Semantic Video Entity Linking based on Visual Content and Metadata

Yuncheng Li, Xitong Yang, Jiebo Luo
University of Rochester
Department of Computer Science
Rochester, New York 14627, USA
{yli, xyang35, jluo}@cs.rochester.edu

Abstract

Video entity linking, which connects online videos to the related entities in a semantic knowledge base, can enable a wide variety of video based applications including video retrieval and video recommendation. Most existing systems for video entity linking rely on video metadata. In this paper, we propose to exploit video visual content to improve video entity linking. In the proposed framework, videos are first linked to entity candidates using a text-based method. Next, the entity candidates are verified and reranked according to visual content. In order to properly handle large variations in visual content matching, we propose to use Multiple Instance Metric Learning to learn a “set to sequence” metric for this specific matching problem. To evaluate the proposed framework, we collect and annotate 1912 videos crawled from the YouTube open API. Experiment results have shown consistent gains by the proposed framework over several strong baselines.

1. Introduction

Watching online video has become a part of many people’s daily lives. For video hosting websites, it is at the core of their business to increase user engagement. A key problem is how to help people access what they want from the massive amount of videos. Many systems have tried to solve this problem from different angles, e.g., text-based search, categorized video browsing, trending video highlights, related video suggestions, and personalized video recommendation [3, 5, 25–27]. Accurate video content understanding is the key to achieve success for all these applications. Unlike texts and images, semantic video understanding is a more open problem for many reasons. In this paper, we propose a vision-based method to solve a specific video understanding problem: video entity linking.

Video entity linking is the task of connecting videos with the entities in a given database, such as Wikipedia. A wide variety of video-based applications can benefit from video entity linking given that the entities are associated with rich attributes and connected together in a semantic graph. However, video entity linking is a very challenging problem. The reasons are many folds. 1. The set of entities of interests are very diverse, such as people, places, art works and artifacts, and the vocabulary size and visual variations make it infeasible to develop classification based models. 2. Visual content information is only part of the video. For example, the information to link some entities may be con-

Figure 1: The overview of the proposed framework. Given an online video, we first extract entity candidates by linking its title with a knowledge graph. Based on a learned matching metric, the entity candidates are then ranked and verified by matching the video with the entity’s representative images.
Figure 2: An illustrative example for entity matching. The first row contains the key frames extracted from a YouTube video entitled “Jason Taylor career highlights”. From the video title, “Jason Taylor” can be linked to either a football player or a rugby player. The second row contains the representative images for the football player, and the third row contains those for the rugby player. Images bordered by the same colors are visually similar according to the learned metric defined in Eqn. 1. Based on the matched images, the football player is the right entity to link with, but it would be difficult to tell only from the title. This example also shows that the entity occurrence (matched with any of the representative images) is smooth over time, which is modeled by the Temporal Smoothness term in Eqn. 4. Furthermore, although the representative images show different aspects of the entity, there is significant irrelevant information which is reduced by the Representative Smoothness term in Eqn. 5.

Even with its limitations, visual content has its unique merits in entity linking. First, there are many cases where video entity linking is very difficult (requiring multi-hop inferences), if not impossible, without examining the visual content, as shown by the example in Fig. 2. Second, compared with other modalities, the rich and dynamic visual content in videos convey accurate video watching intent, at least for human brains. For a video, this means an entity evidenced by the visual content is often more important than the audio track and associated texts. For example, when people search for the song Burn by Ellie Goulding, a music video is usually of higher value and relevance than a video that merely has the song as its background music. In this case, the visual content represents the user searching intent better than the sound track.

In this paper, we focus on developing effective models to leverage visual content for video entity linking. In order to validate and demonstrate the proposed models, we have developed an automatic video discovery system to collect candidate videos within various challenging domains.

The proposed framework is shown in Fig. 1. As in text domain, there are two stages in video entity linking. First, entity mentions, or surface forms, are detected using Named Entity Recognition (NER). Second, supervised entity disambiguation methods are applied to rank potential entity candidates for each surface form, based on various contexts. We rely on a text-based method to identify surface forms and produce entity candidates from the video metadata, e.g., title and description, thus largely reducing the searching space of video entity linking. Consequently, the proposed framework can be considered as incorporating a vision-based bootstrap process to a text-based retrieval system. Image search engine-based data scraping is then employed to find representative images for the entity candidates, which are matched with the video visual content. In consideration of scalability and computational efficiency, we represent the video content by multiple key frames, which can be treated as an image sequence. In essence, video entity linking is cast as matching an image set to an image sequence.

In this paper, we propose to solve this set to sequence matching problem through metric learning, with which a semantic distance between pairwise images is learned to adaptively measure, for example, similar scenes or sensor layout. While the metric is learned for pairwise images, the supervision labels are given for (set, sequence) pairs,
indicating whether an entity (represented by an image set) matches with the video (represented by an image sequence). Multiple Instance Metric Learning (MIML) is used to handle the label ambiguities within this problem. We further propose variants of MIML to capture the smoothness constraints observed in the visual entity spotting structure.

Video entity linking can be seen as an instance of “video hyperlinking” [21], in which video segments are linked with other relevant content, e.g., another video segments or a Wikipedia page. While these existing systems employ a limited number of concept detectors to link video segments together [2], our objective is different in that we propose an effective method to link videos explicitly with semantic entities, which 1) are significantly more comprehensive and 2) conforms to a rigorously defined semantic hierarchy. Moreover, our proposed entity linking method, based on metric learning and exemplar matching, is fundamentally different from the classification approach taken by the video concept detectors, which have been predominately adopted by the TRECVID community [24].

There are several contributions in this paper:

- We demonstrate how visual content can be used to boost entity linking in the video domain;
- We propose MIML-Struct, a metric learning paradigm for set to sequence matching, to solve the video entity linking problem with structural smoothness;
- We propose a new open source dataset and a video discovery framework for future video entity linking research.

2. Related Work

Entity linking in text domain has attracted many Natural Language Processing research efforts to build accessible knowledge graphs based on free text [7]. Wikify! was proposed in [22] to link documents with Wikipedia pages. Others proposed to match contextual information between Wikipedia pages and a target document to solve entity linking in a principled way [22]. Kulkarni et al. proposed to exploit inter-entity dependency to collectively annotate entities in documents [20]. Milne et al. proposed to use feature engineering and supervised learning to solve entity linking [23]. Recent developments, such as DBPedia Spotlight [7], have made entity linking with multilingual Wikipedia pages accessible to non-experts and scalable to large scale corpus.

In contrast to traditional efforts in text domain, video entity linking based on visual content, as proposed in this paper, has a number of unique characteristics: 1) For computers, video content is much more difficult to understand than text. The visual appearance of an entity is often much more diverse than its text surface forms. This motivates us to use a pool of representative images to model the visual variations. 2) Entity occurrences inside a video sequence usually exhibit temporal smoothness, which is different from free text. This motivates us to propose a structural model to leverage such smoothness. In addition, while most of the previous algorithms work well for long texts in which rich contexts are available, they are not effective for short video titles.

Supervised Metric Learning aims to embed domain knowledge into an adaptive Mahalanobis metric. There is a wide range of applications for Metric Learning [31], e.g., k-NN classification [8][28], clustering with side information [50], and domain adaptation [10][19]. Various Metric Learning algorithms have been proposed from different perspectives. In [30], a Semi-Definite Programming formulation was proposed to learn a Mahalanobis metric to draw similar data instances close while preserving distance between dissimilar instances. Davis et al. proposed an information theoretic approach to push the distances between similar instances under a bound and those between dissimilar instances above a bound [8]. Weinberger et al. proposed a max margin formulation to draw instances of the same class together while keeping instances of different classes apart with a margin [28]. In [11], LDML was proposed to perform face verification, in which a logistic discriminant is learnt to predict matches of instance pairs. In this paper, the core of the proposed entity linking framework is an image set to sequence matching problem. Since there are large appearance variations and severe domain mismatches, we propose a Supervised Metric Learning method to learn an adaptive metric to measure the relevance between representative images and video frames.

Multiple Instance Learning is a weakly supervised learning paradigm, first introduced in [9] to learn predictive models from bag-level labels. Since then, there are extensive developments around the MIL scheme, among which two directions are most relevant to this paper. MildML [12] was proposed to perform face verification using automatic text based annotation, in which the Multiple Instance Learning paradigm is applied. From another perspective, a structured prediction model was proposed for interactive segmentation [29]. This work represents an effort to explore structures among instances [32] or within instances [4]. At a high level, the proposed video entity linking model is a combination of these two directions, i.e., Multiple Instance Metric Learning with Structured constraints (we refer to it as MIML-Struct). We differentiate from previous works by formulating the video entity linking problem within the MIML framework and exploring the unique structural information arising from this problem. The difference between [12] and our work is that we explore structural information in the Metric Learning process, and we also employ a smoother bag aggregation function to accelerate the optimization process. The difference between [29] and our
work is that we propose different structure regularization specialized for the task of video entity linking. Moreover, we apply Multiple Instance Learning to learn a semantic metric, instead of a classifier.

3. Proposed Models

In this section, we formulate the video entity linking problem within the Metric Learning framework. Image set and sequence matching is the core problem in our video entity linking framework. In the proposed framework, training data is composed of annotated pairs of image set and frame sequence, both of which are represented by a set of vectors, often called a bag.

3.1. Problem Formulation

Let us denote a key frame sequence as \( F_I = (x_{i1}, x_{i2}, \ldots, x_i) \), in which \( x_{id} \) denotes the \( d \) dimensional image features and the bar above \( I \) denotes length of the sequence. Denote a representative image set as \( R_J = \{x_{j1}, x_{j2}, \ldots, x_j\} \). While \( R_J \) is an unordered set, \( F_I \) is ordered in the temporal domain. In the training data, each pair of \( F_I \) and \( R_J \) is manually annotated as \( iJ \in \{0, 1\} \), indicating whether the entity represented by \( R_J \) can be identified in the video represented by \( F_I \). Given labels \( iJ \), the goal of the proposed model is to learn a Mahalanobis distance metric between set-to-sequence pairs to perform video entity linking. The matching process is illustrated in Fig. 2.

The Mahalanobis distance is defined as a symmetric positive semidefinite matrix \( M \in R^{d \times d} \). In order to reduce the number of parameters, it is beneficial to learn a low rank \( M \), and define \( M = L^T L \), in which \( L \in R^{k \times d} \), so that rank of \( M \) is \( k \). Similar to [12], the Mahalanobis distance is defined as follows using \( L \),

\[
d_L(x_i, x_j) = (x_i - x_j)^T L^T L(x_i - x_j)
\]

Given the training data \( (F_I, R_J, iJ) \), the goal is to learn a metric \( L \) to match new pairs. We first formulate the basic MIML model and then extend to MIML-Struct to incorporate structural information encoded in the video key frame sequence \( F_I \).

3.2. Multiple Instance Metric Learning

The challenge of MIML is that training labels are given at the bag level, while distance is measured at the instance level. We first define the bag level distance by averaging the distances between top matched instance level pairs. Formally, we first define the set of top matched pairs \( s_U \) by,

\[
s_U = \{(i, j) | i \in I, j \in J, r(d_{ij}) \leq \kappa\}
\]

in which \( r(d_{ij}) \) is the rank of \( d_{ij} \) in the ascending sorted list of all pairwise distances \( D_U \triangleq \{d_{ij} | i \in I, j \in J\} \), and \( \kappa \) is a fraction of the size of \( D_U \). With the definition of \( s_U \), we define the bag level distance by,

\[
d_U = \frac{1}{|s_U|} \sum_{(i, j) \in s_U} d_{ij}
\]

Similar to [12], we then map the bag level distance to a matching probability by \( P_U = \sigma(b - d_U) \), in which \( \sigma(z) \) is the sigmoid function \( \sigma(z) = 1/(1 + \text{exp}(-z)) \) and \( b \) is a bias term. We maximize the matching probability for positive bags \( I^+J \triangleq \{(I, J) | t_U = 1\} \) by \( L_+ \),

\[
L_+ = -\frac{1}{|I^+J|} \sum_{I^+J} \log(P_U)
\]

There are no ambiguities inside a negative bag, thus we define the matching probability at the instance level as \( p_{ij} = \sigma(b - d_{ij}) \). We minimize this matching probability by \( L_- \).

\[
L_- = -\frac{1}{|I^+J|} \sum_{I^+J} \frac{1}{|I||J|} \sum_{i \in I, j \in J} \log(1 - p_{ij})
\]

3.3. Structured Multiple Instance Metric Learning

In order to suppress noise introduced by the label ambiguities and multiple representative images, we propose a novel structured multiple instance metric learning to model temporal and representative smoothness.

As shown in Fig. 2, the entity spotting structure exhibits strong temporal structures, which is modeled through a regularization term to enforce temporal smoothness,

\[
L_{TS} = \frac{1}{2 |I^+J|} \sum_{I^+J} \frac{1}{|I||J|} \sum_{i \in I, j \in J} (d_{ij}^l - d_{ij}^l)^2,
\]

where \( d_{ij}^l \) is the aggregated distance between the key frames \( x_i \in F_I \) and the pool of representative images \( R_J \), defined similarly to \( d_U \). In particular, we first define the top matched pairs \( s_G \) as,

\[
s_G = \{j | j \in J, r(d_{ij}) \leq \kappa'\}
\]

where \( r(d_{ij}) \) and \( \kappa' \) are defined as above. Then \( d_{ij}^l \) is defined as follows,

\[
d_{ij}^l = \frac{1}{|s_G|} \sum_{j \in s_G} d_{ij}
\]

In order to model the diverse visual appearance of an entity, we employ multiple representative images for an entity. However, these entity representative images can also introduce misleading information or noise as shown in Fig. 2. In order to suppress such noise, we encourage the metric to capture the shared semantics among the representative
images by graph regularization. Formally, for a key frame 
\( x_i \), all the distances with the representative images, i.e., 
\( \{d_{ij} | j \in J \} \) should be close to each other, as expressed 
by the following regularization term,

\[
\mathcal{L}_{RS} = \frac{1}{|U|} \sum_{i \in U} \sum_{j \in J} \sum_{l \in J, j < l} (d_{ij} - d_{il})^2 / |J|(|J| - 1)
\]

We refer to the regularization term defined in Eq. (4) as
Temporal Smoothness (TS), and the one defined in Eq. (5) as
Representative Smoothness (RS). Note that both constraints
are defined only for the positive bags.

3.4. Optimization

Combining the negative cross entropy objective term and
the regularization terms, the final objective is defined as

\[
\mathcal{L}(L, b) = \mathcal{L}_+ + \mathcal{L}_- + \lambda_1 \mathcal{L}_{TS} + \lambda_2 \mathcal{L}_{RS},
\]

which is a smooth convex function that can be optimized using
gradient descent algorithms, such as L-BFGS. See Appendix A for the gradients of each term.

4. Experiments

In this section, we first validate our MIML formulation
in a well-known face verification task. Second, we apply
MIML-Struct in the video entity linking problem and validate
the effectiveness of each model component by comparison experiments.

4.1. Face Verification

The idea of Multiple Instance Metric Learning (non-
structural variant) was first proposed in [12] for an automatic
face annotation setup. In this setup, each photo is a bag of faces and names, but associations between faces
and names are unknown. In order to learn a semantic metric
for face verification, pairwise photos are assigned with
a label indicating whether they contain faces of the same
person. For example, in Fig. 3 there are two faces in each
photo, and the face set in Fig. 3a is matched with those in
Fig. 3b but the face set in Fig. 3b is not matched with those in
Fig. 3c. As it is in our video entity linking problem, the
supervision is given only in bag/set level. However, there
are many differences between the face verification scenario
and our video entity linking task. For example, although
the representative images and key frames can be seen as
the counterpart of faces, there is no notion of names in our
video entity linking problem. In addition, the number of instances in one bag is much larger in the video entity linking
than in the face verification, which worsen the label ambiguity problem and increase the level of noises. We apply
the unstructured version of MIML for the face verification

4.2. Video Entity Linking

In this section, we present our experiment results for
video entity linking. We first describe how we collect data
for evaluation. Second, we explain the baselines used to validate our assumptions and models, and the evaluation metrics for video entity linking. The comparison results are reported at the end.

4.2.1 Data Collection

There are multiple phases in the data collection process,
and the goal is to discover videos that can potentially benefit from the proposed visual entity linking framework. As mentioned in the introduction, the proposed framework can be seen as a bootstrap step for a text based system and given that the text based system already performs well using various context information, it is important to carefully choose a subset of the entire video corpus to demonstrate the advantages of using visual content. In addition, it is impossible for us to scan through all YouTube videos, so we have to rely on the YouTube video search API to discover videos, which also forces us to design the data collection process carefully. The main video discovery components are explained below. All data has been collected using the public search API from Google and Bing, and can be downloaded at.

\[\text{DBPedia} \text{ is a crowd-sourced knowledge base with structured information extracted from Wikipedia. Each} \]

https://www.google.com/cse/all
http://www.bing.com/developers/
https://goo.gl/IemjVv
Entity category targeting selects a subset of entities according to its category, because we cannot handle all possible entities in the wild. There are several considerations in selecting the entity categories: (1) Entities in the category should be consistent with the visual content. For example, if the Movie entity “Life of Pi” occurs in the title, the movie content will show up in the video. However, this is not the case for MovieDirector entity “Ang Lee” who has directed many movies. (2) The category is popular on YouTube, otherwise it is hard to get example videos by keyword search. (3) The pool of categories should be large enough to cover as many videos as possible. By manually screening through all 500+ entity categories, we select 7 of them for our experiments, including SportsTeam, Athlete, Film, Musical, Single, TelevisionShow, and Automobile. The per category dataset statistics are shown in Table 3.

Video search query selection is used to select queries discover candidate videos from the video search engine. The criteria to select query terms are: (1) The query should carry some ambiguities, so that they can refer to multiple entities. This is because we believe visual content will perform better than text in the disambiguation task. Ambiguous terms and what they may refer to can be found on Wikipedia disambiguation page. (2) The query term should refer to some entities in our selected categories. (3) In order to highlight the benefits of visual based method, we remove queries that have dominant entities in the experiment. For example, although the term “apple” may refer to “Apple Inc.” or “Apple Tree”, the entity “Apple Inc” has dominant popularity on YouTube. The entity popularity is approximated by the Google PageRank of the Wikipedia page for this entity. Note that this constraint is enforced so that the dominant entities do not overwhelm the “long-tail” or less dominant entities. On the other hand, the dominant entities will not suffer in practice due to their high popularity priors. After the filtering, 1500 search terms are sampled for further process. Examples are “The Outsiders”, “Tony Smith”, “Soul Man”, “American Pie”, and “Just Like Heaven”.

Video harvesting and entity annotation occurs when we use the selected terms to query Google video search, generate candidate entities from video title and extract about 30 key frames from each video. By filtering out videos that do not contain any entities from the selected categories, we collect 1912 videos in total. From the video titles, we detect about 60K entity mentions using DBpedia Spotlight, which are linked to about 13K unique entities among which 2113 are in our selected categories. By concatenating the entity name and its category label, we formulate an image search query to obtain the entity’s representative images through the Bing image search API. Note that in real use cases, this step (associating entity with representative images) can be done offline using various techniques, but that is out of scope of this paper. In order to attain matching ground truth, we employ Amazon Mechanical Turk (AMT) to recruit workers to annotate each entity link. Along with detailed instructions, AMT workers are asked the following question, “In the video, can you find the entity represented by the given images?”. This is a well-defined image matching question and easy for workers, so we expect a high agreement among workers, which is verified in Table 1. We take the majority vote as the ground truth label for the entity annotation.

Image visual features is used to characterize entity representative images and video key frames. We adopt the features learned from the Convolutional Deep Neural Nets for the ImageNet challenge and use the Caffe implementation. We use the fc7 layer in the default network architecture as visual features (4096D). This feature is widely used in the research community, because of its extraction efficiency, representation power and effectiveness for a wide range of image based applications.

The number of videos and entities per category are shown in Table 3. For the final dataset, the average number of potential entities per video is 4.32, the average number of ambiguous terms associated with a video is 2.24 and the average number of potential entities per terms is 2.73.

### 4.2.2 Experimental Protocols and Metrics

Because the proposed video entity linking framework is based on refining text based results, there are naturally two experiment protocols, disambiguation and verification.

Disambiguation aims to use the learned Mahalanobis metric to rank the entity candidates generated from the text based system. For this task, we use precision@1 (prec@1) and recall@1 as the evaluation metrics.

Verification aims to verify whether an entity candidate generated from the text based system is actually related to the video visual content. In contrast to the disambiguation task, verification will attempt to reject the output from the
Table 2: Experiment Results. As explained in the main text, the disambiguation task performance is bounded by a low upper bound and different methods are about the same level, which prove the importance of the verification task. The verification task validates the effectiveness of the proposed models.

text based method. We argue that each task has its own significance and utility in practice. Disambiguation is more conservative and it should perform at least as well as the text based method, while because only part of the information is observed by the vision based system, verification is riskier and only proper in cases in which one cares more about the visual information. These two protocols can be combined in a real system, such that the verification module will first try to reject some errors from the text based system, and the disambiguation module will then pick the best entity from the remaining candidates. The verification task can be seen as a binary classification problem: classify a video-entity pair as a match or non-match, so we employ Average Precision (AP) as the evaluation metric.

We divide the dataset into separate parts for two protocols. For all the videos queried from the video search engine using the sampled 1500 ambiguous terms, we select a subset of 564 videos from 500 terms for the verification task and the rest of 1348 videos for the disambiguation task.

4.2.3 Approaches for Comparison

In order to validate the effectiveness of the proposed models, we consider the L2 distance in both the original feature space and PCA subspace, and a sparse reconstruction based method as the baseline metrics to perform fair comparison. The sparse reconstruction based method assumes that if a video-entity pair matches, the representative images $R_f$ should reconstruct the key frames sparsely $F_f$ with small errors, so the sparse reconstruction cost (SRC) can be used to measure video-entity affinity. SRC is defined as follows [6].

$$
\text{SRC} \triangleq \min_B \|F_f - R_f B\|^2 + \lambda|B|_{2,1},
$$

where $B$ is the reconstruction coefficients. The $l_{2,1}$ norm, defined as the sum of L2 norm of rows, is widely used by computer vision researchers to enforce group sparsity in the parameter space. Here, $l_{2,1}$ norm achieves sparse selection of representative images to reconstruct the key frames. We denote the baselines as L2, PCA and SRC, respectively. We also devise several variants of the proposed models to evaluate its individual components, including

- **MIML**. Fixing both $\lambda_1$ and $\lambda_2$ to 0, we evaluate the model without structural constraints.
- **MIML-TS**. Fixing $\lambda_2$ to 0, we evaluate the Temporal Smoothness constraints.
- **MIML-RS**. Fixing $\lambda_1$ to 0, we evaluate the Representative Smoothness constraints.
- **MIML-Struct**. We tune both $\lambda_1$ and $\lambda_2$ to evaluate the full model.

For L2, PCA and MIML-*, the matching score of a video-entity pair is aggregated from all the pairs of key frames and representative images. There are many different schemes for this purpose in data mining, referred to as linkage functions. For example, a single linkage is the shortest distance between sets. The linkage functions we use include single, complete, average, centroid and medoid. Detailed formulas can be found in [13]. For each matching method, we report the best results over all linkage functions.

In Table 2 there is another row “GT” showing the upper bound these methods can achieve. The disambiguation task does not deal with the false alarms from the text-based system (none of the candidates are correct matches), so the upper bound is less than 100%. Note that the recall@1 can be computed by dividing prec@1 by its upper bound.

4.2.4 Experiment Results

Comparison results are shown in Table 2. Models are trained on 2/3 of the entire dataset and tested on the rest. Hyperparameters $\lambda_1$, $\lambda_2$ as well as $\lambda$ in SRC are tuned on 1/5 of the training samples. For other parameters in Eqn. (2) and (4), we fix $\kappa$ and $\kappa'$ as 5% of the length of the ranking list.

In the results, all of the visual based methods can perform very closely to the upper bound in the disambiguation task. However, the upper bound 35.89%, limited by how well the text based system can do, is not acceptable, which indicates that there are many mistakes made by the text based detection system, and that the verification task is very important to remove such false alarms. In the results of the verification task, the proposed MIML based methods outperform the other methods. MIML based methods outperform naive L2 and PCA methods significantly, indicating the importance to learn an adaptive Mahalanobis
The time complexity to compute gradient for \( L_0 \) is \( O(d^2 F_1 R J) \), and for \( L_1 \) is \( O(d^2 F_1 R J^2) \), which are linear to the number of entity candidates. We implement the optimization algorithm with MATLAB, and on average, it takes 2 hours to learn a metric for the full MIML-Struct model.

### 4.2.5 Category-wise Results

For the verification task, we also report the per-category result in the Table 3 which shows that the proposed MIML-Struct method is more stable than the alternatives. Table 3 also shows large performance variations across different categories. The performance largely depends on the sample size and relevance of the visual background context. For example, different matching fields help to recognize different Athlete entities, while the settings help little to recognize TelevisionShow entities. For the categories requiring deeper understanding the video, the video entity linking is still very challenging.

<table>
<thead>
<tr>
<th>Category</th>
<th>AP(%)</th>
<th>MIML</th>
<th>MIML-TS</th>
<th>MIML-RS</th>
<th>MIML-Struct</th>
<th>#Videos</th>
<th>#Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Athlete</td>
<td>66.44</td>
<td>74.36</td>
<td>80.82</td>
<td>81.84</td>
<td></td>
<td>507</td>
<td>253</td>
</tr>
<tr>
<td>Automobile</td>
<td>50.28</td>
<td><strong>63.40</strong></td>
<td>36.52</td>
<td>49.34</td>
<td></td>
<td>45</td>
<td>39</td>
</tr>
<tr>
<td>Film</td>
<td>46.32</td>
<td>66.91</td>
<td><strong>75.93</strong></td>
<td>63.69</td>
<td></td>
<td>753</td>
<td>380</td>
</tr>
<tr>
<td>Musical</td>
<td>46.03</td>
<td><strong>46.11</strong></td>
<td>34.44</td>
<td>31.39</td>
<td></td>
<td>13</td>
<td>32</td>
</tr>
<tr>
<td>Single</td>
<td>44.34</td>
<td>66.09</td>
<td>62.91</td>
<td><strong>72.78</strong></td>
<td></td>
<td>277</td>
<td>369</td>
</tr>
<tr>
<td>SportsTeam</td>
<td>56.41</td>
<td>67.99</td>
<td>72.38</td>
<td><strong>79.02</strong></td>
<td></td>
<td>72</td>
<td>87</td>
</tr>
<tr>
<td>TelevisionShow</td>
<td>71.12</td>
<td>65.12</td>
<td>60.43</td>
<td>61.28</td>
<td></td>
<td>245</td>
<td>209</td>
</tr>
<tr>
<td>Overall</td>
<td>53.77</td>
<td>60.65</td>
<td>61.22</td>
<td><strong>64.59</strong></td>
<td></td>
<td>1912</td>
<td>1369</td>
</tr>
</tbody>
</table>

Table 3: Category wise performance of the verification task and the dataset statistics for each category. The overall AP is calculated from the entire dataset, ignoring the categories, rather than the average of the category wise AP.

4.2.6 Run Time Complexity

The most time consuming part is the model training part. The time complexity to compute gradient for \( L_0 \) and \( L_1 \) are \( O(d^2 F_1 R J) \), and for \( L_2 \) is \( O(d^2 F_1 R J^2) \), which are linear to the number of entity candidates. We implement the optimization algorithm with MATLAB, and on average, it takes 2 hours to learn a metric for the full MIML-Struct model.

### 5. Conclusions

We have proposed a Multiple Instance Metric Learning framework to solve the challenging video entity linking problem which is formulated as set-to-set image matching. Structured constraints, i.e., Temporal Smoothness and Representative Smoothness, are modeled as regularization terms in the MIML formulation. The experiments on a large annotated Youtube video set have demonstrated the effectiveness of the proposed model. Both the video entity linking problem and the proposed learning paradigm MIML-Struct contribute to the structured learning research. Our future investigation include (1) combining text, audio and visual context to perform entity linking in videos; (2) building entity augmented video applications, such as recommendation and browsing.

## Appendix A

### Gradients of Eqn. (6)

To list the gradients, we first define \( X_{ij} \) by

\[
X_{ij} = (x_i - x_j) (x_i - x_j)^T \in \mathbb{R}^{d \times d},
\]

then the gradients are computed as follows,

\[
\frac{\partial L_+}{\partial L} = \frac{2L}{|U^+|} \sum_{u} \frac{1 - P_u}{|s_u|} \sum_{s_u} X_{ij}
\]

\[
\frac{\partial L_+}{\partial b} = -\frac{1}{|U^+|} \sum_{u} (1 - P_u)
\]

\[
\frac{\partial L_-}{\partial L} = \frac{2L}{|U^-|} \sum_{u^-} \sum_{i \in I, j \in J} \frac{-p_{ij} X_{ij}}{|J||J|}
\]

\[
\frac{\partial L_-}{\partial b} = \frac{1}{|U^-|} \sum_{u^-} \sum_{i \in I, j \in J} \frac{p_{ij}}{|J||J|}
\]

\[
\frac{\partial L_{TS}}{\partial L} = \frac{2L}{|U^+|} \sum_{u} \sum_{i \in I} \frac{2d_i - d_{i-1} - d_{i+1}}{|s_u|(|J| - 1)} \sum_{j \in s_u} X_{ij}
\]

\[
\frac{\partial L_{RS}}{\partial L} = \frac{4L}{|U^+|} \sum_{u} \sum_{i \in I, j \in J} \frac{X_{ij} \sum_{j \in J} (d_{ij} - d_{il})}{|I||J|(|J| - 1)}
\]
References


