

Classifying YouTube Channels: a Practical System

Vincent Simonet
Google, Paris (France)
vtst@google.com

ABSTRACT

This paper presents a framework for categorizing channels of videos in a thematic taxonomy with high precision and coverage. The proposed approach consists of three main steps. First, videos are annotated by semantic entities describing their central topics. Second, semantic entities are mapped to categories using a combination of classifiers. Last, the categorization of channels is obtained by combining the results of both previous steps.

This framework has been deployed on the whole corpus of *YouTube*, in 8 languages, and used to build several user facing products. Beyond the description of the framework, this paper gives insight into practical aspects and experience: rationale from product requirements to the choice of the solution, spam filtering, human-based evaluations of the quality of the results, and measured metrics on the live site.

Categories and Subject Descriptors

J.0 [Computer Applications]: General

General Terms

Algorithms, Experimentation

Keywords

Semantic entity, Taxonomy classification, Video

1. INTRODUCTION

1.1 The tackled problem

Having a look at their contents, you would probably say in a few seconds that *Machinima* is a *YouTube* channel about video games, *ligue1fr* about Soccer and *CNN* about news. But how to determine this algorithmically? And how to do this at the scale of *YouTube* corpus? These are in short the challenges this paper tackles.

YouTube is placing the concept of *channel* at the core of its strategy to develop content and audience¹. A channel can be viewed as a living set of videos which share a common property: they are from the same person or organization,

¹<http://youtube-global.blogspot.com/2012/03/welcome-to-your-new-youtube-channel.html>

they are about the same topic, they are related to the same event, etc. A channel might be created by a content creator (a person who uploads original videos to the site) or generated by a curator (a person who recommends videos on the site or even an algorithm doing so, see section 7.2). A channel has a live feed of events, in which videos may be published, and users may subscribe to them. Channels are engaging creators and curators, by gathering an audience for them. Channels are engaging users, by recommending them videos about things they like.

With channels playing such a central role, enabling their discovery becomes more and more crucial. Discovery features for videos can be extended to channels, yielding to channel search, channel recommendations and related channels. These features are very powerful for users who have a watch history, or who know precisely what they are looking for. In order to engage users who are new to the site, or who do not know exactly what they want to watch, we wanted to provide a catalog of user channels. For every thematic category of content available on *YouTube* (e.g. music, sports, news or even sub-categories like rock music, tennis, politics), the catalog should provide a list of interesting channels. For building this, channels must be classified into a taxonomy of thematic categories.

This feature is known as the channels browser², because it allows users to browse a catalog of channels and to subscribe to those they are interested in. It is a major feature, as it allows users to discover new channels they may be interested in, and also acts as a *showcase* for the site.

1.2 A word about manual classification

The channels browser was launched on *YouTube* with a dozen of categories, and a manually curated list of channels for each of these categories, with a focus on the United States. This approach was fine as a starting point, but it does not scale for the following reasons:

- It limits the channels browser to some hundreds or thousands of well known channels (e.g. channels from partners or celebrities), as it is not possible to manually classify millions of channels.
- The channels browser should be a living selection of channels. New channels are created every day, others are closed or abandoned. The audience, the interest or even the theme of every channel is evolving. So, the list cannot be developed once for all, but has to be continuously maintained.

²<http://www.youtube.com/channels>

- *YouTube* is officially launched in more than 50 countries. This requires to develop and maintain this number of versions of the catalog.
- Manual selection of channels might be questioned by the different stakeholders.
- The list of categories, initially limited to 10, should be extended to many more in order to match as close as possible user interests.

An alternative option would have been to ask creators and curators to classify their own channels, *i.e.* specify in which category they want their channel to appear. If it looks simple, this approach has the following disadvantages:

- It requires an additional action from the channel owners, which might be painful or error prone if the list of categories is large.
- It does not work for algorithmically generated channels, as well as it requires some back-fill work for all existing channels.
- It almost prevents changing the taxonomy from time to time (or limits how it can be changed), because it would require every creator or curator to update the classifications.

We decided to work around these problems by algorithmically generating the channels browser, *i.e.* by developing a program able to classify channels within the taxonomy without human intervention.

2. OVERVIEW OF THE SOLUTION

Algorithmic classification of video channels is a difficult problem, because it requires to understand what the channel and its videos are about, with a limited quantity of algorithmically exploitable information. Many works about video content analysis are available in the literature [6, 17, 10, 15, 8, 20, 22]. Some results are available, allowing for instance to recognize objects or persons, or even sequences of events. But such algorithms are not suitable to discover what videos are *really* about, *i.e.* which users will be interested in watching the video. For instance, a video may contain a lot of kites or its soundtrack may have many occurrences of the word *kite*, while the video would neither be really about kites or interesting for users who care about this topic. These algorithms might also be hard to scale to a video corpus of the size of *YouTube*.

The approach proposed in this paper follows a completely different path for determining what videos and channels are about, as it almost completely ignores the image and audio contents, and relies on the meta-data associated with the videos and the user channels. These are mainly the title, the description and the keywords entered as “free text” by the users when they create channels and upload videos. Relying on text allows applying known techniques like semantic entity extraction and disambiguation, but with new challenges as the amount of textual information available for a channel or a video is usually sparse and short, which requires to introduce some additional refinements. Running a simple text-based taxonomic classifier (see section 6.2) on the video meta-data would lead to poor quality results.

The classification process we have developed and we describe in this paper consists in three main steps:

1. Mapping videos to semantic entities,
2. Mapping semantic entities to taxonomic categories,
3. Mapping channels to taxonomic categories (by combining both previous steps).

3. METRICS

This section introduces the metrics which have been used in the three steps of the algorithm (see sections 5, 6 and 7) to assess the effectiveness of our work, and to guide technical decisions.

Precision and *recall* are the usual metrics for classification algorithms [21]. We measured them by “off-line” methods, so that they could be tracked during the development phase before any product launch. After launch, we completed these measures by the *subscription rate* on the channels browser, which allows to assess the overall effectiveness of our approach.

3.1 Precision

The *precision* of a classification algorithm is the fraction of classification results produced by the algorithm which are relevant. Because of the nature of the tackled problems, we measure precision by running evaluations with humans. These evaluations are performed by:

- Selecting a representative sample of items to be classified (typically 500).
- Elaborating a questionnaire and submitting it to a pool of independent raters. The pool of rater should be large enough so that each rater gets a limited number of items to rate (typically 10).
- Analyzing the results provided by the human raters.

The questionnaire contains at least questions about the relevance of the produced classification for the items, but, in general, it also includes some questions in order to ensure that the rater can properly address his/her task. In order to ensure the accuracy of the evaluation, every item is submitted in parallel to five raters. Items which have answers indicating that the raters did not understand the questionnaire, or which have diverging answers are excluded from the analysis. The amount of excluded items is measured in order to track the proper execution of the evaluation. We typically require it to be lower than 5 %.

See sections 5.4, 6.5 and 7.1 for examples of such evaluations.

3.2 Recall and coverage

The *recall* of a classification algorithm is the fraction of classifiable items which are effectively classified by the algorithm. The *coverage* is the fraction of the items which are classified by the algorithm. As it is easier to measure, we decided to use coverage as a proxy for recall.

We measured coverage by running the algorithms on the whole corpus, and counting the number of items for which outputs were produced. It is in general relevant to weight the coverage measurement by a measure of the audience, *e.g.* the number of views or the number of subscriptions (so that the measurement is representative of the actual user experience).

We devoted a lot of attention to balancing precision and recall when developing the framework. As explained in section 1.2, we were replacing a manually classified set of user channels, which was of high quality but of low coverage. We hence strategically decided to first focus on the precision of the system and to ensure it is as close as 100%, so that it could be viewed as an acceptable replacement for the human classification. We then worked out to increase the coverage (without affecting the precision), in order to increase the quality as a whole, and to exceed that of the manual classification.

3.3 Subscription rate

Once a user-facing product, the channels browser, was launched on top of the framework, we have been able to measure its effectiveness relatively to the previous manually curated results by analyzing usage statistics, and in particular subscription rates. See section 8.1 for more details.

4. RELATED WORK

Driven by the popularity of the web, algorithmic taxonomic classification of text document has been the topic of many research works, in particular in the machine learning community, *e.g.* [14, 19, 5]. We took benefits of such techniques in our classification algorithm, see section 6.2. More recently, similar techniques have been developed for image classification [6], and then video classification [18]. Despite their intrinsic interest, these techniques seem to have encountered a limited impact on videos. We believe the main reasons are the difficulty to extract sufficient and relevant information from videos as well as to develop training sets, both having a direct impact on precision and recall.

Video content analysis or computer vision [9] may appear as one way to cope with this data issue. It deserved a lot of attention in the last decade, especially with image analysis techniques [6, 17], action identification [10, 15, 8] and video tagging [20, 22]. However huge challenges remain to be solved, *e.g.* for being able to extract the central topics of a video and to scale the methods to a large corpus of videos.

These are the reasons why we investigated the use of entity extraction and disambiguation in text documents to video content. Entity extraction and disambiguation in text documents has been a very active area of research in the last few years, especially driven by the objective of extending search engines from a purely keyword matching approach to an entity matching approach [7, 2, 4, 3, 16, 13, 11, 12, 23]. Most techniques focus on entities present in a knowledge base, like DBpedia, YAGO or *Freebase* [1], and use content matching approach to disambiguate entities. Our framework re-uses a large amount of this work, and applies it to videos. To the best of our knowledge, this is the first large scale application of such techniques on a large corpus of videos. As for the previously considered techniques, the main challenges are again the limited amount of available information. However, they seem to be much better suited for coping with this challenge with high precision and recall.

5. ANNOTATING VIDEOS

5.1 Semantic entities

Freebase [1] is a knowledge base maintained by a community supported by *Google*, which aims at gathering as much

of the world’s knowledge as possible. It is organized around *entities* (*a.k.a.* *topics*, the nodes of the graph), which are connected together by *properties* (the edges of the graph). By entity, one means every concrete or abstract concepts that people may designate. Persons, places, objects, artworks are entities. Abstract concepts like interview, mathematics or even happiness are also entities. Every entity has *types*, which define the properties it may have.

Freebase contains data harvested from various sources and curated, as well as user-contributed data. As of mid-2012, it contains around 22 millions entities, part of which are mapped to Wikipedia articles. *Freebase*’s content is internationalized, by including entities of country-specific interest, and by having translations of textual data in several languages (see section 7.3).

In this paper, *Freebase* is used as the source of entities for annotating videos (see section 5.3), and for creating algorithmic channels (see section 7.2). There are of course many entities which are not really relevant for *YouTube* videos in *Freebase*. They are simply ignored by our classification algorithm.

5.2 Entity names

In textual documents, humans use *names* to designate entities. The mapping from names to entities is N-to-N. In a given language, every entity may have several names (*e.g.* the *City of Paris* may be designated as *Paris*, *Capital of France* or *City of Lights*). On the other hand, a single name may designate different entities depending on the context (*e.g.* *Jaguar* may designate an animal or a car manufacturer).

A mapping from names to entities has been built by analyzing *Google Search* logs, and, in particular, by analyzing the web queries people are using to get to the *Wikipedia* article for a given entity. After inverting it, we get a table mapping every name to a list of entities with probabilities. For instance, this table maps the name *Jaguar* to the entity *Jaguar car* with a probability of around 45 % and to the entity *Jaguar animal* with a probability of around 35 %.

This table is then enriched by contextual support between entities, by analyzing the links between the entities (or their underlying *Wikipedia* pages). For instance, having the entity *Savannah* in the context increases the probability that the name *Jaguar* designates the animal and not the car. This refinement plays a crucial role for disambiguating annotations in section 5.3.

5.3 Annotation process

Figure 1 represents the process for annotating videos with entities. In the first step, the video is converted into a textual document by gathering the meta-data associated with it (like the title, the description and the keywords entered by the user). In the second step, the textual document is annotated with semantic entities using techniques known as entity extraction and disambiguation [2, 4, 3, 16, 13]:

1. All fragments of the document which may designate an entity are identified. Their relative positions are taken into account in order to disambiguate them (see section 5.2).
2. A ranking of all mentioned entities is produced, reflecting how “topical” (*i.e.* important) they are for the document.

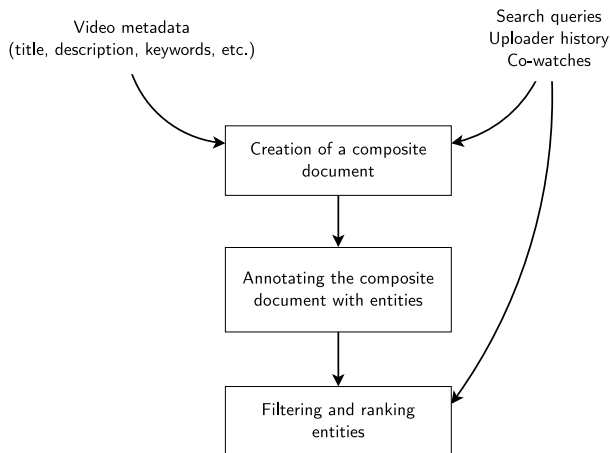


Figure 1: Annotation process

Table 1: Metrics for video annotation

		English videos	French videos	All videos
Precision	Central	82.8 %	83.4 %	
	Relevant	12.4 %	10.6 %	
	Off-topic	4.8 %	5 %	
Coverage	Videos	68.7 %	65.2 %	62.9 %
	Views	91.0 %	87.4 %	83.3 %

We use three main additional signals:

- The top search queries which lead to effective watches of the videos.
- The videos which have been watched in the same user sessions. Watchers tend to watch thematically consistent videos in the same session.
- The upload history of every uploader, *i.e.* the annotation of the videos which have been previously uploaded. Uploaders tend to upload thematically consistent videos over time.

In addition to improving quality, these additional signals allow for filtering some *spam* which may exist in video meta-data. For instance, if an uploader adds the name of some celebrity in the meta-data of one of his/her videos, which is not related to the celebrity, for the purpose of trying to improve its ranking in search; the video will not lead to as many effective watches by users searching for the celebrity name as a video which is actually about the celebrity. Hence, our algorithm will be able to filter out the annotation by the entity.

This annotation process is of course performed by taking the language of the video into account. We currently support 8 languages (including non-Latin ones).

5.4 Metrics

We use both metrics introduced in section 3 to measure the quality of the computed annotation: precision and coverage.

Precision. The precision measures how the generated annotations for a given video are relevant with regards to the

video content. We defined three levels of relationship between a video and an entity:

- *Central.* The entity is one of the key entities to describe what the video is about. For instance, *Lady Gaga* would be a central entity for the clip of *Bad Romance*.
- *Relevant.* The video is about the entity, but this is not one of the key entities to describe what the video is about. For instance, *Pop music* would be a relevant entity for the clip of *Bad Romance*.
- *Off-topic.* The entity is not related to the video (*i.e.* the annotation is wrong). For instance, *Sport* would be an off-topic entity for some music video.

We measured the precision of our annotation process by running human-based evaluations on sample sets of videos. As the content language is a key parameter in the annotation process, we built one set per supported language. For each of these sets, we ensured it is representative in term of categories and audience of videos of the whole *YouTube* corpus. Results for English and French languages are summarized in table 1. They are similar for the other languages we support. They show that less than 5 % of the annotations we produce are wrong, which seems low compared to the difficulty of the task.

Coverage. The coverage is measured by computing the fraction of videos which are annotated by at least one entity. It is relevant to weight this measurement by the respective audience of the videos, *i.e.* their number of views. We measured the coverage by running the the algorithm on all *YouTube* videos in languages we support. Table 1 summarizes the results. In the languages we support, we reach a coverage around 65 % of the videos, but 90 % of the views. The significant discrepancy between both percentages is due to two factors. First, high audience videos tend to have better meta-data (like title and description, but also user queries leading to them). Second, they tend to be more often about well identified topics. Table 1 shows also the coverage with regards of the whole corpus (including languages we do not support). These figures are relatively close to the per-language figures, as the 8 languages we support represent more than 90 % of *YouTube* audience.

6. CLASSIFYING ENTITIES

6.1 Taxonomy

Various taxonomies have been developed in order to classify corpus of written, audio or video contents. Libraries have long used hierarchical taxonomies such as the Library of Congress System³. Similarly, in the early age of the web, portal sites like [dmoz.org](http://www.dmoz.org/)⁴ presented a hierarchical classification of web sites. More recently, *Wikipedia* provides several hierarchical taxonomies in its *Contents* pages⁵. *Google* also developed taxonomies for its advertisement products. Many other examples exist.

These taxonomies are not very well suited for classifying *YouTube* videos and channels. The distribution of *YouTube*

³<http://www.loc.gov/catdir/cpsol/lcco/>

⁴<http://www.dmoz.org/>

⁵<http://en.wikipedia.org/wiki/Portal:Contents>

Table 2: Excerpt of the taxonomy

Category ID	Category name
/animals	Animals
/animals/cats	Cats
/animals/dogs	Dogs
...	...
/music	Music
/music/classical	Classical
/music/pop	Pop
...	...

videos and channels is quite different than in a library or an encyclopedia. For instance, *Mathematics* appears as a first-level node in some *Wikipedia* taxonomies, while it does not deserve such importance for *YouTube* contents and audience. So, we decided to develop a specific taxonomy for our purpose, see an excerpt of it in Table 2. The complete tree has around 300 nodes, with a depth of 2 to 4 depending on the branches. However, it is worth noting that the approach described in this paper is not particularly linked to this taxonomy.

The taxonomy is defined as a directed acyclic graph, with a single root, whose nodes are called *categories*. One says *A* is a *sub-category* of *B* if there is an edge from *B* to *A* in the graph.

6.2 Features

Our classification algorithm works by extracting *features* about entities, and using some simple machine learning. A *feature* is basically a piece of information about the entity selected in a given space. We use several feature spaces, corresponding to several sources of information we combine. For every feature space, we developed a mapping from features to categories, which models the probability that an entity having a certain feature should be categorized in a given category of the taxonomy. We call this mapping a *model*.

The following sub-sections describe the feature spaces and models we currently use in combination to classify entities.

Entity types. As explained in section 5.1, every entity of *Freebase* has one or several types [1]. Types provide useful information about what entities are. For instance, a musical artist (type `/music/artist`) is likely to be classified under the `/music` category, while an Olympics athlete (type `/olympics/olympic_athlete`) is likely to be classified under the `/sports` category.

In *Freebase*, the types for a given entity are split into two sets:

- The *notable* types, which are the types for which the entity is widely known.
- The *other* types, which are other types of lower importance.

(This ranking is specific to the entity, *i.e.* the same type may appear as a notable type for some entities, and as an other type for other entities.) Further to some quality analysis (see section 6.5) we decided to use only the notable types, as the other types tend to introduce some noise.

Freebase properties. Some of the properties which connect entities together in *Freebase* are particularly useful for

classifying entities. For instance, in *Freebase*, every music artist is connected to entities representing music genres. We extract some of this information as features, mainly in the music, film, gaming and sport areas, and use them as input for the classification algorithm. These features can be weighted with data coming from *Freebase* (*e.g.* fraction of music records of an artist which fall into a given music genre).

Ads-related categories. As mentioned in section 6.1, several taxonomies for web documents have already been developed at *Google* for advertising purposes, together with a text classifier.

As they were already developed and well-proven, this classifier was of prominent interest as input for our classification algorithm. Hence, we passed the English description of every entity from *Freebase* through this classifier, and used its output as input features for our classifier with a dedicated model. However, it is worth mentioning that this classifier alone would fall short of solving our classification problem with required precision and coverage.

Portal pages. Wikipedia portals are pages intended to serve as “main pages” for specific topics or areas⁶. As of mid-2012, there are around 2000 portals (gathering all languages), like portals about food, anime and manga or insects.

These portals are a very interesting source of information for classifying entities, as they cover a wide range of topics which can be mapped to our taxonomy. We extract the references from entities to portal pages from *Freebase*, and built a model mapping portals to categories.

6.3 Classification algorithm

The classification algorithm works on one entity at a time, *i.e.* it can be defined as a function from entities to categories. (The only exception to this abstraction is the model training, see section 6.4.) For a given entity, it proceeds as follows:

1. It collects all the features for the entity from the different sources. Every feature is associated to the entity with a numeric weight.
2. It maps the features to actual categories by using the models. The weight of a category for the entity is computed as a combination of the weight of the category for the feature and the weight of the feature for the category. If several features map one entity to the same category, weights are summed up. Weights are also propagated to the root of the taxonomy (*e.g.* the weight of the `/music` category is the sum of the weights of the features directly associated with this category, and the weights of its sub-categories).
3. Last, the taxonomy is traversed from the root to the leaf by selecting at every node the sub-category which has the highest weight. If the ratio between the first and the second higher weighted sub-categories of a given category is lower than a specific threshold, the traversal is stopped. This leads to a classification to a non-leaf category, which makes sense in particular for broad topics.

In order to increase precision of the classification, two additional criteria are added when traversing the taxonomy.

⁶<http://en.wikipedia.org/wiki/Wikipedia:Portal>

First, only the categories whose weights come from several feature spaces are considered (except if the category is mapped in only one space). This allows filtering out errors that every source inevitably contains, as well as resolving some ambiguities.

Second, only categories which are consistent with the types selected by video uploaders are selected. Let’s explain this in more detail. Since its creation, *YouTube* has allowed video uploaders to select one type among a dozen when uploading a video. Using the annotations of videos by entities (see section 5) we are able to compute, for every entity, a distribution of the actual types of the videos it annotates, and to compare this distribution of the types with the same distribution for the whole *YouTube* corpus. This allows determining one or several major video types for the entity, and allows the algorithm to restrict the classification to a subset of the taxonomy.

6.4 Developing the models

The development of the models is a key step in the development of the classifier, as it directly affects the quality of the obtained classification, both in term of precision and coverage (see section 6.5).

We developed a first version of models for the different feature spaces as follows:

1. We mapped every category of the taxonomy to one entity of *Freebase* (or, in a few particular cases, two or three). For instance we mapped the category `/music/pop` to the *Freebase* entity *Pop Music*.
2. For each category, we analyzed the features of the associated entities, and we determined which of these features were representative of it, *i.e.* present for it and its sub-categories, but not on other categories in the taxonomy. For instance, this would recognize that the *Freebase* type `/music/artist` is representative of the category `/music`, because it is present only in this sub-tree of the taxonomy.
3. We performed a manual curation of this result, in order to correct a few obvious issues.

Then, in order to improve the quality of the models, we developed a trainer. The trainer works as follows:

1. It takes as input a set of entities, and an initial version of the models,
2. It runs the classification algorithm with the initial version of the model, on the given set of entities, and determines for every entity a classification in the taxonomy,
3. It computes, for every feature, the probability that an entity having this feature belongs to a given category.

This process allows computing a new version of the models. It can be executed iteratively (replacing “initial version” and “new version” by “version N” and “version N+1”). The interest of this approach lies in the fact that it allows improving the quality of the model for one feature space using the classification obtained from other feature spaces. As all entities do not have features in all spaces, the overall coverage of the classification gets improved.

Table 3: Metrics for classification of entities

Coverage	Classified entities	74.4 %
	... weighted by subscriptions	86.3 %
Precision	Relevant classifications	95.1 %
	Best classifications	74.2 %

6.5 Metrics

Again, we use both metrics introduced in section 3 to measure the quality of the classification algorithm: coverage and precision.

Coverage. The coverage measures the fraction of entities which have been classified by the algorithm (*i.e.* for which the algorithm produced a non-empty output). As mentioned in section 5.1, only a subset of the entities in the *Freebase* are relevant for *YouTube*. We should measure the coverage only on this subset. For this purpose, we filter out irrelevant entities with some thresholds, especially on the number of videos they annotate and their query volume.

Table 3 summarizes the obtained values. It is relevant to weight the coverage by the number of subscriptions to each entity on *YouTube*, which is representative of its user-interest on the site. The algorithm showed a precision of around 95 % with a coverage of around 85 %. A deeper analysis of the results show some slight discrepancies between the different categories. The most challenging categories are *lifestyle* and *news*, probably because they are more subjective and less taxonomic than other categories like music, gaming or sports.

Precision. Precision measures the amount of relevant classifications produced by our algorithm. For this purpose, we ran human based evaluations on a sample set of entities with a set of questions per entity:

1. The first question asks the rater to confirm that he/she understands what the topic is about. We of course filtered out any entry for which the rater answered “no” to this question.
2. The second question asks the rater to browse the taxonomy and to select which category he/she considers as the best category for the entity. The main purpose of this question is to ensure the rater is aware of the taxonomy tree before answering the third question.
3. The last question asks to rate the classification generated by the algorithm for the entity.

In order to reflect the hierarchical structure of the taxonomy, and our objective to produce as precise classifications as possible, we introduced the notion of *best* classification. A category is said to be the best classification for an entity if it is relevant and none of its sub-categories (if any) would be a relevant classification. (For instance `/sports/racket/tennis` is the best classification for *Rafael Nadal*, while `/sports/racket` and `/sports` are relevant classifications.) Hence, in the third question of the questionnaire, we asked the raters whether the algorithmic classification was best, relevant or wrong.

Table 3 shows the obtained results for this evaluation.

7. CLASSIFYING YOUTUBE CHANNELS

7.1 User-generated channels

User-generated channels are the most widely known type of channels on *YouTube*. These are the channels generated by a human creator (who uploaded a set of videos) or a human curator (who selected a set of videos uploaded by one or several creators). From a data point of view, such a channel consists in textual meta-data (similarly to a video), and a list of videos.

Algorithm. The classification algorithm for user channels works as follows.

First, the textual meta-data of the channel is annotated using the same process as for videos (see section 5). It produces a set of relevant entities for the user channel. The algorithm then considers the categories of these entities, as determined by the classifier of section 6, and computes their distribution.

Second, the videos of the channel are themselves annotated with entities, and mapped to categories. Then, for every category, the algorithm computes the fraction of the videos of the channel which are annotated by an entity of this category. The calculation of the fraction is weighted by the relative number of views of the videos (in the last 30 days, to have a current view of the channel contents), and by the weight of the supporting entity for the video.

These two parts lead to associate weights to every category for the given channel. A top-down traversal of the taxonomy is then performed, same as for entities (see section 6.3), in order to select the most relevant category for the channel. Again, the algorithm includes two additional criteria for improving precision:

- Only classifications which are supported by two sources of information (annotation of the channel meta-data and categories derived from video annotations) are considered,
- Only classifications which are consistent with the most prominent video categories chosen by the users for the videos of the channel.

Another interest of this algorithm lies in the fact that it allows assessing the thematic cohesiveness of a user channel. If a user channel contains videos about unrelated topics, the first and second steps will output weights spread in different sub-trees of the taxonomy, and this can be processed in the third step, either to generate a multiple classification, or to exclude the channel (if non-cohesive channels are undesirable for the application).

Metrics. We use the same metrics to measure the quality of the classification of user channels as for entities (see section 6.5). One slight refinement is that we can weight the metrics, and especially the coverage, by the relative audience of channels.

Table 4 gathers the results, obtained by running the classifier on the whole corpus of *YouTube* channels. As intended, the strongest aspect of the algorithm is the precision: less than 5 % of the produced classifications are not relevant, this allowed using it for user facing products. On the other hand, the coverage remains reasonable, with more than 75 % of the *YouTube* audience in supported languages. (Similarly to videos, section 5.4, high audience channels are easier to classify than low audience ones.) It is worth mentioning

that the target of 100 % is not achievable for two reasons: there exist some channels which are not thematically cohesive, and because we support only 8 languages for the time being.

7.2 Algorithmic channels

Entity-centered channels. In 2012, *YouTube* launched *YouTube collections*⁷, which are video channels algorithmically generated by *YouTube*, and internally known as *algorithmic channels*. An algorithmic channel is created for every entity of interest for *YouTube*. The channel's videos are those annotated by the entity (see section 5), and algorithms are used to generate the feed of videos and other aspects of the channel.

As there is a 1-to-1 mapping from algorithmic channels to entities, classifying algorithmic channels is straightforward using the outcome of the classifier presented in section 6: every algorithmic channel is assigned to the category of its underlying entity.

Channels from blogs. *YouTube* also generates channels from videos embedded in blogs⁸. Each of these channels contains all the *YouTube* videos which are included in the posts of a given blog. These channels are particularly powerful for newsy topics, and for providing context about videos. The algorithm we described in section 7.1 can be applied to blogs, replacing the channel meta-data by the blog meta-data. Quality metrics are similar. In the future, we plan to replace the annotation of the meta-data by a more precise set of entities obtained by passing through the annotation algorithm the whole blog posts. Similarly, blog posts could be used as context information for annotating videos.

7.3 A word about internationalization

As *YouTube* is launched in more than 40 countries, we had to consider internationalization as a core feature of the framework. As mentioned in section 5.1, *Freebase* content is internationalized: it contains entities of interest in all countries, and its textual information is translated in various languages. In the first step of the algorithm (see section 5), before annotating a video or a channel with semantic entities, we use a probabilistic language detector in order to detect its language, and then configure the annotation algorithm with a model for this language. The remaining steps of the algorithm are language-independent, as they manipulate entities and not text.

In order to validate the proper internationalization of the framework, we computed the same metrics for user channels in other supported languages as in English (see section 7.1), and obtained similar results.

8. APPLICATIONS

8.1 Channels browser

When we decided to start the development of the taxonomic classifier described in sections 6 and 7, the first application we had in mind was the channels browser of *YouTube* (we already introduced it in section 1.1). The classification

⁷<http://youtube-global.blogspot.com/2012/05/finding-and-following-new-channels-you.html>

⁸<http://youtube-global.blogspot.com/2011/06/as-seen-on-youtube-pages-celebrating.html>

Table 4: Metrics for classification of user channels

		All channels	English channels	French channels
Coverage	In fraction of the number of channels	51.6 %	70.0 %	76.6 %
	In fraction of the audience	70.0 %	75.1 %	76 %
Precision	Relevant classifications	95.7 %	—	—
	Best classifications	65.7 %	—	—

of channels in this browser is directly derived from the classification we algorithmically compute. The only adaptation is that we map the internally used categories to more user friendly ones.

Beyond classification, the channels browser requires to *rank* the channels that appear under a category. This ranking is of major importance, as the user interface can show only a limited number of channels for a category, and the user will probably not browse the complete list. The ranking should ensure that channels which are most likely to interest the user appear on top, and also ensure some diversity and variety in what the user sees. For the first aspect, we decided to rank channels according to their recent audiences (number of views and number of subscriptions in the last 30 days). As the largest audience channels of *YouTube* are mainly from the US, we decided to ensure the promotion of country-specific channels by:

1. Computing for every channel the cross-country distribution of its audience,
2. Identifying the countries of specific interest of a channel by comparing this distribution to the overall distribution of *YouTube* audience,
3. Boosting the score of every channel in these countries.

This leads to a list of channels per category and per country. We ensure diversity and variety by two ways:

- For non-leaf categories, we normalize the audience of the channels between the different sub-categories, in order to ensure some representation of each of them (for instance, we want the *Sports* category to include both *Soccer* and *Tennis* channels, even if the channels of the former category have a larger audience than those of the latter).
- We introduced a random seed in the ranking function, re-initialized every day, so that different channels show on top of every category.

In the future, we plan to customize the ranking of channels within each category, as well as the ranking of categories in the list, using information from user profiles.

As explained in section 1.1, the channels browser was first launched with a manual selection of channels for each category, and then updated with the results of the algorithmic classification. This allowed to compare the performance of both in terms of providing interesting content to end-users, and to assess the overall framework we developed. We globally observed an increase of around 100 % to the subscription rate together with the launch of the algorithmic classification, independently of any other change, confirming the interest of the approach.

8.2 Broad algorithmic channels

Among the hundreds of thousands of algorithmic channels *YouTube* has (see section 7.2), some of them are of particular interest for users to browse the contents of *YouTube*. For instance, the algorithmic channel associated with the *Music* entity should be a great entry point for all users interested by music on *YouTube*, and the same applies for instance for *Pop Music*, *Classical Music*, *Sports* and *Tennis*.

As mentioned in section 7.2, algorithmic channels are generated by inverting the mapping of videos to entities obtained by the annotation process. However, the quality of the contents produced by this approach is not as good as expected for these broad channels. The main reason is the lack of hierarchical inheritance in the annotation process. For instance, the best music videos will probably not be annotated by the entity *Music*, but by more precise entities like *Lady Gaga* or *Bad Romance*.

We use the taxonomic classification of entities introduced in section 6 to solve this problem. Thanks to this classification, we know that *Lady Gaga* is about music, hence we are able to deduce that (most) videos about *Lady Gaga* are about *Music*. More precisely, we first mapped all the categories of the taxonomy to entities (*e.g.* we mapped the category /music to the entity *Music*). Some categories did not correspond to any actual entity. We simply ignored them. Then, we developed an algorithm that computes the distribution of the categories annotating a video (as output by the algorithm of section 5), and adds annotation for the most prominent entities. We evaluated the quality of the added annotations using the same metrics as those described in section 5.4, and got comparable results.

8.3 YouTube public Data API

The classification of channels used in the channels browser is publicly exposed via the public *YouTube* Data API⁹. This allows users to develop new applications on top of this data, we hope the current paper will help this.

9. CONCLUSION

This paper described a complete framework for classifying channels in *YouTube*, from the definition of the product to its actual implementation. Along the technical description, we explained our practical approach for developing and evaluating such a product. This system is currently running daily on the whole *YouTube* corpus, and serving several user facing applications. It has been evaluated in term of precision and coverage, as well as by its performance after launch. To the best of our knowledge, this represents the first large scale application of thematic video content classification on an Internet site.

⁹<https://developers.google.com/youtube/v3/docs>

As further work, we plan to improve the quality and coverage of the classification algorithm, especially by improving the way the features from the different models are combined. We also plan to expand the taxonomy tree in an algorithmic manner, especially by using clustering techniques and extracting further information from *Freebase*. We believe it would not have been worth the effort to build the root of the taxonomy algorithmically (in order to obtain a result that would match user expectations), but that sub-categories like music genres or sport disciplines can be derived algorithmically. We also plan to investigate the possibility to assign several categories to a user channel, when its content is multi-thematic. Last, we plan to expand our language support to cover all languages supported by *YouTube*.

10. ACKNOWLEDGMENTS

The project to develop an algorithmic classifier of *YouTube* channels has been initiated by Palash Nandy and Fabio Lopian, and supported by Mur Viswanathan as product manager. Rich Washington and his team developed the video annotation framework introduced in section 5. Bill Saphir mastered the launch of the algorithmic classifications on the Channel Browser. Fabio Lopian and Cristos Goodrow thoroughly reviewed drafts of this paper.

The contents of this paper submitted for publication are covered by several patent applications.

11. REFERENCES

- [1] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the SIGMOD Conference*, 2008.
- [2] Razvan C. Bunescu, Marius Pasca, and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the EACL conference*, 2006.
- [3] Andras Csomai, Rada Mihalcea, and Rada Mihalcea. Linking documents to encyclopedic knowledge. 2008.
- [4] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the EMNLP-CoNLL conference*, 2007.
- [5] Ofer Dekel, Joseph Keshet, and Yoram Singer. Large margin hierarchical classification. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, New York, NY, USA, 2004. ACM.
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Proceedings of the CVPR conference*, 2009.
- [7] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, Ramanathan V. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, Jason Y. Zien, and Jason Y. Zien. Sementag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the WWW conference*, 2003.
- [8] Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, Jean Ponce, and Larry Ponce. Automatic annotation of human actions in video. In *Proceedings of the ICCV conference*, 2009.
- [9] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach. (Second edition)*. Pearson Education Inc., 2011.
- [10] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, Larry S. Davis, and Larry S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *Proceedings of the CVPR conference*, 2009.
- [11] Xianpei Han, Le Sun, Jun Zhao, and Jun Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the SIGIR conference*, 2011.
- [12] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, Gerhard Weikum, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the EMNLP conference*, 2011.
- [13] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, Soumen Chakrabarti, and Soumen Chakrabarti. Collective annotation of Wikipedia entities in web text. In *Proceedings of the KDD conference*, 2009.
- [14] Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl.*, 7(1), June 2005.
- [15] Marcin Marszalek, Ivan Laptev, Cordelia Schmid, and Cordelia Schmid. Actions in context. In *Proceedings of the CVPR conference*, 2009.
- [16] David N. Milne, Ian H. Witten, and Ian H. Witten. Learning to link with Wikipedia. In *Proceedings of the CIKM conference*, 2008.
- [17] Vicente Ordonez, Girish Kulkarni, Tamara L. Berg, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In *Proceedings of the NIPS conference*, 2011.
- [18] Yang Song, Ming Zhao, Jay Yagnik, and Xiaoyun Wu. Taxonomic classification for web-based videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010.
- [19] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, Washington, DC, USA, 2001. IEEE Computer Society.
- [20] George Toderici, Hrishikesh Aradhye, Marius Pasca, Luciano Sbaiz, Jay Yagnik, and Jay Yagnik. Finding meaning on youtube: Tag recommendation and category discovery. In *Proceedings of the CVPR conference*, 2010.
- [21] Cornelis Joost "Keith" van Rijsbergen. *Information Retrieval*. Butterworth, London, Great Britain; Boston, Massachusetts, 1979.
- [22] Weilong Yang, George Toderici, and George Toderici. Discriminative tag learning on youtube videos with latent sub-tags. In *Proceedings of the CVPR conference*, 2011.
- [23] Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y. Chang, and Xiaoyan Zhu. Entity disambiguation with Freebase. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, December 2012.