VideoCLEF 2008: ASR Classification based on Wikipedia Categories

Jens Kürsten, Daniel Richter and Maximilian Eibl
Chemnitz University of Technology
Faculty of Computer Science, Dept. Computer Science and Media
09107 Chemnitz, Germany
[jens.kuersten | daniel.richter | maximilian.eibl] at cs.tu-chemnitz.de

Abstract

This article describes our participation at the *VideoCLEF track* of the CLEF campaign 2008. We designed and implemented a prototype for the classification of the Video ASR data. Our approach was to regard the task as text classification problem. We used terms from Wikipedia categories as training data for our text classifiers. For the text classification the Naive-Bayes and kNN classifier from the WEKA toolkit were used. We submitted experiments for classification task 1 and 2. For the translation of the feeds to English (translation task) Google's AJAX language API was used. The evaluation of the classification task showed bad results for our experiments with a precision between 10 and 15 percent. These values did not meet our expectations. Interestingly, we could not improve the quality of the classification by using the provided metadata. But at least the created translation of the RSS Feeds was well.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing

General Terms

Measurement, Performance, Experimentation

Keywords

Automatic Speech Transcripts, Video Classification

1 Introduction

In this article we describe the general architecture of a system for the participation at the *VideoCLEF track* of the CLEF campaign 2008. The task was to categorize dual-language video into 10 given classes based on provided ASR transcripts [2]. The participants had to generate RSS Feeds that contain the videos for each of the 10 categories. The content of the RSS items for each of the videos was also given[1].

Our approach to solve the problem mainly relies on the application of a text classifier. We use the textual content of Wikipedia[2] categories that are equal or at least highly related to the 10 given categories. The classification of the ASR transcripts will be done by classifiers from the WEKA toolkit [3].

The remainder of the article is organized as follows. In section 2 we describe our system and its architecture. In section 3 we present the results of our submitted experiments. A summary of the result analysis

[1] http://ilps.science.uva.nl/Vid2RSS/Vid2RSS08/Vid2RSS08.html
[2] http://en.wikipedia.org

is given in section 4. The final section concludes our experiments with respect to our expectations and gives and outlook to future work.

2 System Architecture

The general architecture of the system we used is illustrated in figure 1. Besides the given input data (archival metadata, ASR transcripts and RSS items) we used a snapshot of the English and the Dutch Wikipedia as external training data. We extracted terms related to the given categories by applying a category mapping. These extracted terms were later used as training data for our text classifiers. In the following subsections we describe the components and operational steps of our system.

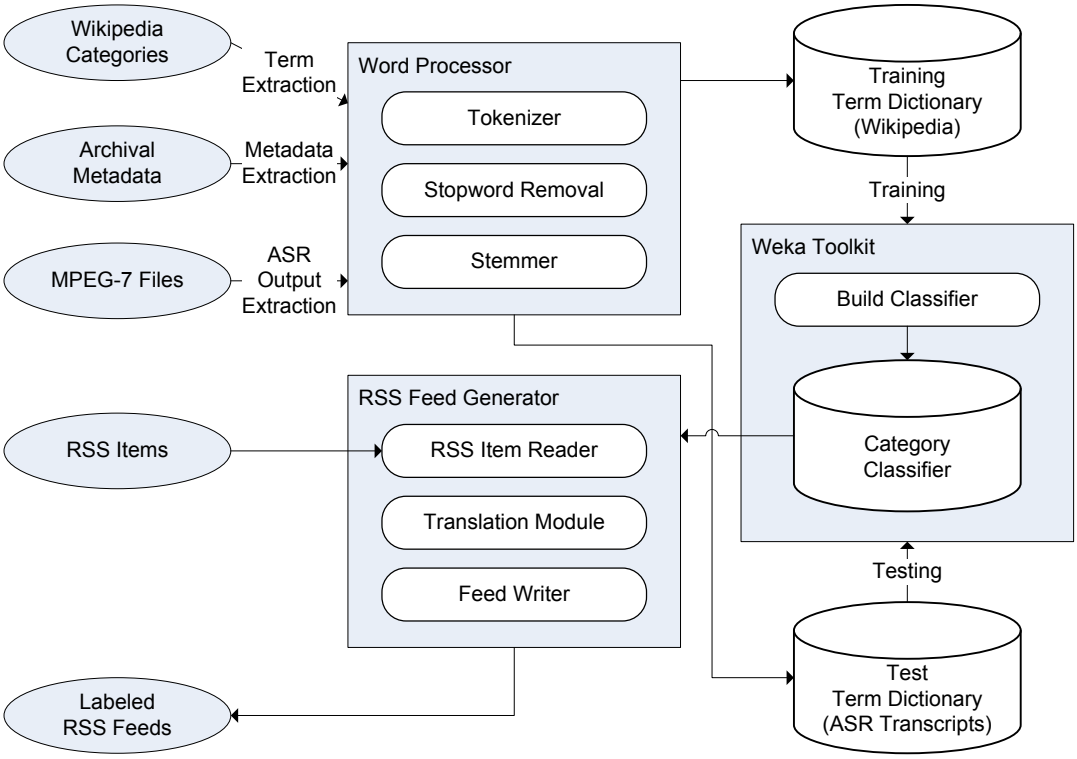

Figure 1: General System Architecture

2.1 Classifier Training

The training of the classifier consists of three essential steps that will be explained in the subsections below. At first a fixed number of terms were extracted by using the JWPL library [4] for each of the 10 categories. These terms were then used to train two classifiers of the WEKA toolkit. Namely the Naive-Bayes and the kNN (with k = 4) classifier were used. In the last step of the training the classifiers were stored because they should remain available for the later classification step.

2.1.1 Wikipedia Term Extraction

Before the extraction of the terms was done, we needed to specify a mapping between the two source language categories and the available Wikipedia categories. The specified categories formed the starting points for the Wikipedia term extraction procedure. The final mapping is presented in table 1.

Table 1: Category mapping for Dutch and English: Specified - Wikipedia

id	*specified NL cat.*	*mapped NL cat.*	*specified EN cat.*	*mapped EN cat.*
0	archeologie	archeologie	archeology	archaeology
1	architectuur	architectuur	architecture	architecture
2	chemie	scheikunde	chemistry	chemistry
3	dansen	dans	dance	dance
4	film	film	film	film
5	geschiedenis	geschiedenis	history	history
6	muziek	muziek	music	music
7	schilderijen	schilderkunst	paintings	paintings
8	wetenschappelijk onderzoek	wetenschap	scientific research	scientific research
9	beeldende kunst	beeldende kunst	visual arts	visual arts

2.1.2 Training Set Creation

To create a training set we extracted a specified number (TMAX) of unique terms from both Wikipedia snapshots by using the JWPL library[3]. This maximum number of terms is one of the most important parameters of the complete system. We have conducted several experiments with different values for TMAX, varying from 3000 to 10000 (see section *Evaluation* for more details). Since the extraction of the terms is very time consuming due to the large size of the Wikipedia we also stored the training term dictionaries (TRTD) for the categories and for different variations of the parameter TMAX. The training term dictionaries consist of a simple term list with term occurrence frequencies.

Another important parameter of the system and also for the creation of the TRTDs is the depth (D) we use to descend in the Wikipedia link structure. The maximum size of each TRTD directly depends on the parameter D, because only when we descend to a certain depth in the linking structure of the Wikipedia category tree we could extract a sufficient number of unique terms.

2.1.3 Word Processing

Before the extracted terms were added to the TRTD, they were processed by our word processor (WP). The word processor simply applied a language-specific stopword list and reduced the term to its root with the help of the Snowball stemmers[4] for English and Dutch.

2.1.4 TRTD Balancing

After our first experiments with the creation of the TRTDs for all 10 categories we discovered, that the TRTDs were unbalanced with respect to the number of unique terms. This is due to the fact that the categories have different total numbers of sub-categories and these again contain different amounts of terms. To avoid that some categories will get a large weight because of a high TMAX that could never be satisfied by a category with a smaller number of pages, we decided to implement two different thresholds to balance the TRTDs in terms of their size. The first strategy was simply to use the term amount of the smallest category as TMAX, but it turned out that this creates bad classifications when TMAX and D are small. So we decided to use the mean of the term amounts of all 10 categories, which means that some categories might have a too small number of terms, but in general the TRTDs are balanced.

2.1.5 TRTD Discrimintation

For a better discrimination of the categories we implemented a training term duplication threshold (WT). This threshold is used to delete terms from the TRTDs that occur in at least (WT) categories. We assumed that this might help during the classification step. Our idea is that a natural term distribution that can be

[3]http://www.ukp.tu-darmstadt.de/software/jwpl
[4]http://snowball.tartarus.org

found in the Wikipedia could not be categorized very well. By implementing this assumption we hoped to improve the precision of the classification.

Another parameter that might be useful for the discrimination of the TRTDs is the frequency based selection (FS) of the terms. As mentioned before we selected a maximum number of terms (TMAX) for each category. We could use different strategies for that because the TRTDs most likely contain much more terms than we may want to extract. We implemented two options for the selection of the terms. The first is just to use the terms with the highest occurrence and the second is to take the average term occurrence frequency and to extract 0.5 times TMAX of the terms above and below this average.

2.1.6 TRTD Term Statistics

Table 2 represents the TRTD term statistics for all categories (columns 1-10) depending on selected parameter settings for the discrimination threshold (WT) and the depth of the linkage extraction (D) for the English Wikipedia snapshot. We marked the category with the minimum and maximum amount of terms for each of the configurations. It is obvious that the amount of terms increases for all categories when the depth for

Table 2: Training Term Dictionary Statistics

WT	*D*	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	2	3064	5658	**7999**	4073	5359	6062	6781	**408**	7611	7996
2	2	680	688	685	681	690	689	689	**93**	689	**3712**
4	2	1181	1471	1445	1275	1470	1509	1437	**268**	4404	**4777**
5	2	1396	1830	1810	1528	1841	1930	5015	**300**	5049	**5461**
6	2	1576	2178	2152	1748	2176	4607	5533	**340**	5743	**6109**
8	2	1972	3138	3142	4017	5223	5606	6651	**396**	7137	**7612**
0	3	8804	19404	16591	10003	15874	23210	29230	**1178**	**32720**	24694
2	3	2190	2247	2226	2207	2237	2246	2243	**480**	2247	**9259**
4	3	3648	4853	4368	3816	4675	4974	4988	**831**	**18906**	12778
5	3	4160	6112	5186	4390	5746	6367	21029	**946**	**21343**	15364
6	3	4677	7468	6041	4936	6816	17713	23494	**1060**	**24786**	18028
8	3	6193	11143	8190	9944	15715	21632	28726	**1172**	**29963**	22869
0	4	18115	36674	27080	14024	42627	81622	75143	**1284**	**96156**	56295
2	4	3838	3881	3847	3822	3882	3883	3883	**653**	3883	**15340**
4	4	7148	9974	8106	6735	9886	10408	10359	**974**	**50259**	24318
5	4	8356	13150	9612	7774	13161	14800	52403	**1062**	**57710**	31773
6	4	9588	16326	10976	8622	16350	63649	62477	**1200**	**72852**	41990
8	4	13572	25228	15168	14013	42465	76956	74530	**1284**	**88037**	54257

descending in the link structure (D) is increased. In the English Wikipedia the category *science* contained the largest amount of terms, followed by *history*, *music* and *visual arts*. The smallest amount of terms could be extracted for the category *paintings*. In our opinion the statistic allows to draw the following conclusions for the parameter sets. With WT=2, i.e. that all term duplicates were removed that occur in at least two category TRTDs, we could create the most balanced TRTDs. All parameter sets with WT¿3 create TRTDs with more realistic term distributions.

For the term statistics of the Dutch Wikipedia one could draw similar conclusions, but there are some differences. The most important difference is the smaller number of entries in the Dutch Wikipedia, which generally results in smaller TRTDs. Also the distribution of the specified categories is little different. There are no outliers like *science* or *paintings*, which consequently follows from the smaller amount of pages. For the Dutch Wikipedia the category *dans* produced the smallest TRTD.

2.1.7 Training Setup

In the first step of the classifier training process we loaded the relevant TRTD for each category. Thereafter, we fed the instances of the TRTD into the Naive-Bayes and kNN classifiers. Finally, the classifiers were

stored, because we wanted them to remain for further evaluations of different parameter sets for the complete system.

2.2 Test Set Creation

For the preparation of the classification it was necessary to parse the ASR transcripts and to extract the textual information. We also parsed the metadata that could be used for the classification task 2. We used the same procedure for the creation of the test term dictionaries (TSTD) as we did before for the creation of the TRTDs. At first the word processor removes stopwords and then it stems all terms to their root. For the TSTDs we also applied a parameter (VT) for the removal of duplicate terms. We hoped this would help in discriminating the ASR transcripts.

2.3 Classification

In the classification process the stored classifiers were reloaded into memory. They were then used to classify the contents of the TSTD for each video. The results of the classification are 10 probabilities for the membership of the video in all of the 10 specified categories. These probabilities sum up to 1 for each term of the TSTD. This was repeated for all terms in the TSTD in order to get a final classification. For the classification task 2 we also used the terms from the metadata files for classification.

As next step we normalized the returned classifications, i.e. each of the 10 specified categories were normalized to find the final classification of the videos. The normalization is defined as the sum of the arithmetic mean and the standard deviation of each category. This sum was used as final classification threshold (CT) for each corresponding category.

In the last step the final classification was created. Therefore we iteratively decreased a predefined score (S), which is always larger than CT, until at least one of the ten CT values is larger than S. Finally, we compared the resulting S with all CT for the 10 categories and assigned the corresponding classification to the video.

2.4 RSS Feed Creation

The RSS Feeds were created continuously during the last step of the classification. Thereby, the RSS item for each video was subsequently added to the corresponding category RSS Feeds.

2.5 RSS Feed Translation

The translation of the RSS Feeds was conducted when all categories were complete. For the translation we used Google's AJAX Language API[5], which is the actual translation component of the *Xtrieval* framework [1]. The translation was technically limited to a maximal amount of 100 characters per time. Therefore we split the Feed contents into sentences and translated these. Thereafter we rebuilt the RSS Feed in the translated language.

3 Evaluation

This section provides experimental results on the development and test sets. At first, we describe the determination of the parameters by using the development set and finally we present the setup of the complete system for the experiments on the test set.

3.1 Parameter Tuning with Development Data

We used the development data for the tuning the parameter set of our system for the experiments on the test data. The system has six important parameters:

[5] http://code.google.com/apis/ajaxlanguage/documentation

- Depth of Wikipedia Category Extraction (D)
- Frequency-based Selection of Training Terms (FS); 0 for high frequency terms and 1 for mid frequency terms
- Maximum Number of Training Terms (TMAX)
- Training Term Duplicate Deletion (WT); 5 for deletion of terms that appear in at least 5 categories
- Test Term Duplicate Deletion (VT); 5 for deletion of terms that appear in at least 5 video ASR transcripts
- Classifiers (C); we used both Naives-Bayes and kNN (k = 4) for all experiments

Table 3 shows selected experiments on the development data. We chose the best performing parameter sets for different sizes (TMAX) of the training term dictionaries. For the evaluation of the performance we used the mean average precision (MAP) over the 10 specified categories.

Table 3: Selected Experiments on the Development Data

TMAX	*FS*	*D*	*WT*	*VT*	*MAP*
10000	0	3	2	5	0.4
10000	0	3	2	7	0.33
10000	0	2	9	9	0.18
5000	0	3	2	5	0.4
5000	1	2	2	2	0.4
5000	1	2	8	2	0.34
3000	0	3	2	5	0.4
3000	1	2	2	2	0.4
3000	0	5	9	2	0.37
1000	1	2	2	7	0.51
1000	0	5	9	2	0.48
1000	1	2	2	2	0.48

We derived two possibly useful parameter sets from table 3. At first for large TRTDs with TMAX > 3000 the parameter set (0;3;2;5) seemed to be promising. For smaller TRTDs with TMAX ≤ 3000 the parameter set (1;2;2;2) could be useful. Unfortunately, we tested the configuration with TMAX = 1000 after the deadline of the submission.

3.2 Experimental Setup and Results

We submitted two experiments for each of the two classification tasks. The results of the evaluation are presented in table 4.

Table 4: Experimental Results based on the Evaluation Data

TMAX	*FS*	*D*	*WT*	*VT*	*P*	*R*
3000	0	3	2	5	0.15	0.14
5000	0	4	5	5	0.10	0.12
3000	0	3	2	5	0.13	0.12
5000	0	4	5	5	0.12	0.14

The results were not very well and did not meet our expectations and observations on the development data. Interestingly, using the metadata in classification task 2 did not improve the classification performance in both cases.

Additionally, we submitted a translation of the RSS Feeds. The translation was evaluated by three assessors in terms of fluency (1-5) and adequacy (1-5). The higher the score the better was the quality of the translation. The results are summarized in table 5.

Table 5: Assessment of the Translation

Criterion	*Ass. 1*	*Ass. 2*	*Ass. 3*	*Average*
fluency	2.88	2.65	2.93	2.82
adequacy	3.53	3.15	3.80	3.49

4 Result Analysis - Summary

The following items conclude our observations of the experimental evaluation:

- *Classification task 1:* The quality of the video classification was not as good as expected, both in terms of precision and in terms of recall.
- *Classification task 2:* Surprisingly, the quality of the video classification could not be improved by utilizing the given metadata. The reason for that might be the small impact of the metadata in comparison to the large size of the TRTD we used.
- *Translation task:* The translation of the RSS Feeds was quite good, but there is also room for improvement, especially in terms of fluency.

5 Conclusion and Future Work

The experiments showed that the classification of dual-language video based on ASR transcripts is a quite hard task. Nevertheless, we presented an idea to tackle the problem. But there are a number of points to improve the system. The two most important problems are the size of the training data on the one hand and the balance of the categories on the other hand. We consider to omit the TRTD balancing step and to shrink the TRTD size in further experiments. Another point might be to weight the TRTD based on an approximated distribution of the categories in the video collection, because this could be a good indicator on how to find the correct classes for a given video.

References

[1] Jens Kürsten, Thomas Wilhelm, and Maximilian Eibl. Extensible retrieval and evaluation framework: Xtrieval. *LWA 2008: Lernen - Wissen - Adaption, Würzburg, October 2008, Workshop Proceedings*, October 2008, to appear.

[2] Martha Larson, Eamonn Newman, and Gareth Jones. Overview of videoclef 2008: Automatic generation of topic-based feeds for dual language audio-visual content. *CLEF 2008: Workshop Notes*, September 2008.

[3] Ian H. Witten and Eibe Frank. *Data mining : practical machine learning tools and techniques*. Elsevier, Morgan Kaufman, Amsterdam, 2. ed. edition, 2005.

[4] Torsten Zesch, Christof Müller, and Iryna Gurevych. Extracting lexical semantic knowledge from wikipedia and wiktionary. *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, May 2008.