Enriching Media Fragments with Named Entities for Video Classification

Yunjia Li
University of Southampton
Southampton, UK
yl2@ecs.soton.ac.uk

Giuseppe Rizzo
EURECOM
Biot, France
giuseppe.rizzo@eurecom.fr

José Luis Redondo García
EURECOM
Biot, France
redondo@eurecom.fr

Raphaël Troncy
EURECOM
Biot, France
raphael.troncy@eurecom.fr

Mike Wald
University of Southampton
Southampton, UK
mw@ecs.soton.ac.uk

Gary Wills
University of Southampton
Southampton, UK
gbw@ecs.soton.ac.uk

ABSTRACT

With the steady increase of videos published on media sharing platforms such as Dailymotion and YouTube, more and more efforts are spent to automatically annotate and organize these videos. In this paper, we propose a framework for classifying video items using both textual features such as named entities extracted from subtitles, and temporal features such as the duration of the media fragments where particular entities are spotted. We implement four automatic machine learning algorithms for multiclass classification problems, namely Logistic Regression (LG), K-Nearest Neighbour (KNN), Naive Bayes (NB) and Support Vector Machine (SVM). We study the temporal distribution patterns of named entities extracted from 805 Dailymotion videos. The results show that the best performance using the entity distribution is obtained with KNN (overall accuracy of 46.58%) while the best performance using the temporal distribution of named entities for each type is obtained with SVM (overall accuracy of 43.60%). We conclude that this approach is promising for automatically classifying online videos.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

Keywords

Media Fragment, Video Classification, Media Annotation, Named Entity Extraction, Concept Extraction, NERD

1. INTRODUCTION

The amount of videos shared on the Web is constantly increasing. Recently, the Media Fragment URI 1.0 (basic)[1] and the Ontology for Media Resource[2] specifications have been standardized by W3C in order to enable linking video segments to structured annotations. These specifications have opened new possibilities for innovative video classification services based on the semantic description of the video content and its media fragments. However, extracting structured content from video at a fine grained level for video classification is not yet a common practice.

Video subtitles (or timed text) are an ideal textual resource for getting insights of the content. Katsiouli *et al.* have applied named entity recognition techniques on video subtitles together with domain ontologies in order to improve video classification [6]. This work pointed out as a future work that video segments could be used together with named entities to improve the classification results. This approach is described in this paper where we aim to best combine named entities and media fragments for providing a video classification framework.

The contribution of this paper is three folds: *1)* a framework for linking media fragments to the Linked Open Data Cloud (LOD) using named entities extracted from subtitles as a follow up of our previous work [8]. In particular, we propose a novel RDF model for the integration of video metadata, media fragments and named entities, that could be reused for various online media; *2)* interesting insights regarding the named entity distribution along the timeline of videos for a subset of Dailymotion channels; *3)* a video classification framework using this named entity distribution and other temporal features sampled from media fragments.

We evaluate our framework by designing an experiment that exploits the number of named entities extracted from video subtitles, their type and their appearance in the timeline as features for classifying videos into different categories. We have randomly selected 805 videos with subtitles from Dailymotion, coming from different channels (or categories) and we have extracted named entities using the NERD framework [12]. We implement four basic machine learning algorithms for multiclass classification problems namely: Logistic Regression (LG), K-Nearest Neighbour (KNN), Naive Bayes (NB) and Support Vector Machine (SVM). For each approach, we compute the overall classification accuracy as well as the precision, recall and F1-score for each channel, and for each algorithm-experiment pair. The research questions addressed by this experiment are:

[1] `http://www.w3.org/TR/media-frags`
[2] `http://www.w3.org/ns/ma-ont`

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

1. is the number of named entities for each NERD type correlated with a channel?

2. is the total number of named entities across the different temporal groups correlated with the channels?

3. are the number of named entities for every NERD type and temporal group correlated whith the channels?

4. which machine learning algorithm(s) can best find correlations allowing us to predict what is the category of a video?

The reminder of the paper is organized as follows. Section 2 presents some related work, while Section 3 details our implemented framework for extracting video metadata, generating media fragments, and linking them to the LOD cloud using NERD. Section 4 describes the dataset and the data model used for the experiment, showing the number, types, and temporal distribution of named entities in the video items taken under investigation. Section 5 presents our evaluation methodology and Section 6 discusses the results of this experiment. Finally, we conclude and propose some future work in Section 7.

2. RELATED WORK

Media Fragment URI 1.0 (basic) is a W3C recommendation that defines the syntax to address temporal and spatial fragments of a multimedia resource directly in the URI, and the guidelines for handling those URIs over the HTTP protocol. Several systems have been proposed to annotate media fragments with LOD cloud resources. The LEMO multimedia annotation framework provides a unified model to annotate media fragments while the annotations are enriched with contextual relevant information from the LOD cloud [4]. Yovisto [15] provides both automatic video annotations based on video analysis and collaborative user-generated annotations which are further linked to entities in the LOD cloud with the objective to improve the searchability of videos. Synote [9] has proposed an RDF model to link media fragments with user generated content. SemWebVid automatically generates RDF video descriptions using their closed captions [14]. The captions are annotated by third-party web services such named entity extractors. The EU NoTube project used semantic web technologies to link TV channels' data with LOD resources [13]. The semantic data was used to further exploit the complex relations between users' interests and the background information of TV programs. Regarding of all these previous works, to the best of our knowledge, no attempts have been made to analyze the characteristic of the temporal distribution of named entities based on the annotated media fragments.

The automatic video classification has usually been treated as a supervised classification task using lower-level features from multimedia analysis or higher-level textual features. A survey on automatic video classification show that there are mainly three modalities being used in the classification: text, audio and visual [2]. As an important attribute for videos the temporal feature is applied in many algorithms. Hence, Niebles *et al.* used temporal correlations in videos to detect audio-visual patterns for classifying concepts [11].

The users' watching behavior and his social interactions are rarely used for improving the accuracy of video classifications. YouTube co-watch data is used for training in [16].

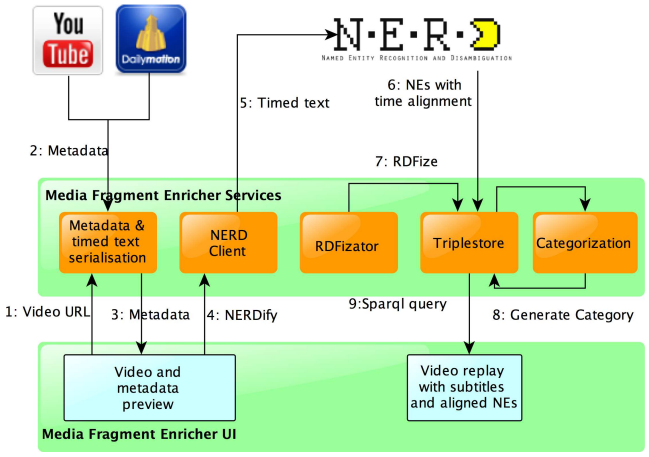

Figure 1: Architecture diagram composed of modules for metadata extraction, name entity recognition and disambiguation, media fragment creation and visualization

The results demonstrate that the proposed method has superior performance when there is not enough manually labeled data available. Filippova *et al.* categorized YouTube videos based on textual information, especially user-generated comments [3]. Subtitles are also identified as an important resource to provide new features for video classifications on the Web [5]. Katsiouli *et al.* have explored an unsupervised approach for semantic video classification by analyzing subtitles [6]. They used WordNet [1] as the external knowledge sources for named entity disambiguation and they also suggested that the "subtitles of each segment can be processed with the support of domain ontologies" in order to improve the classification results. Our work considers the previous literature on using subtitles, named entities and media fragments for video classification, but we extend them by exploiting new features under the notion of linked media.

3. FRAMEWORK ARCHITECTURE

Figure 1 shows the modular implementation of our proposed framework. In a nutshell, the framework enables to retrieve video metadata and subtitles from the video sharing platforms, to extract named entities from timed text, to model the resulting semantic metadata in RDF, and it provides a user interface available at `http://linkedtv.eurecom.fr/nerdviewer` that supports browsing in enriched hypervideos.

The workflow is as follow. First, a viewer enters the URI of a video page from one of the supported media sharing platforms (YouTube and Dailymotion). The *Metadata and Time Text Serialization* module retrieves the media resource and its associated metadata (title, description, statistics about its popularity and subtitles). Second, the viewer launches the annotation process using a *NERD Client* that will extract named entities from the video subtitles. These annotations are then serialized in RDF by the *RDFizator* module that creates media fragments for each subtitle block in which entities have been spotted. The model used is the LinkedTV Ontology[3]. The resulting RDF graph is stored in a *Triple-*

[3]`http://data.linkedtv.eu/ontologies/core`

store and the *Categorization* module uses it for performing the video classification process. Finally the video, subtitles and the named entities extracted are pulled together for the viewer that can interact with the content and get additional information coming from the LOD cloud. In the following subsections, we further detail those processes.

3.1 Metadata Retrieval

The video platforms we support have made available media items and their related metadata via APIs. We have aligned manually their respective data model in a common schema. This information is both depicted to the user in the final interface and further processed by the *Metadata and Time Text Serialization* module. Metadata is important since it contains general information about the video being retrieved such as: title, description, tags, channel, category, duration, language, creation date, publication date, view, comment, favorites and rankings and subtitles. Those elements will be key for the classification process.

3.2 Named Entity Extraction

We perform named entity recognition using the NERD framework. A multilingual entity extraction is performed over the video subtitles and the result is a collection of entities attached to each video. The entities are classified using the core NERD Ontology v0.5[4]. The extraction result is serialized in JSON which is picked up by the *RDFizator* module that will consider the named entities as temporal anchors for creating the annotated media fragments.

3.3 RDF Generation

The *Triplestore* contains RDF descriptions of video annotations. The general metadata that is already published by the video content providers is not included in the RDF graph in order to avoid data duplication. Powered by the LinkedTV Ontology, the video content is annotated at different degrees of granularity using the Media Fragments URI 1.0 specification for addressing segments of this content. Hence, the instances of the `ma:MediaFragment` class are the anchors where entities are attached. The media fragment generation introduces also a very important level of abstraction that opens many possibilities when annotating certain parts of videos. The underlying annotation model relies on well-known ontologies such as the The Open Annotation Core Data Model[5], the Ontology for Media Resources[6], the NERD ontology, and the Programmes Ontology[7]. Figure 2 shows an example of a `ma:MediaFragment` instance. The entity labeled as `Neuhardenber` and classified as `nerd:Location` is attached to the media fragment through *oa:annotation*. The media fragment is associated to a subtitle block using the `linkedtv:hasSubtitle` property. Both the entity label and the subtitle block are serialized according to the NIF Specification[8].

The Turtle serialization of the example provided in Figure 2 follows below. For this excerpt and for the following ones, the instance URIs are automatically created. The temporal references are encoded using the NinSuna Ontology[9]. Finally, the relationship that a `ma:MediaFragment` belongs to an entire video is modeled with the property `ma:isFragmentOf`.

```
<http://data.linkedtv.eu/media/e2899e7f-67c1-4a08
    -9146-5a205f6de457#t=1563.56,1566.8>
  a nsa:TemporalFragment , ma:MediaFragment ;
  linkedtv:hasSubtitle <http://data.linkedtv.eu/text
      /1ca03938-c7ae-4311-a6ed-0540152b651a> ;
  nsa:temporalEnd "1563.56"^^xsd:float ;
  nsa:temporalStart "1566.8"^^xsd:float ;
  nsa:temporalUnit "npt" ;
  ma:isFragmentOf <http://data.linkedtv.eu/media/
      e2899e7f-67c1-4a08-9146-5a205f6de457>.
```

The entity `Neuhardenber` is further described using Dublin Core[10] and LinkedTV properties in order to specify the entity label, the confidence and relevance scores of the extraction, the name of the extractor used in the process, the entity type and a disambiguation URI for the entity that will generally point out to a LOD resource.

```
<http://data.linkedtv.eu/entity/9f5f6bc5-fa3a-4de1-
    b298-2ef364eab29e>
  a nerd:Location , linkedtv:Entity ;
  rdfs:label "Neuhardenber" ;
  linkedtv:hasConfidence "0.5"^^xsd:float;
  linkedtv:hasRelevance  "0.5"^^xsd:float ;
  dc:identifier "77929" ;
  dc:source "semitags" ;
  dc:type "location" ;
  owl:sameAs<"http://de.dbpedia.org/resource/
      Neuhardenberg">.
```

For each entity, an instance of the class `oa:Annotation` is created. This annotation establishes an explicit link between the entity extracted and both the media fragment and its subtitles. The provenance information is also attached by using the Provenance Ontology[11].

```
<http://data.linkedtv.eu/annotation/b85339f5-8b89-4
    bf9-a049-d663c50e7ae9>
  a oa:Annotation , prov:Entity ;
  oa:hasBody <http://data.linkedtv.eu/entity/9
      f5f6bc5-fa3a-4de1-b298-2ef364eab29e> ;
  oa:hasTarget <http://data.linkedtv.eu/media/
      e2899e7f-67c1-4a08-9146-5a205f6de457#t
      =1563.56,1566.8> ,
    <http://data.linkedtv.eu/text/1ca03938-c7ae
        -4311-a6ed-0540152b651a#
        offset_12770_12776_Turkey>,
  prov:startedAtTime "2013-02-08T14:14:39.4Z"^^xsd:
      dateTime ;
  prov:wasAttributedTo <http://data.linkedtv.eu/
      organization/EURECOM> ;
  prov:wasDerivedFrom <http://data.linkedtv.eu/text
      /1ca03938-c7ae-4311-a6ed-0540152b651a> .
```

Finally, all the instances are interconnected creating a graph that can be queried through the SPARQL endpoint exposed by the *Triplestore*. At this point the video classification takes place.

4. DATASET

We obtained a random set of 805 videos with their subtitles (in the SRT format) from Dailymotion. Using the site API, we also collected the video metadata including the channel the video belongs to and the video duration. The complete dataset has been processed using the framework

[4] `http://nerd.eurecom.fr/ontology/nerd-v0.5.n3`
[5] `http://www.openannotation.org/spec/core`
[6] `http://www.w3.org/ns/ma-ont`
[7] `http://purl.org/ontology/po`
[8] `http://nlp2rdf.lod2.eu/schema/string`
[9] `http://multimedialab.elis.ugent.be/organon/ontologies/ninsuna`
[10] `http://dublincore.org/documents/2012/06/14/dces`
[11] `http://www.w3.org/TR/prov-o`

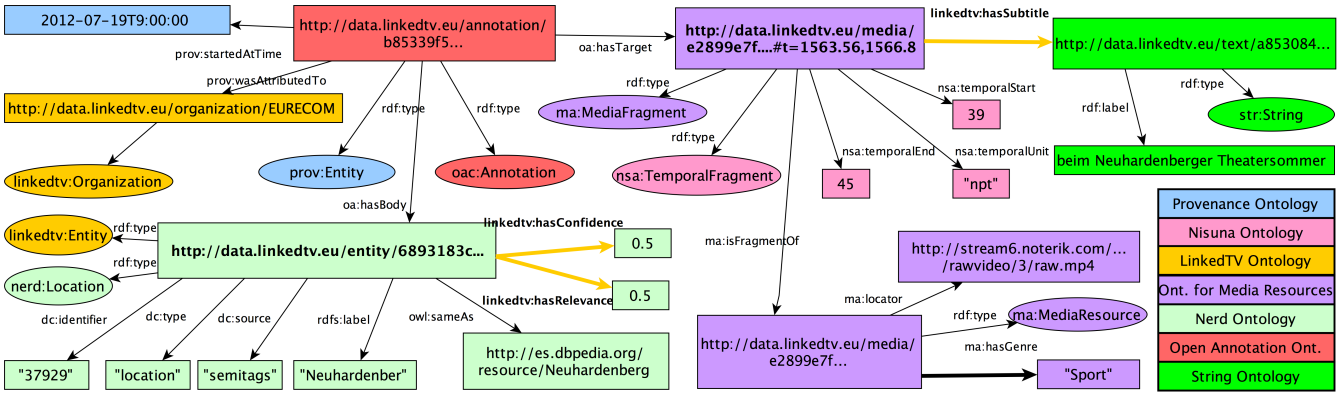

Figure 2: The graph depicts the MediaFragment serialization and how an Entity and its corresponding Subtitle are attached to a MediaFragment though an Annotation.

Table 1: Video and video metadata distribution for the different channels (*ne* stands for named entity)

ch.	id	video	ne	length	ne/video
fun	1	96	1026	30220s	10.67
tech	2	44	4071	24201s	92.57
sport	3	**163**	2794	35940s	17.14
news	4	66	4921	28419s	74.58
creat	5	55	1966	24283s	35.75
lifes	6	**194**	**6996**	62490s	36.09
films	7	81	**16806**	**231657s**	**207.64**
music	8	42	1617	17432s	38.52
other	9	64	4279	29775s	66.88
total	-	805	44476	484417s	55.28

Table 2: Number of named entities grouped by type for each channel

	Thing	Amo	Ani	Evt	Func	Loc	Org	Person	Prod	Time
fun	274	106	0	4	11	103	125	182	151	70
tech	1514	689	**92**	5	66	269	233	571	358	274
sport	618	544	2	20	55	362	197	462	184	350
news	1018	810	3	8	138	827	554	789	374	400
creat	581	274	11	4	60	194	132	379	189	142
life	**2175**	**2010**	5	6	107	328	550	867	589	359
films	1511	1729	14	63	492	1369	1705	**7532**	1233	1158
music	337	201	2	3	45	206	163	403	136	121
other	933	686	49	11	126	604	371	791	381	327
total	**8961**	**7049**	178	124	1100	4262	4030	**11976**	3595	3201

described in Section 3, where the named entities are automatically extracted from the video subtitles and aligned with the corresponding media fragments according to a start time and end time provided by NERD. Even though the original language for the videos in the collection varies, including English, French and Cyrillic languages, all the subtitles are written in English with some special characters in different languages. The duration of the videos ranges from 17 to 7654 seconds. There are 9 different channels covered in our video collection and the distribution of videos per channel is: fun (96), tech (44), sport (163), news (66), creation (55), lifestyle (194), shortfilms (81), music (42) and other (64). The way videos are associated with channels is provided by the video owner and is therefore potentially incorrect. We also assign a unique id to each channel (we also use the following abbreviations: *creation* for *creat*, $lifestyle$ for $life$ and $shortfilms$ for $film$). The number of videos per channel and the total number of named entities extracted per channel are shown in Table 1. The NERD framework enables to group the named entities according to 10 top-level types, namely Thing, Amount, Animal, Event, Function, Location, Organization, Person, Product and Time. Table 2 shows the number of named entities for each of the 10 NERD types per video channel. We observe that most of the named entities belong to Thing, Amount and Person, while Animal and Event are much less extracted. Furthermore, $shortfilms$ has a large amount of named entities of type Person and Function while more than one third of the named entities in Product and Time also belong to $shortfilms$. It is also interesting to notice that most of the named entities of type Animal are extracted from the *tech* channel.

As we mentioned earlier, each named entity extracted from the video subtitles is aligned with a media fragment, where the start and end time of the media fragment correspond to the subtitle block time boundaries. We can further group the named entities in different types in Table 2 by the temporal position this named entity is aligned with. As the duration of the videos vary, we need to normalize the temporal position instead of using the actual value so that video with different duration can be compared with each other. We define the variable tp (named temporal position) as $0 \leqq tp = \frac{st+et}{2\times dur} \leqq 1$, where st and et are the start time and end time of the media fragment the named entity is aligned with, and dur is the duration of the video. When grouping the named entities according to their tp, each video is equally divided into N fragments, so every named entity is included into the fragment the tp falls into. A named entity with temporal position tp belongs to group n if $1 \leqq n = tp \times N \leqq N$. Figure 3 demonstrates the tp distribution of different types of named entities for each channel. For all the plots in Figure 3, the x axis is the temporal position of the named entities in the videos and the y axis is the number of named entities in the temporal segments. Different colors in the figure represent the different NERD types.

We observe that, for some channels, a large amount of named entities are aligned with the end of the videos (Figure 3). For $shortfilms$, 4268 named entities are extracted when $tp \in (0.9, 1]$ and a large proportion of them is of type

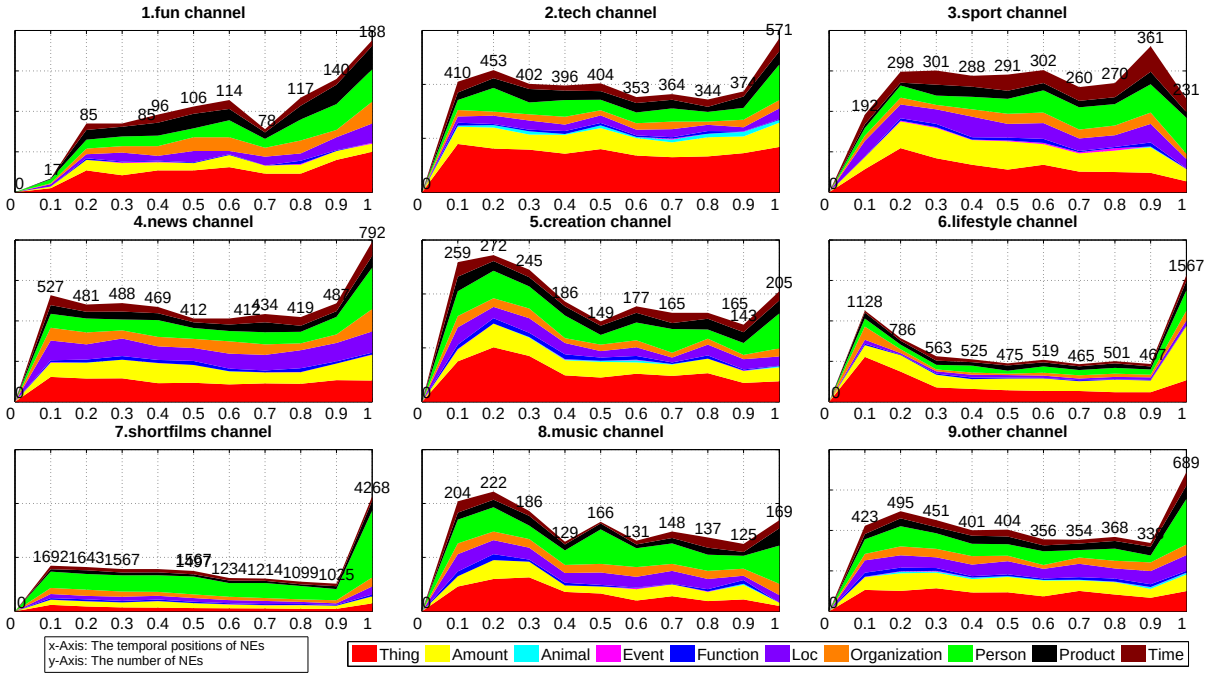

Figure 3: Distribution of named entities extracted from subtitles for each channel and the summary of their temporal position in the videos

Person. The *lifestyle* channel has spikes both at the beginning and the end. The *fun* channel has a very low number of named entities at the beginning and the named entities in *tech*, *news* and *other* channels have a relatively even distribution when $tp \leqq 0.9$ but the numbers are slightly higher at the end. Different from other channels, *sport* has a low number at both the beginning and the end, while it is difficult to see a pattern for *creation* and *music*. If we suppose that the named entities extracted are all correct (which is generally not true but without modifying the trend), these patterns imply some important information that could be useful for video classification and retrieval based on the temporal features.

5. METHODOLOGY

We conducted a multiclass classification experiment, for each research question defined in the Section 1: *Exp1*, *Exp2* and *Exp3*. We run three different experiments to categorize the video dataset into the 9 different Dailymotion channels. The channel information retrieved from the Dailymotion API is considered as the labeled (ground truth) data.

As shown in Table 1, we assign an *id* to each channel c, and $c \in \{1, 2 \cdots 9\}$. In Exp1, we use the number of named entities per each NERD type t as the features. As there are 10 NERD types, each observation is a feature vector $\vec{\mathbf{x}} = [x_1, x_2 \cdots x_{10}]$ and $|\vec{\mathbf{x}}| = 10$, where x_t represents the total number of named entities per NERD type t for a given video. For Exp2, we weight the named entities with their temporal position values tp and group them into N groups. The feature vector in Exp2 is $\vec{\mathbf{x}} = [x_1, x_2, x_3 \cdots x_n]$, where $1 \leqq n \leqq N$. The choice of N may affect the prediction results: at the beginning, we choose a relatively large number $N = 20$ and we then gradually decrease N and see how the results change. For Exp3, we combine Exp1 and Exp2 together and we use the temporal distribution of named entities for each NERD type as features. Consequently, there are $10 \times N$ features in Exp3 and $\vec{\mathbf{x}} = [x_{1,1}, x_{1,2} \cdots x_{t,n}]$. When $N = 20$, $|\vec{\mathbf{x}}| = 200$. Exp1 is a subset of Exp3 where $N = 1$.

For the research question 4, we applied four basic classification algorithms for each experiment: Logistic Regression (LG), K-Nearest Neighbour (KNN), Naive Bayes (NB) and Support Vector Machine (SVM). The mathematical details of each algorithm are out of the scope of this paper. Instead, we will explain how the algorithms are used in the experiments. Firstly, as they are supervised algorithms, we need to divide our dataset into a training and a test set. To make the full use of the dataset and reduce the overfitting problem of each algorithm, we applied a 10-fold cross validation. The 805 videos are divided into 10 equal-sized groups and in each fold, we use 9 groups as the training set and 1 group as the test set. In this way, each video in the dataset appears only once in the test set. Then, when applying different algorithms into each fold, the results can be generically defined as:

$$\hat{\mathbf{R}} = predict(\mathbf{X}^e, \mathbf{Y}^e, \mathbf{X}^r, \mathbf{Y}^r, params) \quad (1)$$

$\mathbf{X}^r$ is a $m_r \times |\vec{\mathbf{x}}|$ matrix of the training data, where m_r is the number of all training videos in the 9 groups. $\mathbf{Y}^r$ is a $1 \times m_r$ matrix of grouping variables, where each entry in $\mathbf{Y}^r$ is the labeled channel id c. Similarly, $\mathbf{X}^e$ is a matrix of testing data. Both $\mathbf{Y}^e$ and $\hat{\mathbf{R}}$ are $1 \times m_e$ matrix and m_e is the number of videos in the test set. Each entry in $\mathbf{Y}^e$ is the labeled channel id c for the videos in the test set, while each entry in $\hat{\mathbf{R}}$ is the predicted channel $\hat{c}$ given the feature vector $\vec{\mathbf{x}}$. The actual definition of the *predict* function in Equation 1 changes according to the different algorithms

used. *params* is a set of parameters that we use to tune each algorithm so that the best results can be obtained.

LG is a statistical machine learning algorithm and it uses exponentiation to convert linear predictors to probabilities. For the experiments, we adopted the multinomial LG and the linear predictors are defined as: $g(\vec{\mathbf{x}}) = \ln \frac{\pi(\vec{\mathbf{x}})}{1-\pi(\vec{\mathbf{x}})} = \beta_0 + \sum_{m=1}^{|\vec{\mathbf{x}}|} \beta_m x_m$. The result for each video using the LG classifier is a vector $\vec{\mathbf{r}} = [r_1, r_2 \cdots r_c]$, where r_c is the probability that this video belongs to channel c. We have therefore $|\vec{\mathbf{r}}| = 9$ and $c = col(r_c)$. In our experiment, we select the channel which has the largest possibility as the final prediction result, i.e.:

$$\hat{c} = col(\max_{c=1}^{9}(r_c)) \tag{2}$$

To reduce the overfitting problem, we applied to the logistic regression the L2-Regularization [10]. Given our settings, we empirically assessed that $\lambda = 0.0001$ has the best bias-variance tradeoff. NB is also a statistical machine learning algorithm, but it has many choices to model the data distribution. We choose the multivariate multinomial distribution, which best fits our problem. Similar to LG, we use Equation 2 to get the prediction result $\hat{c}$ in NB.

KNN is an instance based algorithm and the main tuning parameter is the choice of k. As there is still a lack of principled ways to choose k [7], we run several tests with $k = 1, 2, 3 \cdots 40$. The best results were obtained when $k \in [18, 22]$ and we choose $k = 20$ in all our experiments. For the method to calculate the distance between two instances, we choose the Euclidean Distance. Unlike other algorithms, SVM cannot be directly applied for multiclass classification problems, so we use LIBSVM[12] to implement a 1-vs-1 SVM algorithm and choose the linear kernels $K(x, y) = (x \cdot y + 1)^P$ as the kernel function for all the experiments. Finally, to measure the accuracy of each experiment and algorithm, we need to define the precision P and recall R and F1-score $F1$ for each channel c.

$$P_c = \frac{\sum_{f=1}^{10}, |\hat{R}_f(c) \bigcap Y_f^e(c)|}{\sum_{f=1}^{10} \hat{R}_f(c)} \tag{3}$$

$$R_c = \frac{\sum_{f=1}^{10} |\hat{R}_f(c) \bigcap Y_f^e(c)|}{\sum_{f=1}^{10} Y_f^e(c)} \tag{4}$$

$$F1_c = 2 \times \frac{P_c \times R_c}{P_c + R_c} \tag{5}$$

$\hat{R}_f(c)$ is the set of videos that have been predicted belonging to channel c in the fth fold of cross validation, while $\hat{Y}_f(c)$ is the videos that have been labeled in channel c. Therefore, $|\hat{R}_f(c) \bigcap Y_f^e(c)|$ is the number of videos that are correctly categorized in the channel c in a cross validation fold. There is the possibility that $\sum_{f=1}^{10} \hat{R}_f(c) = 0$ if no video have been categorized in the channel c. In this case, the value of P_c is NaN. Our dataset has videos in all channel, so $\sum_{f=1}^{10} Y_f^e(c) \neq 0$. To evaluate the overall accuracy $acc =$ of the algorithm in each experiment on the entire dataset, we define:

$$acc = \frac{\sum_{c=1}^{9} \sum_{f=1}^{10}, |\hat{R}_f(c) \bigcap Y_f^e(c)|}{805} \tag{6}$$

[12] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

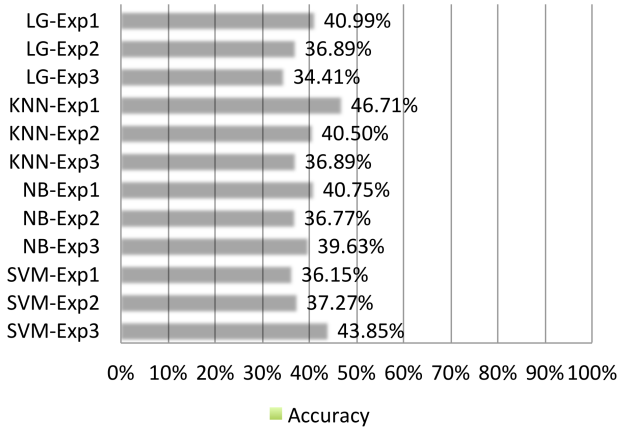

Figure 4: Accuracy comparison for each algorithm-experiment pair.

The overall accuracy is the total number of videos that have been correctly classified divided by the total number of the video since each video appears exactly once in the test set.

6. EXPERIMENTS AND DISCUSSION

We analyze below the results obtained for each experiment in order to see which set of features and which algorithm(s) perform best for the automatic video classification task depending on the channels.

6.1 Overall Accuracy

Figure 4 shows the overall accuracy for each experiment (see Section 5). We have tried $N = 5, 10, 20$ and in most of the experiments, $N = 20$ outperformed the other groupings. We only present Exp2 and Exp3 when $N = 20$. The best accuracy is obtained with KNN-Exp1 (46.58%) and the worst one is LG-Exp3 (33.54%). Generally speaking, there are no major differences between each algorithms using different sets of features for the overall accuracy. The features chosen in Exp1 perform better than the other two feature sets using LG and KNN. For NB, the *acc* for Exp1 and Exp3 are close and they are all better than Exp2. SVM-Exp3 outperform Exp1 and Exp2 and it is also the best accuracy in Exp3 compared with other algorithms. We cannot clearly draw any conclusions regarding which algorithm and feature set combination performs the best (Figure 4). However, if the feature set includes the breakdown of named entities based on NERD types (Exp1 and Exp3), the accuracy is quite likely to be better than the ones using only temporal positions (Exp2). From this point of view, it is possible to infer that the number of named entities and their type is an indicator to be taken into account for improving the video classification algorithm, assuming of course that there is a sufficient number of named entities detected for each NERD type.

6.2 Breakdown scores per Channel

Table 3, 4, 5 and 6 report the breakdown scores for each channel and experiment considering precision, recall and $F1$. The 3 largest numbers for each measurement are highlighted in bold. If we use $F1$ as the general measure of the accuracy, *sport*, $life$ and $shortfilms$ obtain usually the best accuracy

Table 3: Precision (P), recall (R) and F-measure (F1) on various channels for the experiments using logistic regressions (%), $\lambda = 0.0001$.

	Exp1			Exp2			Exp3		
Ch.	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
fun	28.87	29.17	29.02	35.71	31.25	33.33	18.87	20.83	19.8
tech	33.33	15.91	21.54	24	13.64	17.39	17.54	22.73	19.8
sport	**35.69**	**71.17**	**47.54**	**32.18**	**68.71**	**43.84**	**38.82**	**36.2**	**37.46**
news	32.26	15.15	20.62	30.77	12.12	17.39	15.39	18.18	16.67
creat	8.33	1.82	2.99	5.26	1.82	2.7	7.02	7.27	7.14
life	**49.78**	**58.25**	**53.68**	**50.23**	**56.19**	**53.04**	**57.9**	**56.7**	**57.29**
films	**73.13**	**60.49**	**66.22**	**66.67**	**41.98**	**51.52**	**54.76**	**56.79**	**55.76**
music	NaN	0	0	5.88	2.38	3.39	10	11.91	10.87
other	16	6.25	8.99	0	0	0	12.9	6.25	8.42

Table 4: Precision (P), recall (R) and F-measure (F1) on various channels for the experiments using K-Nearest Neighbour (%), $k = 20$.

	*Exp1			Exp2			Exp3		
Ch.	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	***R**	**F1**
fun	23.91	22.92	23.4	**47.69**	32.29	38.51	21.05	20.83	20.94
tech	45	20.46	28.13	37.93	25	30.14	**42.86**	6.82	11.76
sport	**50**	**66.87**	**57.22**	42.48	**58.9**	**49.36**	29.86	**26.38**	**28.01**
news	**54.17**	19.7	28.89	18.75	9.09	12.24	**42.86**	4.55	8.22
creat	28.57	18.18	22.22	6.25	1.82	2.82	33.33	1.82	3.45
life	48.01	**74.74**	**58.47**	**44.04**	**81.96**	**57.3**	34.36	**86.08**	**49.12**
films	**72.29**	**74.07**	**73.17**	**86**	**53.09**	**65.65**	**80.65**	**61.73**	**69.93**
music	20	2.38	4.26	20	2.38	4.26	NaN	0	0
other	23.08	9.38	13.33	19.05	6.25	9.41	0	0	0

in the different experiments and using different regression algorithms, while the $F1$ of *news*, *creation*, *music* and *other* are usually below 20%. This behavior makes sense since the number of samples available for that first set of channels is bigger than for the second group, therefore the training phase of the classification performs better.

Using LG with $\lambda = 0.0001$, $lifestyle$ and $shortfilms$ consistently gain high accuracy in all the three experiments. All P, R and $F1$ scores are high for $shortfilms$ in Exp1, but the $F1$ for *creation*, *music* and *other* is very low ($\leqq 10\%$). When the temporal distribution of media fragments is considered (Exp3), the $F1$ for *sport* and $shortfilms$ is lower than Exp1, but *creation* and *music* are improved. For KNN,

Table 5: Precision (P), recall (R) and F-measure (F1) on various channels for the experiments using Naive Bayes (%).

	Exp1			Exp2			Exp3		
Ch.	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
fun	31.82	29.17	30.43	18.75	12.5	15	22.68	22.92	22.8
tech	40.74	25	30.99	30.77	9.09	14.04	28.26	29.55	28.89
sport	**44.87**	**42.95**	**43.89**	32.89	**60.74**	**42.67**	**47.4**	**55.83**	**51.27**
news	29.83	25.76	27.64	**38.46**	15.15	21.74	33.33	25.76	29.06
creat	26.32	9.09	13.51	9.09	5.45	6.82	13.51	9.09	10.87
life	**44.06**	**72.68**	**54.86**	**46.28**	**60.83**	**52.56**	**52.56**	**63.4**	**57.48**
films	**55.77**	**71.61**	**62.7**	**62.9**	**48.15**	**54.55**	**61.18**	**64.2**	**62.65**
music	12	7.14	8.96	3.7	2.38	2.9	19.36	14.29	16.44
other	0	0	0	8.33	3.13	4.55	12.5	6.25	8.33

Table 6: Precision (P), recall (R) and F-measure (F1) on various channels for the experiments using Support Vector Machine (%).

	Exp1			Exp2			*Exp3		
Ch.	**P**	***R**	**F1**	**P**	**R**	**F1**	**P**	***R**	**F1**
fun	33.33	**8.33**	**13.33**	**45.71**	16.67	24.43	**52.63**	20.83	29.85
tech	NaN	0	0	NaN	0	0	26.92	15.91	20
sport	**50**	**62.58**	**55.59**	**43.48**	**61.35**	**50.89**	34.78	**88.34**	**49.91**
news	**50**	4.55	8.33	0	0	0	25.81	12.12	16.49
creat	26.67	7.27	11.43	37.5	10.91	16.9	0	0	0
life	31.37	**87.63**	**46.2**	**49.1**	**70.1**	**57.75**	**66.47**	**57.22**	**61.5**
films	**36.36**	4.94	8.7	26.75	**80.25**	**40.12**	**49.17**	**72.84**	**58.71**
music	NaN	0	0	NaN	0	0	NaN	0	0
other	0	0	0	NaN	0	0	40	3.13	5.8

the P, R and $F1$ are all above 70% for $shortfilms$ in Exp1, which is the overall best score. Compared with Exp1, Exp2 and Exp3 obtained worst results for nearly channels. In Exp3, no instances are correctly recognized in *music* and *other*. Using NB, $F1$ for *sport*, $lifestyle$ and $shortfilms$ are good in both Exp1 and Exp3. SVM performs better when dealing with multi-dimensional data, so the best result for SVM is in Exp3 where 200 features are used for the classification. SVM also generates the best $F1$ for $lifestyle$ (61.5%) among all the other algorithms. But unlike other algorithms, the accuracy for the $shortfilms$ channel in SVM-Exp1 is very low, while *sport* and $lifestyle$ are still high. This is due to the fact that SVM relies more on the size of the samples when the size of the features are small.

Algorithms such as LG and SVM require large training dataset to achieve better classification results [7]. Hence, channels with large sample size (*sport* and $lifestyle$) are more likely to obtain high accuracy in most of the algorithms. However, even though the sample size of $shortfilms$ is not big, the $NEs/Video$ value in Table 1 is much larger than for the others channels. This is because the average length of the video in $shortfilms$ channel is longer than the videos in other channels and more named entities can be extracted from their subtitles. Considering the use of media fragments in this experiment, the characteristics of temporal and NERD type distribution of named entities for $shortfilms$ are also outstanding: large number of named entities are associated with the end of the videos and most of them are Person. So considering those two factors and the sample size of $shortfilms$, it is possible to understand why the accuracy of this channel is higher for most of the experiments. Sample size is still the key factor for SVM regression in this context.

These algorithms achieve very high recall score but pretty low precision in some experiments. For example, the R of $lifestyle$ in KNN-Exp3 (Table 4) is very high (86.08%), but the P is very low (34.36%). Taking a deeper look at the content of the datasets, it is possible to see that there are 194 videos in $lifestyle$, but in the results of the classification, many more instances have been marked as belonging to this channel (486 in total). However, 319 of them have been wrongly categorized. Similar situations occur in the $lifestyle$ channel in SVM-Exp1 and in the *sport* channel in SVM-Exp3 (Table 6). In some channels, we observe that the classification accuracy is very low when not enough instances are predicted to belong to this channel. If we put

these two phenomenons together with Table 1 and Figure 3, we find out that channels with very clear entity distribution patterns or with large sample size (e.g. *sport*, *lifestyle* and *shortfilms*) will tend to have high R but low P. We conclude that classification can be improved with larger sample size but also by investigating which features will be the most influential for each algorithm.

7. CONCLUSION AND FUTURE WORK

The explosion of online video data repositories has increased the need for semantic video indexing techniques. In this paper, we discussed a new way for classifying video content by extracting named entities from the video subtitles which are associated with media fragments. The results obtained for the three proposed experiments indicate that the implemented method is very promising in the context of online videos classification.

Summarizing the experiment results, we get positive answers for the first 3 research questions described in Section 1. We also conclude that there is no dominant algorithm that outperforms the others for the 3 experiments in terms of the overall accuracy given the dataset we use in this experiment. When using named entities and media fragment features together, SVM obtains the best overall result. Among the individual channels, *sport*, *lifestyle* and *shortfilms* have the highest prediction accuracy. The best accuracy is obtained when using the number of named entities in each NERD type and KNN ($k = 20$) algorithm to predict videos in *shortfilms* channel. Except for the type of regression algorithms used, we roughly observed three different factors that affect the prediction accuracy in each channel: the sample size, the average number of named entities in each NERD type per video, and the temporal position distribution of the entities along the duration of the media item. Among those three factors, the sample size can be increased when collecting more data, but the other two may follow some distributions for each channel, which requires further investigation.

In the future, we plan to collect more data from Dailymotion and see if the patterns observed are similar in YouTube. We will also study how accurate the human classification is with respect to the automatic classification based on the video metadata. The proposed new features regarding named entities and media fragments can be combined with other features, either low-level or high-level, to improve video classification and retrieval in future work.

8. ACKNOWLEDGMENTS

The authors thank Dailymotion for providing the video collection used in the experiment. This work was partially supported by the European Union's 7th Framework Programme via the project LinkedTV (GA 287911).

9. REFERENCES

[1] L. Bentivogli, P. Forner, B. Magnini, and E. Pianta. Revising the wordnet domains hierarchy: semantics, coverage and balancing. In *Workshop on Multilingual Linguistic Resources*, 2004.

[2] D. Brezeale and D. J. Cook. Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(3):416–430, 2008.

[3] K. Filippova and K. B. Hall. Improved video categorization from text metadata and user comments. In *34th International ACM SIGIR Conference on Research and development in Information Retrieval*, pages 835–842, 2011.

[4] B. Haslhofer, W. Jochum, R. King, C. Sadilek, and K. Schellner. The LEMO annotation framework: weaving multimedia annotations with the web. *International Journal on Digital Libraries*, 10(1):15–32, 2009.

[5] C. Huang, T. Fu, and H. Chen. Text-based video content classification for online video-sharing sites. *Journal of the American Society for Information Science and Technology*, 61(5):891–906, 2010.

[6] P. Katsiouli, V. Tsetsos, and S. Hadjiefthymiades. Semantic video classification based on subtitles and domain terminologies. In *Workshop on Knowledge Acquisition from Multimedia Content (SAMT'07)*, 2007.

[7] S. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques. *Frontiers in Artificial Intelligence and Applications*, 160:3, 2007.

[8] Y. Li, G. Rizzo, R. Troncy, M. Wald, and G. Wills. Creating enriched YouTube media fragments with NERD using timed-text. *11th International Semantic Web Conference, Demo Session*, 2012.

[9] Y. Li, M. Wald, T. Omitola, N. Shadbolt, and G. Wills. Synote: Weaving Media Fragments and Linked Data. In *5th International Workshop on Linked Data on the Web (LDOW'12)*, 2012.

[10] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *21st International Conference on Machine learning*, 2004.

[11] J. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. *Computer Vision–ECCV 2010*, pages 392–405, 2010.

[12] G. Rizzo and R. Troncy. NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. In *13th Conference of the European Chapter of the Association for computational Linguistics (EACL'12)*, 2012.

[13] B. Schopman, D. Brickly, L. Aroyo, C. Van Aart, V. Buser, R. Siebes, L. Nixon, L. Miller, V. Malaise, M. Minno, et al. Notube: making the web part of personalised tv, 2010.

[14] T. Steiner. SemWebVid - Making Video a First Class Semantic Web Citizen and a First Class Web Bourgeois. In *9th International Semantic Web Conference (ISWC'10), Demo Session*, 2010.

[15] J. Waitelonis, N. Ludwig, and H. Sack. Use what you have: Yovisto video search engine takes a semantic turn. In *5th International Conference on Semantic and digital media technologies (SAMT'10)*, 2011.

[16] J. R. Zhang, Y. Song, and T. Leung. Improving video classification via youtube video co-watch data. In *Workshop on Social and behavioural networked media access*, 2011.