UPMC/LIP6 at ImageCLEF's WikipediaMM: An Image-Annotation Model for an Image Search-Engine

Ali Fakeri-Tabrizi, Massih-Reza Amini, Sabrina Tollari, Patrick Gallinari
Université Pierre et Marie Curie-Paris6
Laboratoire d'Informatique de Paris 6 - UMR CNRS 7606
104 avenue du président Kennedy
75016 Paris, France
firstname.lastname@lip6.fr

Abstract

In this paper, we present the LIP6 retrieval system which automatically ranks the most similar images to a given query constituted of both textual and/or visual information through a given textual-visual collection. The system first preprocesses the data set in order to remove stop-words as well as non-informative terms. For each given query, it then finds a ranked list of its most similar images using only their textual informations. Visual features are then used to obtain a second ranking list from a manifold and a linear combination of these two ranking lists gives the final ranking of images.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 [**Database Management**]: Languages—*Query Languages*

General Terms

Measurement, Performance, Experimentation

Keywords

Visual Information Retrieval, TF-IDF, Language Model, Ranking on Data Manifolds

1 Introduction

ImageCLEF's WikipediaMM uses image collection created and employed by the INEX competition [1]. This image collection contains approximately 150,000 images that cover diverse topics of interest. These images are associated with unstructured and noisy textual annotations in English. There are 75 topics as the query to be searched on the collection. Each topic consists of textual data (and/or sample images and/or concepts) describing a user's (multimedia) information need. These data are given as the XML files in each one there is image textual information such as image title, image concept, image location path and a short text which describes the desired results, i.e it explains what could be the relevant result and what is the irrelevant results. The aim of this task is to try to find as many relevant images as possible from the Wikipedia image collection for a given textual-and/or-visual query.

In section 2, we propose a strategy to perform the retrieval process. Section 3 describes the models we used in our experiments and in section 4 we show the results we obtained.

2 Retrieval Strategy

We have a collection of textual-visual documents constituted of images and their corresponding textual notes. The aim is hence to find a list of ranked documents for each query in a such way that the most relevant documents have the highest ranks. A query can be in a textual or both textual and visual formats. In order to benefit from the whole given data, we introduce a retrieval strategy which combines two lists of ranked documents obtained separetely from textual and image informations for each given query. If a query is just described by a text, the system provides the ranked list obtained from the first textual strategy. In the following section we detail these different retreival strategies.

3 Models

The textual strategy is based on the *Term Frequency-Inverse Document Frequency Model* [2] and the *Language Model* [3]. We run each of these models separately and then combine linearly their outputs. This stage follows a preprocessing step in which stop-words from the Glasgow University [4] stop-list are removed. We then use the porter stemming algorithm to reduce terms into their common stem.

3.1 Term Frequency-Inverse Document Frequency Model

In the TF-IDF model [2], each document in the collection and a query are represented by their associated vector of the length of the vocabulary. Each term w_i in the vocabulary has a real-valued weight in x, where x is a document or a query, equal to the product of $tf(w_i, x) \times idf(w_i)$ where

$$tf(w_i, x) = \frac{|\{w|w \in x \cap w = w_i\}|}{|\{w|w \in x\}|}, idf(w_i) = log\left(\frac{1 + |D|}{1 + |\{d \mid |w_i \in d\}|}\right)$$

and $|D|$ representes the total number of documents in the collection.

3.2 Language Model

In the Language Model [3], a query $Q = [q_1 \dots q_n]$ is considered to be generated by a probabilistic model based on a given document $D = [d_1 \dots d_m]$ and the aim is to estimate the strength of this generation via $P(D|Q)$. The Bayes formula lets us extend the desired probability as follows:

$$P(D|Q) = p(Q|D)P(D).$$

At this stage, we suppose that the distribution of documents in the corpus is uniform and that document terms are independant which gives:

$$P(D|Q) \propto \prod_{q_i \in Q} P(q_i|D)$$

This model is also known as the unigram model. By dissociating query terms occuring or not in D we get:

$$P(D|Q) \propto \prod_{q_i \in D} P_s(q_i|D) \times \prod_{q_i \notin D} P_u(q_i|D) \tag{1}$$

P_s is the probability model of occuring query terms in D and P_u is the probability model no-occuring terms. $P_u(q_i|D)$ could be estimated by $\alpha_d \times P(q_i|C)$ in which α_d is a constant factor and

C is the collection of documents. Documents are then ranked according to the logarithm value of equation (1):

$$\log\ P(Q|D) = \sum_{q_i \in D} log\ \frac{P_s(q_i|D)}{N(q_i,d)} + |q_i| \times log\ \alpha_d + \sum_{q_i \notin D} log\ P(q_i|C).$$

Where $N(q_i, d)$ is a normalization factor. For the estimation of $P_s(q_i|D)$, we used the Jelinek-Mercer smoothing approach[5]:

$$P_s(q_i|D) = (1-\lambda)P_{ml}(q_i|D) + \lambda P(q_i|C)\ , \lambda \in [0,1]$$

in which P_{ml} is the maximum likelihood estimation of the probability of presence of a query term q_i in D

$$P_{ml}(q_i|D) = \frac{|q_i \in D|}{\sum_{j=1\ldots N} |q_j \in D|}.$$

3.3 Ranking on Data Manifolds

The goal of using Data Manifolds [6] is to rank the data with respect to their intrinsic global manifold structure revealed by their visual information. For many real world data types this model is superior to a local method, which rank data simply by pairwise Euclidean distances or inner products. The idea of model is as follows. First, we form a weighted network on the data (using a simple Kruskal's algorithm [7] by the weights corresponding to the Euclidean distances), and assign a positive ranking score to each query, ranked with respect to the queries, and zero to the rest. Afterward, all points spread their ranking score to their neighbors via the weighted network. The spread process is repeated until a global stable state is achieved, and all points except queries are ranked according to their final ranking scores.

4 Experimental Results

4.1 Preprocessing Phase

As the first step of preprocessing, all unuseful characters should be removed, like non-alphabetic ones. As we are sure that in Wikipedia task there exist only the English characters, we delete also the non-English characters. In next step of preprocessing, each word in the corpus existing in the anti-dictionary will be deleted. As third step, we try to retrieve the keywords and verbs' roots hidden in the combined terms. In this task, all the words appeared in the topics will create our keywords set. The idea is based on searching substrings in each word or combined term to find the hidden keywords through them. As we are sure that in Wikipedia task there exist only the English characters, we delete also the non-English characters.

In next step of preprocessing, each word in the corpus existing in the anti-dictionary will be deleted. As third step, we try to retrieve the keywords and verbs' roots hidden in the combined terms. In this task, all the words appeared in the topics will create our keywords set. Our strategy based on searching substrings in each word or combined term to find the hidden keywords through them.

After this step, we remove the words with an appearance frequency less than 10 times in the corpus unless the word is a keyword. The reason why we do this step is to reduce the size of the corpus to make it easier to process the documents in the models. We lost a part of information, but it is not so important because we try to delete the least frequent words which have usually the least importance.

Afterward, we remove all documents containing less than 5 words for the reason that they have not so much information to be important in models' process. Here also, we prevent the documents containing any keywords being removed. During the preprocessing, we create the dictionary (vocabularies list) and we also index each document respectively to the dictionary to facilitate future calculation. Dictionary is created simply as a list of two columns: first column contains the words existing in the obtained corpus from the latter step. In second column, each line indicates the times of appearance of the word located in the corresponding line in first column.

4.2 Description of the runs

All runs are fully automatic and do not use feedback nor query expansion. The visual features are composed of a simple global HSV histogram (8x3x3) and of the standard deviations of H, S and V. The visual vectors are so composed of 17 dimensions.

Run1 TFIDF A textual approach using TF-IDF. We used exclusively the text information here. Each topic text (query) and document were characterized in the bag-of-word space using a TF-IDF encoding. By performing a cosines-similarity between the given query and documents, we had a value for each document represents the relevance. Descending sort on documents upon these obtained values made the most similar documents be ranked higher. As the final result, we chose the 2000 first documents in the ranking list.

Run2 LM A textual approach using Language-Model. We used exclusively the text information again but this time we perform another textual model. For each topic we estimated the probability that every document in the collection generates the query using a unigram language model as explained in models section. At the end, the probability calculated for each document was counted as the relevance value. By sorting these values we obtained the ranked list of documents and we got the N first documents as the final result.

Run3 TFUSION_TFIDF_LM A textual approach using a fusion of the results of Language-Model and TF-IDF. For each query q, we obtained two lists of ranked documents according to the TF-IDF and the Language Model approaches as calculated in Run1 and Run2. The final ranked list for q is based on the fusion of the rank values upon these two lists. As the fusion in this part, we got 50% of each ranked list to combine.

Run4 visualonly A visual approach using Image only. We used exclusively the image information here. Each topic image (query) and document were characterized using global HSV image features. For each query, its most similar documents in the sense of the euclidean measure were chosen. We did not provide any results for those topics without an associated image query.

Run5 TIFUSION_LMTF_COS A texto-visual approach using Fusion Text-image (1). For each topic, we ranked documents upon two approaches considering only the text information as done in Run3. We got the 2000 first documents and try to re-rank them upon their similarity using image features, i.e the final rank of documents was obtained by re-sorting the 2000 first documents upon their similarity to the image(s) of the given topic. We used the Euclidean distances as the similarity measure.

Run6 TI_LMTF_MANIF A texto-visual approach using Fusion Text-image (2). For each topic, we ranked documents upon two approaches considering only the text information as done in Run3. We got the 2000 first ranked document and try to re-rank these 2000 document upon their similarity using image features. but here, we used manifold-based technique in place of the simple Euclidean distance. Briefly, we create the network of data manifold on base of these N documents, through them we search the most relevance to query and the values of data manifolds model was the final rank of documents.

Run7 TIFUSION_LMTF_MANIF A texto-visual approach using Fusion Text-image (3). In this part, we used two ranking lists obtained from Run5 and Run6 and the final ranking list was obtained from the fusion of rank values of these two lists. In the fusion in this part, we took 60% as the weight factor for the ranked list of Run5 and 40% as the weight factor for Run6.

Run	Modality	MAP	P@5	P@10	Number of retrieved documents	Number of relevant retrieved documents
1 TFIDF	TXT	0.113	0.219	0.192	30863	1798
2 LM	TXT	0.108	**0.253**	0.197	74990	1813
3 TFUSION_TFIDF_LM	TXT	**0.119**	0.240	0.216	30632	1792
4 visualonly	IMG	0.001	0.011	0.008	41986	157
5 TIFUSION_LMTF_COS	TXTIMG	0.105	0.227	**0.227**	30632	1795
6 TI_LMTF_MANIF	TXTIMG	0.074	0.192	0.192	14710	1283
7 TIFUSION_LMTF_MANIF	TXTIMG	0.105	0.229	0.224	30632	1797

Table 1: WikipediaMM 2008 results. All runs are automatic and used no feedback no extension.

4.3 Results and Discussion

Table 1 shows that Language Model (run2) gives better P5 and P10 scores than TFIDF model (run1). The reason why the probabilist model works better may be because WikipediaMM task is a case in which there are not the same keywords existing in query and in our database. Using a combination of results of TFIDF and Language Model (run3) leads to an improvement, because we can benefit of strength points of both models. The visual only run (run4) - we have submitted it as a visual base line - gives, as expected, the worst results and shows how difficult is the image retrieval task with visual features only. The ranking on data manifold model (run7), as shown in table 1, gives similar results than euclidean ones (run5). The reason why this model appeared not better than euclidean ones - as expected - is that it is more sensitive on feature extraction manner. Finally, the texto-visual approaches improve slightly P10 scores compare to text scores, but not MAP scores.

5 Conclusion

In this working note, we focus our efforts on the study of how to find automatically the most similar images through a given textual-visual collection to a given query which is a collection of textual and/or visual information. First, we perform a preprocessing on the textual part, then for each given query, we use the textual information retrieval models which give us ranking lists. We also use the visual features to obtain another ranking list. A linear combination of these ranking lists give us the new ranking lists which benefit both of visual and textual part of information. As a new idea, we also perform the method of ranking on data manifold in place of using euclidean distance to find the similarity.

As for several participants of WikipediaMM 2008, our text runs obtain better MAP than texto-visual runs, but our texto-visual runs give better P10 scores, then using a combination of textual and visual information seems, in our case, improving the first results. That shows the difficulty to obtain a good texto-visual fusion. We use the same TF-IDF model and the same Language Model in ImageCLEFphoto 2008 as in WikipediaMM2008. Contrary to WikipediaMM task, in ImageCLEFphoto 2008, these two models give similar scores when the query is composed of the title words (see [8] for more details). The ranking on data manifold seems a promising way, but we have to improve our feature extraction manner.

As perspectives, we will study how the feature extraction manner influence the results obtain by the data manifold model. We will also consider the different parameters of language model and how the preprocessing phase could be improved in our future works. We also expect to investigate the effect of learning with both labeled and unlabeled data in the search phase [9].

Acknowledgment

This work was partially supported by the French National Agency of Research (ANR-06-MDCA-002 AVEIR project).

References

[1] http://inex.is.informatik.uni-duisburg.de/2007/mmtrack.html

[2] Salton, Gerard and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing and Management 24 (5): 513-523.

[3] F.Song and W.B.Croft (1999). A General Language Model for Information Retrieval. Research and Development in Information Retrieval: 279-280.

[4] http://ir.dcs.gla.ac.uk/resources/test_collections

[5] Chengxiang Zhai and John Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. Carnegie Mellon University 2004

[6] Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet and Bernhard Scholkopf. Ranking on Data Manifolds. NIPS 2003

[7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 23.2: The algorithms of Kruskal and Prim, pp.567-574.

[8] Sabrina Tollari, Marcin Detyniecki, Ali Fakeri-Tabrizi, Massih-Reza Amini, and Patrick Gallinari. UPMC/LIP6 at ImageCLEFphoto 2008: on the exploitation of visual concepts (VCDT). In *Working Notes of ImageCLEFphoto2008*, 2008.

[9] Jean-Noël Vittaut, Massih-Reza Amini and Patrick Gallinari. Learning Classification with Both Labeled and Unlabeled Data. Proceedings of the 13th European Conference on Machine Learning (ECML'02), pp. 468-476, 2002.