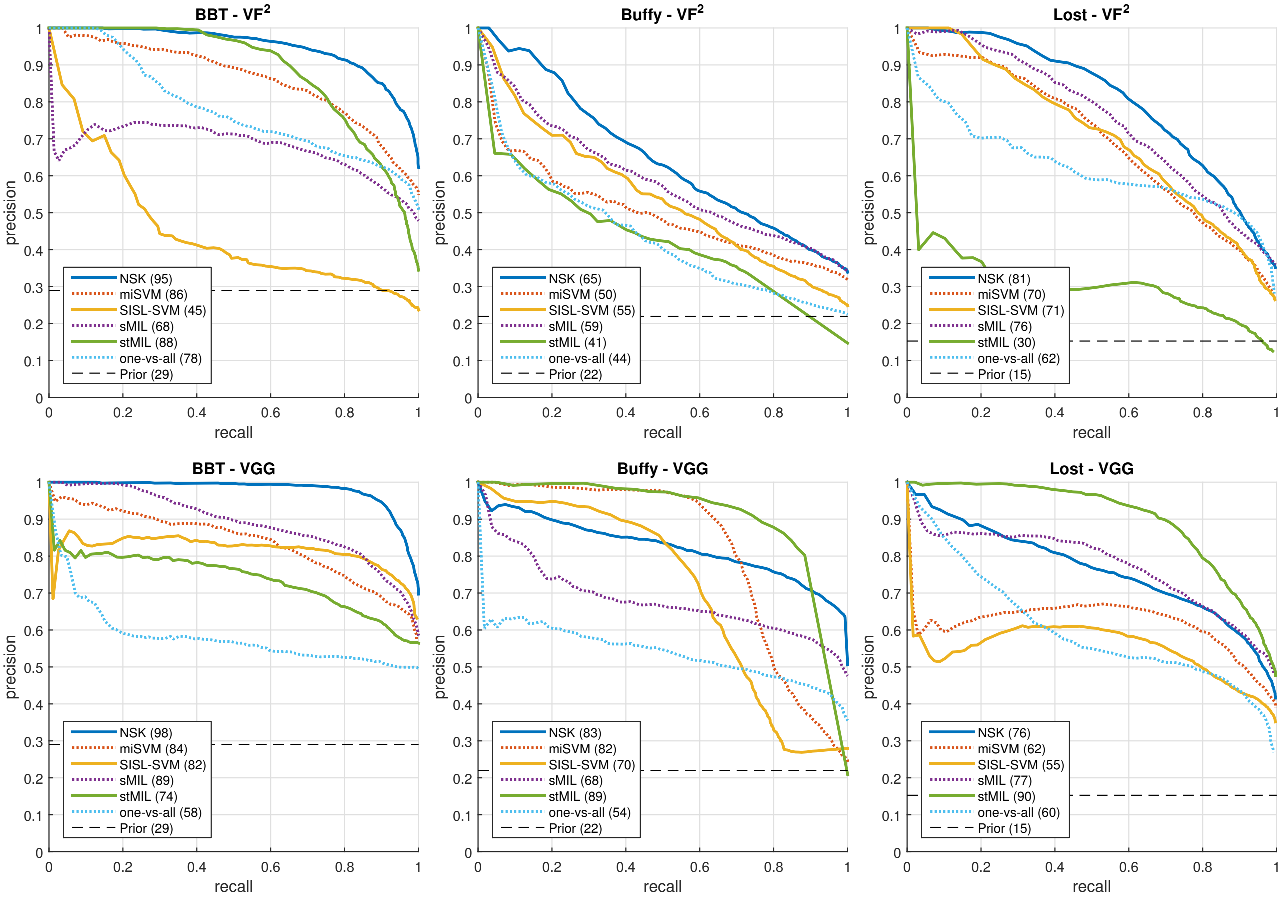

Figure 3: Comparison of different MIL approaches on the three TV series BBT, Buffy and Lost using Video Fisher Vector Faces VF^2 and VGG face descriptors and scene-level bags.

dict" scheme [10], Figure 3 provides a deeper insight on the performance of various MIL methods through precision-recall (PR) curves. On average, NSK performs best in most cases, while sMIL and its iterative transductive version stMIL are close or slightly better in some scenarios.

The PR curves of Lost demonstrate that stMIL is very sensitive to a good initialization. stMIL, initialized by sMIL, iteratively pushes the decision boundary towards low data density regions. When using VF^2 in Lost, stMIL pushes the boundary such that the results become worse, while when using the VGG face descriptor the addition of the transductive constraints are able to improve the results.

Other iterative approaches such as miSVM, SISL-SVM and one-vs-all translate the MIL problem to SISL and are unable to demonstrate good performance as compared to native non-iterative algorithms.

Finally, the prior in Figure 3 exposes the difficulty of the three TV series where all tracks are labeled as the most frequent character.

Comparison to [9]. For a fair comparison with [9], we present the performance of our approach with NSK on the 10 most frequent characters of Lost. We compare to two approaches adopted by [9]: (i) only using face cues; and (ii) with gender and temporal grouping (clustering) information in addition to faces. Our approach is able to outperform both while only using face information. Among the top 10 most frequent characters, we are able to obtain an accuracy close to 70% and an average precision of 93%. We demonstrate that MIL with scene-level bags and improved descriptors is able to outperform an approach that requires a lot of contextual information.

Establishing an upper bound. Classically, subtitles and transcripts have been used to obtain weak labels, which are in turn used to train character-specific face models. We establish an upper bound for our subtitle-only identification scheme through this approach. In particular, we use the improved weak labeling approach described in [29] and train character-specific SVMs using VF^2 descriptors.

Table 5 shows the results of the two identification schemes when only using subtitles (S) and the correspond-

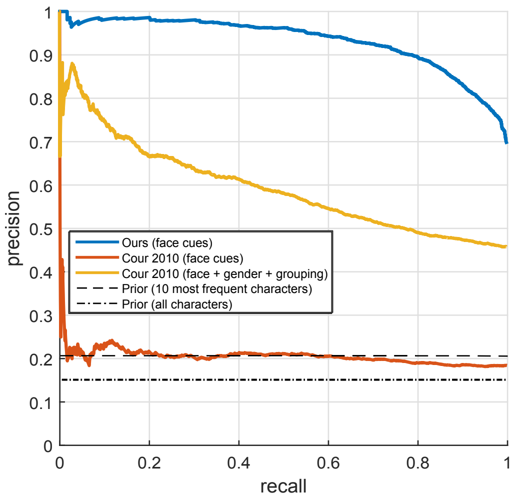

Figure 4: Precision-recall curves for our approach with NSK and VGG face descriptor compared against Cour *et al*. [9]. These results are obtained on the 10 most frequent characters of Lost.

BBT		Buffy		Lost	
S	S+T	S	S+T	S	S+T
62.2	79.1	34.0	80.5	35.1	70.9

Table 5: Accuracies of our approach using solely subtitles (S) compared against an upper bound established by using subtitles and transcripts (S+T).

ing upper bound when using both subtitles and transcripts (S+T). As compared to [9], we come closer to the performance of subtitles and transcripts, however, there is large scope for future work.

5. Conclusion

In this work we revisit the problem of labeling all character appearances in TV series by only using subtitles to obtain character annotation for training. In contrast to subtitles and transcripts, using subtitles alone is a realistic goal for large-scale labeling as transcripts are often hard to find or incomplete. We propose to model the problem via Multiple Instance Learning, and show how MIL bags can be created using name mentions from subtitles. We discuss and evaluate several iterative and non-iterative MIL techniques and show promising identification performance while using this very sparse form of supervision. With this analysis and the additional data set, we wish to encourage the community to look towards this problem thus making person identification methods widely applicable to all forms of TV series.

Acknowledgments This work was supported by the Deutsche Forschungsgemeinschaft (DFG - German Research Foundation) under contract no. STI-598/2-1. The views expressed herein are the authors' responsibility and do not necessarily reflect those of DFG.

References

[1] Internet Movie Database. http://www.imdb.com/. 5

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, 2002. 4

[3] M. Bäuml, M. Tapaswi, and R. Stiefelhagen. Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2, 5, 6

[4] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding Actors and Actions in Movies. In *International Conference on Computer Vision (ICCV)*, 2013. 1, 2

[5] R. C. Bunescu and R. J. Mooney. Multiple Instance Learning for Sparse Positive Bags. In *International Conference on Machine Learning (ICML)*, 2007. 4

[6] C.-H. Chen and R. Chellappa. Character Identification in TV-series via Non-local Cost Aggregation. In *British Machine Vision Conference (BMVC)*, 2015. 2

[7] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *European Conference on Computer Vision (ECCV)*, 2008. 1

[8] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[9] T. Cour, B. Sapp, A. Nagle, and B. Taskar. Talking Pictures: Temporal Grouping and Dialog-Supervised Person Recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2010. 1, 2, 5, 7, 8

[10] M. Everingham, J. Sivic, and A. Zisserman. "Hello ! My name is ... Buffy" – Automatic Naming of Characters in TV Video. In *British Machine Vision Conference (BMVC)*, 2006. 1, 2, 7

[11] B. Fröba and A. Ernst. Face Detection with the Modified Census Transform. In *Automatic Face and Gesture Recognition (FG)*, 2004. 5

[12] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-Instance Kernels. In *International Conference on Machine Learning (ICML)*, 2002. 4

[13] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007. 1

[14] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision (IJCV)*, 29(1):5–28, 1998. 5

[15] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof. Learning to Recognize Faces from Videos and Weakly Related Information Cues. In *IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, 2011. 2

[16] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In *Computer Vision and Pattern Recognition (CVPR)*, 2008. 1

[17] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the Association for Computational Linguistics: System Demonstrations*, 2014. 3

[18] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A Compact and Discriminative Face Track Descriptor. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 5

[19] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. In *British Machine Vision Conference (BMVC)*, 2015. 1, 5

[20] D. Ramanan, S. Baker, and S. Kakade. Leveraging archival video for building face datasets. In *International Conference on Computer Vision (ICCV)*, 2007. 2

[21] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. Linking people in videos with "their" names using coreference resolution. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 2

[22] P. Sankar, C. V. Jawahar, and A. Zisserman. Subtitle-free Movie to Script Alignment. In *British Machine Vision Conference (BMVC)*, 2009. 2

[23] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[24] J. Sivic, M. Everingham, and A. Zisserman. Person Spotting: Video Shot Retrieval for Face Sets. In *International Conference on Image and Video Retrieval (CIVR)*, 2005. 2

[25] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" – Learning person specific classifiers from video. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 1, 2

[26] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[27] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. "Knock! Knock! Who is it?" Probabilistic Person Identification in TV-Series. In *Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[28] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. StoryGraphs: Visualizing Character Interactions as a Timeline. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 4

[29] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Improved Weak Labels using Contextual Cues for Person Identification in Videos. In *International Conference on Automatic Face and Gesture Recognition (FG)*, 2015. 2, 7

[30] L. Wolf, T. Hassner, and I. Maoz. Face Recognition in Unconstrained Videos with Matched Background Similarity. In *Computer Vision and Pattern Recognition (CVPR)*, 2011. 1

[31] X. Xiong and F. D. l. Torre. Supervised Descent Method and its Applications to Face Alignment. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 5