FINT: Find Images aNd Text

Menno van Zaanen
ILK,
Tilburg University, the Netherlands
mvzaanen@uvt.nl

Guido de Croon
Computer Science,
Universiteit Maastricht, the Netherlands
G.deCroon@CS.unimaas.nl

Abstract

In this article, we describe the FINT system, which stands for Find Images aNd Text. This system is built within the VindIT project, that focuses on handling large amounts of multi-media data. The current approach concentrates on searching in a combination of textual and visual data. The system described here is an iterative system that computes distances between the images. From each image (and corresponding case), a feature vector is extracted. The distances are now computed using these feature vectors. The distance computation can be different in each step of the iterative system. Here we will describe the system and settings that were used in the medical retrieval task of the ImageCLEF 2004 competition.

1 Introduction

The research described in this article is done in the context of the VindIT project[1], which is part of the ToKeN2000 research programme[2].

The goal of the ToKeN2000 research programme is "to focus on fundamental problems of interaction between a human user and a knowledge and information system". The research programme contains many different projects, of which VindIT is one.

The VindIT project concentrates on handling large collections of multi-media data. This includes clustering, indexing, retrieving, and navigating mainly textual and visual information. The project is a co-operation between researchers of the universities of Maastricht, Nijmegen and Tilburg, all in the Netherlands.

The ImageCLEF competition was taken to be a first test case of the implemented system. Entering the competition allowed us to test the system, even though the actual setup of the problem does not completely match the original idea behind FINT, it showed the flexibility of the system and indicated the current problems of the system and also what specific directions should be taken as future work.

In the rest of the article, we will give a brief description of the task of the ImageCLEF competition including the information that has been used in the FINT system. Next, the system will be described in detail. Both the visual and textual features that are incorporated in the system are described. The implementation is discussed next, followed by the conclusions.

[1] See http://www.niwi.knaw.nl/en/oi/nod/onderzoek/OND1297559/toon for more information.
[2] See http://www.ins.cwi.nl/projects/Token2000/index-en.html for more information.

2 Task description

The goal of the medical information retrieval task given in the ImageCLEF competition is to find similar images in a given set of images starting from an image that is not in the given set. The underlying idea here is that a doctor who has, for example, an image of an x-ray, can find similar images of known cases.

The dataset is taken from the CasImage medical database and is developed by the University Hospitals Geneva. It consists of images and corresponding (textual) case information. All images are linked to a case, but some cases contain no real information. A case may be linked to several images, but this is not necessarily the case.

The 8,725 images contained in the database are mainly x-rays, scans and some photos. All images are encoded using the JPG format. The size of the images is not always the same, which introduces some problems as will be discussed below.

The database consists of 2,078 XML encoded cases. A case has several entries containing plain text. Not all fields contain information (and some cases are completely empty apart from a case number). We store all information of the cases in our own database, but we only use the following information per case:

File This field contains the filename of the case;

Description This field contains general information on the case;

Diagnosis Here, the diagnosis of the case is given;

ClinicalPresentation More information on the case is given in this field. It may be more general information on the case or on the patient;

Commentary In this field, general comments can be given;

Chapter This indicates a certain subset in the database. Related cases are stored in the same chapter;

The information contained in other fields in the database might provide additional information, but since they are often empty, we decided not to incorporate them in the current system.

Note that there is also a "Language" field in the XML entries, but this is often empty or incorrect. We will discuss this further in section 3.2.1.

3 System Overview

Within the VindIT project, we have developed a relatively generic multi-modal IR system. It is completely feature-based, which allows for the integration of all types of data as long as features of the data can be extracted.

The advantages of using features are manifold. Many types of multi-modal data can be represented in a simple way, the system remains relatively simple and fast, and feature vectors can be applied to machine learning techniques.[3]

The flexibility of the system is actually used in this task, because the system has originally been developed to take textual (or a combination of textual and visual) information as input. Effectively, this is a similar task considering that all this information is encoded in (numeric) features. Note that the system can handle both numeric and symbolic features, however, only numeric features are used in this specific task.

Figure 1 gives an overview of the FINT system. The upper row illustrates the initial step. First of all, the search information (in this case a search image) is handed to the feature extractor. This outputs a feature vector representing the original data.

[3]In this particular case, no annotated data was provided, so supervised machine learning techniques could not be used. We expected that unsupervised techniques would not provide adequate results.

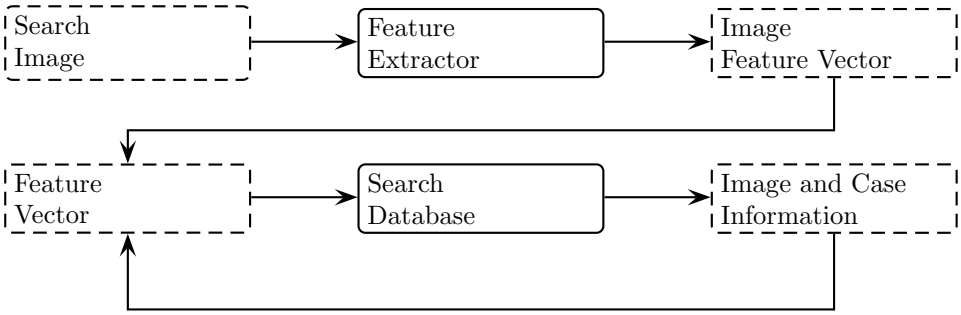

Figure 1: Overview of the FINT system

The lower row shows the iterative phase of the system. It uses a database containing the feature vectors from the images and corresponding cases in the database provided for the competition. These feature vectors are generated similarly to the way the feature vector is generated in the second step in the upper row.

The iterative phase starts with a feature vector (in the first iteration, this is the feature vector from the original data). This feature vector is compared to all feature vectors in the database. The best feature vector (one or more) from the database are returned. This contains information on the best matching image and possibly case information with respect to the search feature vector. These feature vectors can again be used to search the database.

The iterative nature of the system allows for the use of different features. In the first iteration, only visual features are present in the feature vector, because the system was started with a search image and the search image is not present in the database, so no corresponding case information can be found. At the end of the first iteration, visual and textual features can be found, because only feature vectors from the database are returned. These contain visual and textual information.

Selecting the best feature vectors is done by computing the distance between the search feature vector and the feature vectors in the database. The feature vectors are then sorted on distance and the ones with the shortest distance are returned. Note that feature weighting can also be used to give certain features a bigger influence in the distance.

Next, we will describe the features have been implemented in the system. We will start with a discussion of the visual features, followed by the textual features.

3.1 Visual Features

The medical database offered by the University Hospitals of Geneva contains X-rays, scans, and normal pictures. Therefore, the contents of the image-database are rather specific. We have based our image retrieval techniques on the specific properties of the database. We use three types of features for the image retrieval part of the system: *color features*, *principal components of the images*, and *intensity grid features*. We discuss these three types of features in the following subsections.

3.1.1 Color

There are two reasons why color is rather irrelevant for the medical retrieval task. First, the amount of color images in the medical database is almost negligible. Most of the images in the database are gray-value images. In fact, most images represent X-rays or black-and-white scans. Second, the medical image retrieval task demands some color-insensitivity. If the query to the image database consists of a color photo of a leg, we do not want to exclude black-and-white photos from the result set.

Because of the relatively low importance of color in the medical database, the simplest of color features adequately captures the necessary color information. We use three features to code the

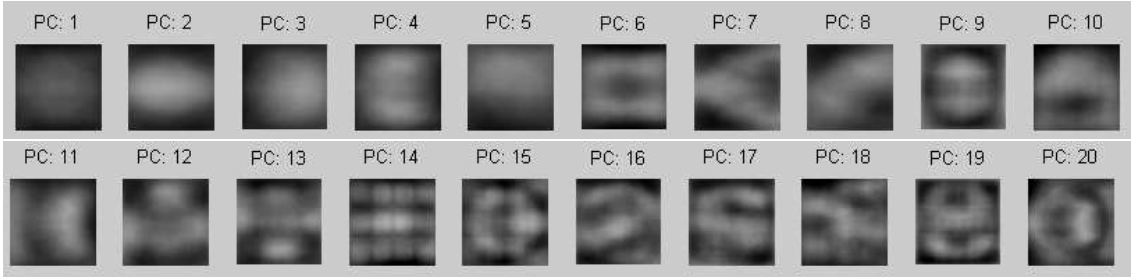

Figure 2: Principle components

color information: the average red, green, and blue values of all pixels in an image. The average values are divided by 255, mapping them to the interval [0, 1]. As stated before, color plays only a complementary role in our image retrieval technique.

3.1.2 Principal components

The shape of the 'object' in the image is much more important to image retrieval in the medical database. How can we measure the shape of an object? The gray-values of the image contain all shape-information, but it would be too cumbersome to use all gray-values as features for retrieving images from the database. Principal component analysis (PCA) is a technique that reduces data dimensionality, while retaining as much information as possible. PCA searches for orthogonal eigenvectors that capture as much variance of the data as possible. PCA is often used in image analysis, for example in facial expression recognition [CBM+01, DCPA02]. Hereafter we explain how we applied PCA to the medical database.

PCA can only be applied to images of the same size. Therefore, the first step in PCA is to resize all images from the database to the same size, in our case to a 40 x 40 pixel format. Naturally, this results in some information-loss. In particular, the ratio of width and height of an object is neglected. After resizing, an image can be represented by a vector of 1600 gray-values. We constructed the data-matrix for PCA by combining such vectors of all images in the database. With the help of the data matrix, 20 principal components were obtained. Since the principal components are also vectors of size 1600, we can visualize them to illustrate the shape-information that they capture. Figure 2 shows the first 20 principal components. The principal components capture some 'elementary' shapes occurring in the medical database. A clear example is the 14th principal component that seems to represent pictures of multiple X-rays on the same sheet.

After PCA, every picture in the medical database can be represented by its projection on the principal components shown in figure 2. We normalize the resulting 20 feature values so that they are in [-1, 1]. The calculation of the projection on the 20 principal components comes down to a multiplication of the image vector with the matrix containing all principal components. Hence, projecting a query image on these components is computationally cheap.

3.1.3 Intensity grid

The final type of visual features that we extract from the images also captures shape information. The shape of an object is partly determined by the overall intensity-distribution in the images. We measure the intensity-distribution by placing a grid over each image in the database and determining the average intensity per grid cell. This average value is divided by 255, so that the values will be in [0, 1]. In the implementation of the FINT-system we have chosen for a grid of 5 x 5, as a trade-off between the number of features and the results that the method yields. In consequence, the number of features per image is 25. This is comparable to the 20 features resulting from the principal component analysis. The intensity grid is important, because

it complements the principal-component approach to shape representation. Both type of features lead to different retrieved images.

3.2 Textual Features

The textual information, contained in the cases, created some problems that had to be solved beforehand. The texts contained a lot of errors. First of all, the original text was not proper UNICODE, so accented characters needed to be converted into their proper codes. Fortunately, a straightforwards mapping to UNICODE could be found.

Once the proper UNICODE encoding of the text was created, we tried to do more complex language handling. However, we noticed that the text contains many spelling errors, non-accented characters that should have been accented, unexpected punctuation marks, incorrect or incomplete abbreviations, ungrammatical and incomplete sentences. This made the linguistic tools we have available (such as stemmers, taggers, chunkers, etc.) almost unusable.

Additionally, the multi-lingual aspect of the competition, discussed in the next section, made the task even more difficult. Whereas the focus of the VindIT system is mainly to search in multi-modal information, multi-lingual information can be incorporated, but it is not an important aspect of our current research.

3.2.1 Languages

A case may contain English or French text. The "Language" field in the case should indicate what language is used in that particular case. Unfortunately, we found that some cases even contained both English and French fields. Additionally, deciding the language of a case, is quite difficult, because the "Language" field of a case is often incorrect or empty.

We have tried to figure out in what language a field is by handing it to van Noord's implementation[4] of the TextCat Language Guesser [CT94]. Unfortunately, this did not work well, since most fields do not contain enough text to decide on which language it is. Also, the words are mainly medical terms, which look similar in English and French. The language models used by the guesser are build on "standard" English and French. However, even with specially built language models, the language guesser cannot be certain in which language certain fields are.[5]

Since the focus of the project is not really on solving multi-lingual retrieval, we have effectively given up on performing complex linguistic feature extraction methods. We could not easily find in which language a piece of text was. Also, the fact that (especially the French texts) contained a lot of errors, which made an extremely simple word-for-word translation of the texts difficult. Not to forget that the actual text consists of mainly highly specific medical terms, for which we could not find an electronic dictionary. We decided on taking a generic approach to try and incorporate English and French texts together in one cluster of features.

3.2.2 Infomap

The text contained in the cases need to be encoded in the form of feature values. Of course, there are many different ways this can be accomplished. The actual FINT system can incorporate features (numeric and symbolic), so the decisions made here are not restricted by the FINT system. Here we chose to use relatively simple features, because the focus of the VindIT project is not directed towards multi-lingual information retrieval. We expect that selecting better textual features will improve the results of the system.

We have extracted plain text from the "Description", "Diagnosis", "ClinicalPresentation", "Commentary", and "Chapter" fields. These fields are often filled with a varying amount of text. Next, we removed the most obvious errors from the text. This included removing all punctuation, correcting some abbreviations, expanding all truncated words (such as converting "l'" to "le" in

[4]This implementation can be found at `http://odur.let.rug.nl/~vannoord/TextCat/`.

[5]We have also tried to annotate language information semi-automatically, but often even humans could not decide in what language certain cases were.

French and "doesn't" to "does not" in English). Also, dates, ranges, percentages, numbers, units and words containing numbers are grouped together in their respective class (e.g., denoted by "[DATE]"). We argue that, for example, specific numbers are not very important, but the fact that there is a number present is indeed important.

The cleaned-up plain text excerpts are used as input of the *infomap* system.[6] This system is developed by Schütze [Sch97] and uses frequency of co-occurring words in the context. When words are often used in the same context, this indicates that they share a similar meaning. Clustering words together gives some sort of semantic clusters. This is generalized between the texts per case, showing how similar cases are conceptually.

Infomap has been applied in several systems. Interesting applications (and related to this research) is the use of infomap in multi-lingual information retrieval systems [MFKP99]. Multi-lingual, aligned corpora are used to find semantically similar clusters, that can be used to handle the texts or queries in the different languages.

Unfortunately, we do not have aligned corpora here, so we simply treat all the data as similar. In effect this will probably result in a strong preference for texts that are in the same language as the query. Of course, this is not preferable, but at least texts within languages are grouped according to semantic content.

Applying the infomap system to the texts extracted from the cases, resulted in 33 numeric features (ranging between -1 and 1).

4 Implementation

The implementation of the FINT system is currently divided over several components, that run on different computers (although that is not necessary). The user interface is implemented using PHP to work over the web. This has several advantages. Firstly, it allows for easy access for the members of the project, who are working in different locations, using different operating systems. Secondly, it is easy to display the graphical content of the database. Thirdly, specific system settings and selections can be made using forms that can be linked to underlying software. Output can again easily be feed back to the user.

Once the user has made a selection of the test image, the distance function, the features that need to be used combined with their weights, the FINT program is started. This program extracts the correct feature vector of the test image and computes the distances of all the similar feature vectors in the database. The images of the best feature vectors are returned to the user. The case information attached to the images can be reached by clicking on the images. This allows for an easy way to get all the information related to an image.

Next, the user can continue with the new images and perform a next iteration of the system. Again, the settings can be adjusted. In the final iteration, the user can specify that TREC output is needed. This will generate a web-page with the TREC output of the current image ordering with their distances.

The database is implemented in MySQL [Wid02]. It is extremely flexible in that the features themselves are encoded in the database as well. This means that using information taken from the database, select statements are created dynamically. This allows the entire system to be reused with a different dataset without any reimplementation. All parts that need to be changed can be found in the database itself.

The interface between the web interface and the database is a program that computes the distances between feature vectors and returns this information to the user. Effectively, the PHP page starts this program with the settings given by the user, the program connects to the database to retrieve the correct feature vectors and computes distances between them. These are then ordered and the images belonging to the best feature vectors are put in a new PHP page that is presented to the user again.

[6] The implementation and documentation of the infomap system can be found at `http://infomap.stanford.edu/`.

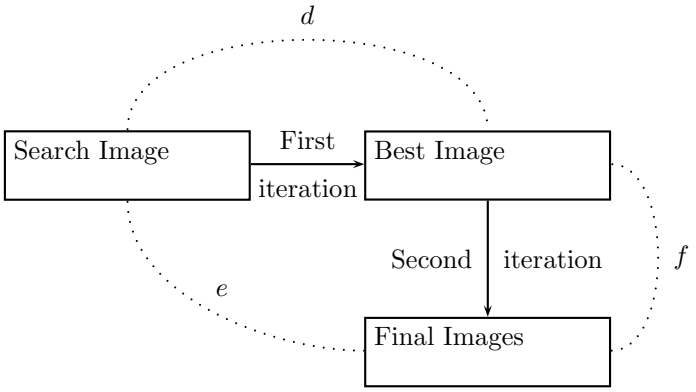

Figure 3: Distance computation in multiple iterations

The computation of the current results is done in two iterations. The first iteration is based on all visual features, with weight 10 for the red, green, and blue features, and 1 for the other visual features. From this iteration we only select the best image. This is the image from the database that looks most similar to the original search image. The second iteration uses only the image from the first iteration and including the textual infomap features (all 33) with weight 30 combined with the visual features (with the same weights) the distances from all images in the database are computed. These results have been submitted to the competition. Several distance functions have been implemented. We have used a weighted numeric Euclidean distance here. This is computed between two vectors $V_1 = (i_1, i_2, \ldots, i_n)$ and $V_2 = (j_1, j_2, \ldots, j_n)$ and weight vector $W = (w_1, w_2, \ldots, w_n)$ as follows:

$$d(V_1, V_2, W) = \sqrt{\sum_{l=1}^{n} (w_l * i_l - w_l * j_l)^2}$$

The main problem with this current approach is that the distance computation is in fact broken. The problem is illustrated in figure 3. The first iteration finds the image that is most similar to the original search image. There is of course a distance between these feature vectors, in the image, this distance is d. In the next iteration, this image is taken as the seed to find similar images. This means that the distances of the final images after two iterations are computed with respect to the best image of the first iteration. Of course, the result image of the first iteration is in the set of final images (because the distance is 0).[7] Since the distances of the other images of the final result are computed with respect to the image of the first iteration, this can be seen as e, whereas to correctly compare the distances of all the final images, distance f should have been computed. However, it is only possible to compute f with respect to visual features, because the search image does not have any case information associated with it.

5 Conclusion and Future Work

This competition allowed us to apply the FINT system to real data for the first time. It showed that the system is flexible and usable with different datasets. Multiple iterations allowed for different visual and textual features to be used, even when these features cannot be found in the initial search data.

The application of the system also revealed problems and shortcomings of the system. The main problem is the incorrect distance calculations (as described above). This will need to be solved in future versions of the system. Additionally, certain implementation problems had to be solved. The speed of the current system could be improved by moving functionality to different parts of the system (such as moving the distance computation to the database itself).

In the future, we would also like to evaluate many different settings. In addition to varying weights, multiple iterations can be used with different feature combinations. Also, the amount of

[7] To handle distances over several iterations, we add the distances of the separate iterations. This means that the distance of the image of the first iteration still has distance d in the final result.

images that are retained after each iteration can be modified. Adjusting these parameters may result in a wide range of results.

The main problem we encountered when generating the results was that evaluation of the results is (nearly) impossible without annotated data. No (annotated) training data was given, which meant that no machine learning approaches could be incorporated (as was originally intended in the FINT system). When annotated data becomes available, more and more interesting approaches can be evaluated.

References

[CBM+01] A.J. Calder, A.M. Burton, P. Miller, A.W. Young, and S. Akamatsu. A principal component analysis of facial expressions. *Vision Research*, 41:1179–1208, 2001.

[CT94] W.B. Cavnar and J.M. Trenkle. N-gram-based text categorization. In *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval; Las Vegas:NV, USA*, pages 161–175. UNLV Publications/Reprographics, April 11–13 1994.

[DCPA02] M.N. Dailey, G.W. Cottrell, C. Pradgett, and R. Adolphs. EMPATH: A neural network that categorizes facial expressions. *Journal of Cognitive Neuroscience*, 2002.

[MFKP99] Hiroshi Masuichi, Raymond Flournoy, Stefan Kaufmann, and Stanley Peters. Query translation method for cross language information retrieval. In *Proceedings of the Workshop on Machine Translation for Cross Language Information Retrieval, MT Summit VII; Singapore*, pages 30–34, September 1999.

[Sch97] Hinrich Schüte. *Ambiguity Resolution in Language Learning*. Number 71 in CSLI Lecture Notes. CSLI Publications, 1997.

[Wid02] Michael "Monty" Widenius. *MySQL Reference Manual*. O'Reilly Community Press, 2002.