Multi-Modal Information Retrieval using FINT

Menno van Zaanen[1] and Guido de Croon[2]

[1] ILK,**
Tilburg University, the Netherlands
mvzaanen@uvt.nl
[2] Department of 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 in the context of the VindIT project, which focuses on handling large amounts of multi-media data. The system described here iteratively searches through a multi-media database by computing distances between the data entries (images and text). From each entry, a feature vector is computed. Distances between database entries are computed using a weighted version of their corresponding feature vector and entries similar to the initial search query are selected based on these distances. Here, we will describe the system and settings that were used in the medical retrieval task of the ImageCLEF 2004 competition.

1 Introduction

This article will describe the Find Images aNd Text (FINT) system. This system was developed within the context of the VindIT project[3], which is part of the ToKeN2000 research programme[4].

The ToKeN2000 research programme focuses "on fundamental problems of interaction between a human user and a knowledge and information system". This research programme contains several projects, of which VindIT is one. The VindIT project concentrates on handling large collections of multi-media data. At the moment, the project focuses on clustering, indexing, retrieving, and navigating of mainly textual and visual information. The project is a cooperation between researchers of the universities of Maastricht, Nijmegen and Tilburg, all in the Netherlands.

The FINT system is the first implementation of a flexible multi-modal system build within VindIT. It has several requirements, the most important being flexibility. Ideally, one should be able to use the system for many tasks, such

** The first author currently works at ICS, Macquarie University, Sydney, Australia.

[3] See http://www.niwi.knaw.nl/en/oi/nod/onderzoek/OND1297559/toon for more information.

[4] See http://www.ins.cwi.nl/projects/Token2000/index-en.html for more information.

as searching the database, clustering and indexing entries, but also including interactive, user-driven tasks.

The ImageCLEF competition is taken to be an initial test case of the system. The main aim is not necessarily to get the best results in the competition, the current system is too simple for that, but to show the flexibility of the approach taken.

The original task for which FINT is developed is a search tool that should help searching large databases of pairs of images and corresponding text. Instances can be found, for example, in museums or other institutions that have pictures and descriptions of items.

The actual setup of the ImageCLEF task does not completely match the original idea behind FINT. However, participating in the competition will show the flexibility of the system. Additionally, it indicates problems of the current implementation and the outcomes may help to specify directions for future work.

In the rest of the article, we first give a brief description of the task of the ImageCLEF competition mainly focusing on how the task is different from the intended task of the FINT system (section 2). Section 3 describes the system in detail. Next, both the visual and textual features that are incorporated in the current system are described respectively in sections 3.1 and 3.2. The actual implementation is discussed in section 4 and is followed by the conclusions.

2 Task Description

The goal of the medical information retrieval task in the ImageCLEF competition is to find similar images in a given set of images starting from a search image. The underlying idea here is that a doctor who has, for example, an image of an X-ray, can find similar images belonging to known cases. Additionally, information from the case corresponding to the found images can give clues on how to treat the patient further.

As described in [1] in more detail, the dataset used for the competition is taken from the CasImage medical database and is developed by the University Hospitals Geneva. The dataset consists of medical cases. A case contains textual case information and is linked to one or more images. All images belong to a case and a case may have several images linked to it.

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.

There are several aspects of the competition that do not completely match with the original task set for the FINT system. Because of the flexibility of the system, it can be applied to the competition tasks, although some adjustments need to be made.

- The cases contain textual information in two languages, English and French. This aspect will be discussed in more detail in section 3.2.
- There is a many to one relationship between the images and the text. The fact that certain images are related because they belong to the same case may represent important information. However, at the moment this information is not used.
- The search query is an image only. Of course, this is not a problem, but it means that textual information can only be used when at least a two stage search is used. The first stage searches for similar images. These images have textual information attached to them, so the second stage can use this information as well.

3 System Overview

The FINT system is a generic multi-modal system. Here it is used for information retrieval, but it could be used for other tasks as well. 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. If a types of multi-modal data can be represented using features, it can be incorporated in the system. In practice, this is true for many types of data. A system based on features, therefore, remains relatively simple and flexible. Additionally, feature vectors can be applied to machine learning techniques.[5]

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.

[5] 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. In section 5, we describe some parameter tuning, but this was only done after the competition.

The lower row shows the iterative phase of the system. It matches the search feature vector against 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 feature vector in the second step in the upper row.

The iterative phase, in general, starts with one or more feature vectors. The input feature vectors are compared to the feature vectors in the database and distances are computed. The feature vectors that have the smallest distances to the input feature vectors are returned and can be used as input feature vectors in another iteration.

In this particular competition, there is always only one input feature vector. The first iteration describes the search image, so the feature vector only contains visual features. The output of this iteration gives the image (according to the features), that best matches the input image. Since this image is in the database, textual features can now also be used, so the second iteration uses, next to visual features, textual information as well.

The final output of the system is a list of the best 1,000 images that correspond to the feature vectors that are the output of the second phase. To summarise, the first iteration searches for the image in the database that closely resembles the original search image. The second iteration uses visual and textual features of the output of the first iteration and this feature vector is used to generate the output of the system that is checked according to the trec evaluation method.

Of course, the performance of the system depends heavily on the features used. Additionally, feature weighting is implemented, which allows for certain features to have more influence in the distance computation.

Next, we will describe the features that 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 content of the image-database is rather specific. Our image retrieval techniques are based 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*. All features are relatively simple, but are rather effective in this particular context. We discuss the three types of features in the following subsections.

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

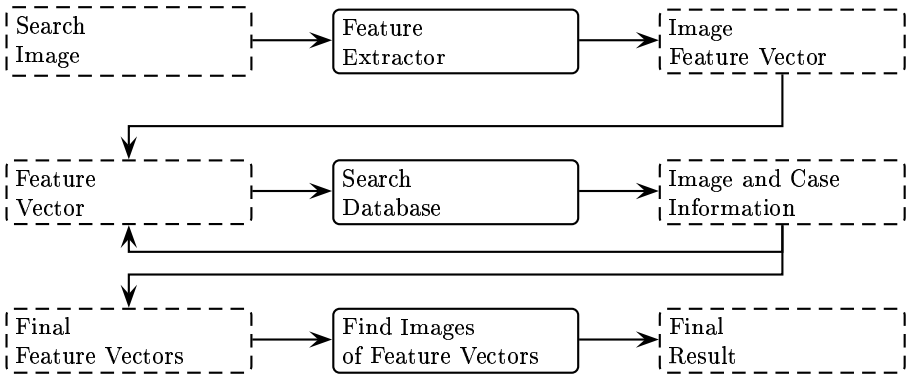

Fig. 1. Overview of the FINT system

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 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.

Principal components The shape of the "object" in the image is much more important to image retrieval in the medical database than color information. 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 [2, 3].

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 40×40 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.

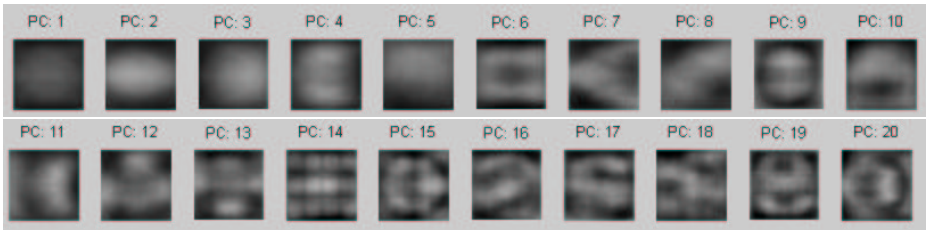

Fig. 2. Principle components

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 5×5, 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 types of features lead to different retrieved images.

3.2 Textual Features

Images are linked to cases that contain text describing the patient, diseases and treatments. To be able to treat the text in a similar way to the visual information, features need to be extracted from the texts. These features should represent the "important" aspects of the text as close as possible. However, selecting features that do this is not easy.

Before features can be extracted, the text should be as "clean" as possible. Unfortunately, the content of the cases showed some aspects that have to be

addressed before. First of all, the original text is not proper UNICODE, so accented characters need to be converted into their proper codes. Fortunately, a straightforward mapping to UNICODE can 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, that will be discussed in the next section, makes 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.

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, some cases even contain fields of both English and French. Additionally, deciding the language of a case, is quite difficult, because the "Language" field of a case is often incorrect or empty.

To figure out what language a field in a case is in, we have tried to run it through van Noord's implementation[6] of the TextCat Language Guesser [4]. Unfortunately, this does 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.[7]

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. Firstly, we cannot easily find language of a piece of text. Secondly, the fact that (especially the French texts) contain a large number of errors, which make even an extremely simple word-for-word translation of the texts difficult. Thirdly, the actual text consists of mainly highly specific medical terms, for which we cannot find a good electronic dictionary. Based on these findings, we decided on taking a generic approach to try and incorporate English and French texts together in one cluster of features.

Infomap The text contained in the cases needs to be encoded in the form of feature values. Of course, there are many different ways in which this can be accomplished. The FINT system can incorporate features (numeric and symbolic), so the actual decisions made here are not restricted by the FINT system.

[6] Implementation can be found at `http://odur.let.rug.nl/~vannoord/TextCat/`.

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

Here we describe 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 extract plain text from the "Description", "Diagnosis", "ClinicalPresentation", "Commentary", and "Chapter" fields. These fields are often filled with a varying amount of text. Next, we remove 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 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.[8] This system is developed by Schütze [5] 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 [6]. 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 bi-lingual, 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, results in 33 numeric features ranging [-1, 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 fed back to the user.

The FINT program starts after the user has made a selection of the test image, the distance function, the features, and the weights assigned to the fea-

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

tures. This program extracts the correct feature vector from 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 textual 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 [7]. 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 re-implementation. 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.

The computation of the results documented in the competition 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.[9] From the results of this iteration, we only select the best image. This corresponds to the image from the database that looks most similar to the original search image.

The second iteration uses the textual infomap features with weight 30 in addition to the visual features (with the same weights). Using these settings, the distances from all images in the database are computed. These results were 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:

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

There is an interesting problem with the distance computation. Even though the distance function works, using it to compute distances over multiple iterations does not work as expected. The problem is illustrated in figure 3. The first

[9] In the first iteration, textual features cannot be used, because the input image does not have textual information attached to it.

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 called d. In the next iteration, the feature vector of 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).[10] 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.

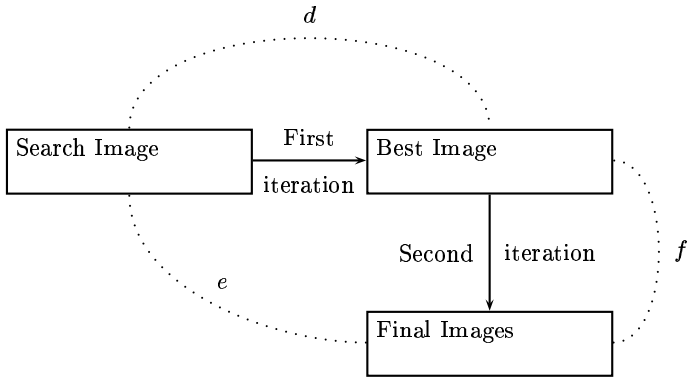

Fig. 3. Distance computation in multiple iterations

5 Parameter Tuning

The FINT system does not perform spectacularly well in the competition, but this could be expected. It is an extremely simple system that does not use many features to get a good description of the data. However, we are especially interested in the influence of the second iteration with respect to the textual information.

To investigate the influence of the second iteration and hence the usefulness of the textual information (in its current form), we performed some parameter tuning based on the annotated data that was made available after the completion of the competition.

[10] As a temporary fix, 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.

Table 1[11] gives an overview of the results. The best parameters are given together with the results obtained with these settings. Note that the distance computation in these results is slightly different. These results are denoted by "New". We used the absolute distances here, in contrast to the distances given in the competition, where the maximum distance minus the absolute distance is used.[12] Additionally, in the results of the second iteration, the distance of the image that is used as the seed is 0. The results with the same parameter setting according to the original system are denoted by "Original". Finally, the original results (with non-tuned parameters) of the ImageCLEF competition are given (denoted by ImageCLEF). These results were generated with the following weights: red, green, blue have 10, principle components and mean intensity values have values 1 in the first iteration, and the same values in the second iteration, but additionally, the infomap features are added with weight 30.

Table 1. Best parameters for first and second iteration

Distance		Visual					Textual	
Computation	Iteration	Red	Green	Blue	PC	MIV	Infomap	Result
New	1	12	12	12	4	2	n/a	0.2508
Original	1	12	12	12	4	2	n/a	0.2406
New	2	4	4	4	1	1	2	0.2752
Original	2	4	4	4	1	1	2	0.2752
ImageCLEF	2	See text						0.1519

From the results it is clear that adding textual information improved the results of the system. We have tried many parameter settings, but the one including the textual infomap features performs best.

6 Conclusion and Future Work

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

The main results of the system showed that with the ImageCLEF competition data, including textual information improved the results over the same system with visual features only. We expect that the results of FINT can be further improved by incorporating more (and perhaps more informative) features.

[11] PC denotes the principle components features and MIV denotes the mean intensity values features.

[12] The competition required the distances to be descending instead of ascending.

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 incorporate machine learning algorithms that automatically learn the best parameter settings. Of course, in order to do this, one needs training data (which was not available in the competition). Parameter tuning can of course vary weights for each feature (and each iteration), but it may also tune the number of iterations, the distance metric, the amount of images that are retained after each iteration (which may be combined using several clustering techniques), etc. Adjusting these parameters may result in a wide range of results.

References

1. Clough, P., Müller, H., Sanderson, M.: The clef cross language image retrieval track (imageclef) 2004. In Peters, C., Clough, P., Gonzalo, J., Jones, G., Kluck, M., Magnini, B., eds.: Fifth Workshop of the Cross–Language Evaluation Forum (CLEF 2004). Lecture Notes in Computer Science, Berlin Heidelberg, Germany, Springer-Verlag (2005)
2. Calder, A.J., Burton, A.M., Miller, P., Young, A.W., Akamatsu, S.: A principal component analysis of facial expressions. Vision Research **41** (2001) 1179–1208
3. Dailey, M.N., Cottrell, G.W., Pradgett, C., Adolphs, R.: EMPATH: A neural network that categorizes facial expressions. Journal of Cognitive Neuroscience **14** (2002) 1158–1173
4. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval; Las Vegas:NV, USA, UNLV Publications/Reprographics (1994) 161–175
5. Schütze, H.: Ambiguity Resolution in Language Learning. Number 71 in Lecture Notes. Center for Study of Language and Information (CSLI) Publications, Stanford:CA, USA (1997)
6. Masuichi, H., Flournoy, R., Kaufmann, S., Peters, S.: 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. (1999) 30–34
7. Widenius, M.M.: MySQL Reference Manual. O'Reilly, Sebastopol:CA, USA (2002)