IBM T.J. Watson Research Center, Multimedia Analytics: Modality Classification and Case-Based Retrieval tasks of ImageCLEF2012

Liangliang Cao, Yuan-Chi Chang, Noel Codella, Michele Merler,
Quoc-Bao Nguyen, and John R. Smith

19 Skyline Dr.
Hawthorne NY 10532
{liangliang.cao,yuanchi,nccodell,mimerler,quocbao,jsmith}@us.ibm.com
http://www.watson.ibm.com/

Abstract. In this paper we present the modeling strategies that were applied by the IBM T.J. Watson research team to the modality classification and case-based retrieval tasks of ImageCLEF 2012.
The primary challenges of this year's medical modality classification task were as follows: 1) the supplied training data was extremely limited, with some categories having as few as 5 positive examples, leaving little room for internal testing, and 2) some modalities appeared to be visually similar.
In order to address these challenges, we approached the task from two fronts: 1) we attempted to augment the training data with additional examples of each category, and 2) we experimented with a broad range of modeling strategies and feature extraction techniques.
For the case based retrieval task, we employed a semantic similarity approach to measure the relatedness among medical concepts found in the text corpus. We believe the lack of using additional lexical database besides the UMLS-methathesaurus led to poor performance in relation to other approaches.

Keywords: SVM, Multiclass, Kernel Approximation

1 Introduction

The ImageCLEF 2012 Medical Modality Classification Task is a standardized benchmark for systems to automatically classify medical image modality from PubMed journal articles. The 2012 dataset has changed from the previous year in 3 significant ways: 1) there are more categories, 2) the number of training examples is far fewer, and 3) some modalities are more similar.

Our approach can be described as scaling up to utilize as many features and data as possible. Our experiments demonstrate that increasing either axes tends to boost performance. In addition, we present a method for kernel approximation to help address the computational time costs of using a wide variety of methods.

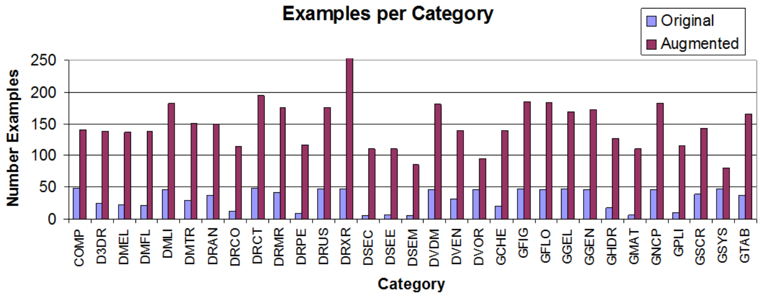

Fig. 1. Number of visual examples per category in each dataset used for experiments.

For data augmentation, we drew from several sources outside the ImageCLEF2012 collection, such as a Bing web-crawl for each category, as well as publicly available medical image datasets, such as The Cancer Imaging Archives (TCIA), and Image Retrieval in Medical Applications (IRMA). For our modeling approaches, we selected multiple features extracted from a set of image granularities, such as SIFT variants [1], GIST [2], Local Binary Patterns (LBP) [3], edge and color histograms, and Curvelets [4]. In addition, we experimented with a variety of feature fusion and learning approaches, including early, late, and kernel fusion, kernel approximation, multiclass SVM, and one-vs-all. We discovered that multiclass SVM using an early fusion of many features with an augmented dataset yielded the best performance. Kernel approximation methods were able to significantly increase the efficiency of modeling at a small cost to performance.

2 Modality Classification Task

2.1 Datasets

Our experiments are performed on the following datasets:

- **Dataset 1**: The original ImageCLEF2012 training dataset.
- **Dataset 2**: A dataset augmented with up to 100 additional examples per category. Augmenting data was collected from Bing Image Search, a Cornell University Vision and Image Analysis Group & International Early Lung Cancer Action Program (VIA/I-ELCAP) Public CT datase[1], The Cancer Imaging Archives (TCIA)[2], Image Retrieval in Medical Applications (IRMA)[3], and the Japanese Society of Radological Technology (JSRT) [5].

[1] http://www.via.cornell.edu/lungdb.html

[2] Image data used in this research were obtained from The Cancer Imaging Archive (http://cancerimagingarchive.net/) sponsored by the Cancer Imaging Program, DCTD/NCI/NIH.

[3] courtesy of TM Deserno, Dept. of Medical Informatics, RWTH Aachen, Germany

Due to the low number of positive examples in some categories in the original training dataset, we chose to construct an additional augmented dataset. The number of examples per category for both is shown in Fig. 1.

2.2 Feature Collections

All our experiments are run using 1 of 7 sets of low-level visual features. Here we name and describe the contents of 7 sets of features that were used for modeling. Each feature is described by name, with the granularity of extraction in parenthesis. The granularities are as follows:

- **Global**: Feature extracted from entire image. Native feature dimensionality preserved.
- **Grid**: 5x5 image grid, with feature vector extracted from each grid block and concatenated. Increases dimensionality by factor of 25.
- **Grid7**: 7x7 image grid, with feature vector extracted from each grid block and concatenated. Increases dimensionality by factor of 49.
- **Layout**: 5 image regions including the center and the 4 quarters. Increases dimensionaltiy by a facor of 5.
- **Pyramid**: Spatial pyramid, with global as first level, and 2x2 image grid as second level. Increases dimensionaltiy by a facor of 5.

The feature sets referenced later in the section are as follows:

- **Feature Set 1** Color Correlogram (grid), Edge Histogram (grid), Image Type (grid), LBP histogram (grid7), Color SIFT AM Codebook Size 2000 (pyramid), SIFT AM Codebook Size 2000 (pyramid), HSV SIFT AM Codebook Size 1000 (pyramid), Image Stats (grid), Gist (layout), Curvelet Texture (layout).
- **Feature Set 2** Feature Set 1, removing SIFT features, and adding the following: Color Histogram (grid), Color Moments (grid), Dominant Colors (global), Thumbnail Vector (global), Color SIFT AM Codebook Size 1000 (pyramid), SIFT AM Codebook Size 1000 (pyramid), FourierOrientationVector (grid). FourierOrientationVector is a feature representing the average of diameters in Fourier-Mellin space, across varying angles from 0 to 180 degrees.
- **Feature Set 3**: Feature Set 2, adding FourierPolarPyramid (layout). FourierPolarPyramid is a pyramid constructed in polar coordinates of Fourier-Mellin space. 4 radial levels are employed (partitions of size 1, 2, 4, and 8), with 6 angular levels, across 4 color channels (RGB and Grayscale).
- **Feature Set 4**: Color Correlogram (grid), Color Histogram (grid), Edge Histogram (layout), Edge Histogram (grid), Gist (layout), FourierPolarPyramid (global), FourierPolarPyramid (layout), Image Type (grid), LBP Histogram (grid7), SIFT Codebook Size 1000 (global),
- **Feature Set 5**: Feature Set 2, minus all SIFT variants.

- **Feature Set 6**: Image Stats (grid), LBP Histogram (grid), Image Type (grid), Edge Histogram (grid), Shape Moments (grid), Dominant Colors (grid), image stats (global), LBP histogram (global), Image Type (global), Shape Moments (global), Edge Histogram (global), Color Wavelet (global), Dominant Colors (global), Color Moments (global), Color Correlogram (global), Color Moments (global), Gist (global), Wavelet Texture (global), Tamura Texture (grid).
- **Feature Set 7**: SIFT Codebook Size 1000 (pyramid), HSV SIFT AM Codebook Size 1000 (pyramid).

2.3 Multiclass SVM

We employed the LibSVM library [17] to perform Multiclass SVM classification. The process involves learning a set of 1-vs-1 classifiers, one for each pair of classes in the dataset. In order to classify a new example, each 1-vs-1 model is evaluated on it and the most likely label is selected based on a majority voting scheme. No data sampling was performed: the Multiclass SVM model was learned directly from the whole training set (either original and augmented). As such, one of the advantages of this learning strategy consists in explicitly modeling the priors of the classes in the dataset. All the parameters of the models chosen for the final submissions to ImageCLEF 2012 were chosen according to 5-fold (for the original dataset) and 3-fold (for the augmented one) cross validation performances. After experimenting on the internal cross-validation splits, a set of the best performing descriptors was selected for fusion. For the original dataset, Feature Set 1, as described in Section 2.2, was chosen. Feature Sect 2 was instead adopted for the extended dataset. Furthermore, the Chi-square kernel was selected (in preference over linear, RBF and histogram intersection) for all the Multiclass SVM runs, computed as

$$K(\mathbf{x},\mathbf{y}) = 1 - \sum_{d} \frac{(x_d - y_d)^2}{\frac{1}{2}(x_d + y_d)}, \tag{1}$$

Three types of feature fusion methods were experimented: early, late, and kernel fusion.

- **Early fusion**: consists of a concatenation of different descriptors, before SVM modeling
- **Kernel Fusion**: consists of a point-wise pooling (max or average operator) over the kernel matrices produced by each descriptor. The Multiclass SVM is then learned on top of the aggregate matrix
- **Late Fusion**: consists of a pooling (max, average or product) operator over the predictions of the models learned from individual features for each test image. For this type of fusion we employed the probabilistic output option in each SVM, which converts the 1-vs-1 comparisons into class probabilities. For each test image, each model produced a vector with N probabilities (where N is the number of classes, 31 in our case). After the pooling was applied in a point-wise manner over the prediction vectors of the models,

the class with the maximum aggregate probability was chosen as the final prediction.

Each strategy is exemplified in Figure 2. As reported in Section 2.6, the Multiclass SVM trained from the augmented dataset with early fusion strategy (Experiment 12) provided the best performance. Kernel fusion proved to be equivalent, while late fusion performed worse than the other methods.

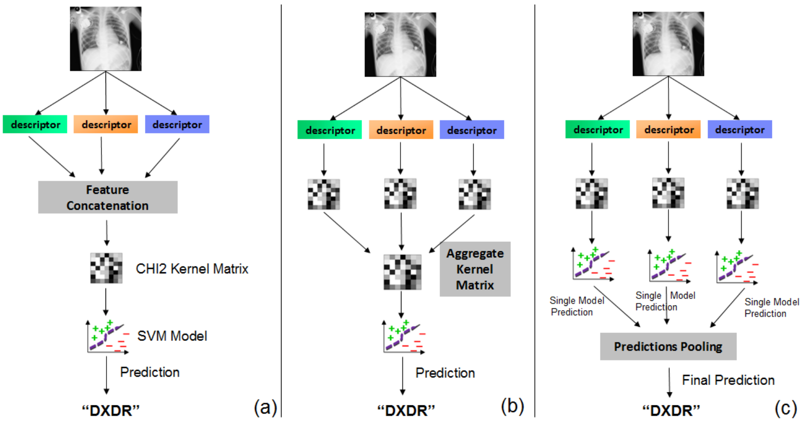

Fig. 2. Multiclass SVM fusion strategies: (a) early fusion, (b) kernel fusion and (c) late fusion. The fusion element in the pipeline is indicated in gray.

2.4 Efficient Feature Fusion with Kernel Approximation

To explore different aspects of visual phenomenon, we employed 19 different features, described in Feature Set 5. We used an early fusion strategy by concatenating these features together and training a kernelized Supporting Vector Machine (SVM). However, a practical problem of using so many features lies in the computational cost, both in the training and testing stage. When the number of images grows, or when the feature dimension increases, traditional SVM solvers may not work well or take a very long time to compute the optimal solution.

Among all the kernels in practice, the Chi-square kernel often yields very good performance compared with the others. Moreover, a large amount of our features, including LBP histogram, edge histogram, color histogram, and SIFT histogram, are in the form of histogram features. Chi-square kernel is arguable regarded as the first choice for histogram form features. In our work, we focus on how to efficiently solve Chi-square kernel only. We do not consider the problem of general kernels.

We consider the Chi-square kernel in the form of

$$K(\mathbf{x},\mathbf{y}) = \sum_d \frac{2x_d y_d}{x_d + y_d}, \tag{2}$$

where $\mathbf{x} = [x_1, x_2, \cdots, x_d, \cdots, \mathbf{y} = [y_1, y_2, \cdots, y_d, \cdots.$

It is easy to see that Eq.(2) is defined as the additive sum of different dimensions. Such a kernel is referred to as an additive kernel. As suggested by [13], such a group of kernels can be approximated by mapping the feature into a high dimensional space. By the representer theorem [14], the solution of classification model can be written in the form of

$$f(\mathbf{x}) = \sum_{i=1}^{N} K(\mathbf{x}, \mathbf{x}_i),$$

where i denotes the index of training samples. For any positive definite kernel, there exists a mapping $x \to \phi(x)$ so that the final classification model becomes

$$f(\mathbf{x}) = \mathbf{w}_\phi^T \phi(x) + b$$

where $\mathbf{w}_\phi$ denotes the weights of the linear model in the mapped space. In this work, we will use Nystrom's approximation to construct the mapping function explicitly.

To make the representation simply we let

$$k(x,y) = \frac{2x_d y_d}{x_d + y_d},$$

then we can see the kernel is

$$K(\mathbf{x},\mathbf{y}) = \sum_d k(x_d, y_d). \tag{3}$$

Next we will discuss how to approximate $k(x,y)$, which is a function on 1D space.

To approximate $k(x,y)$, we employ

$$\phi_j(x) = \begin{cases} \sqrt{\kappa_0} & \text{if } j = 0 \\ \sqrt{2\kappa_{\frac{j+1}{2}} cos(\frac{j+1}{2} Lx)} & \text{if } j > 0 \text{ odd} \\ \sqrt{2\kappa_{\frac{j}{2}} sin(\frac{j}{2} Lx)} & \text{if } j > 0 \text{ even} \end{cases} \tag{4}$$

where ϕ_j and κ_j constitute one pair of Fourier transform, L is the frequency parameter, and we use $L = 2\pi/15$ in practice. Then we can convert the nonlinear kernels with the linear model over $\phi_j(x)$. For more details, please refer to [13].

So far we have discussed how to approximate the kernelized SVM using a linear model. Now we can compare the computational complexity of both methods. Suppose we have m features, $1 \leq m \leq M$, and each feature is of dimension d_m. The number of training samples is N, and size of testing set is T. For each

d_m features, we map it to the space of dimension $7d_m$. Note that the evaluation of kernel SVM depends on the number of support vectors, which in practice is proportional to the number of training examples. Also our linear approximation requires the extra cost of feature mapping, which is $7dN$ for training and $7dT$ for testing. Table 1 compares the computational complexity of the two methods. It is easy to see our linear approximation is much more efficient in both training and testing stage. Our linear approximation is even plausible for the scenario with a lot of features.

Table 1. Comparing the computational cost of kernelized SVM and linear approximation.

	Training	Testing
Kernel SVM	$O(N^2 \sum_m d_m)$	$O(TN \sum_m d_m)$
Linear approx	$O(\alpha N \sum_m d_m) + O(N \sum_m d_m)$	$O(\alpha T \sum_m d_m) + O(T \sum_m d_m)$

To measure the effectiveness of our kernel approximation method, we compare how much difference exists between Chi-square kernels and our approximated kernels. Table 2 illustrates the speed up and percentage of kernel approximation error using randomly-generated features. It is easy to see the approximation error is low, while the speed up will be increasingly significant when the number of training samples grows.

Table 2. Comparing the kernel matrix approximation using our method.

Number of samples	Dim.	Dim. of projection	Speed up (x times)	Approx. error (ratio)
100	100	700	14.83	0.0027
500	100	700	28.19	0.0026
1000	100	700	76.69	0.0026
2000	100	700	118.41	0.0026

It is also interesting to see the classification accuracy after our kernel approximation. We implement the Chi-square kernel with LibSVM, and also use liblinear [16] to train a linear classifier using our projected features. As Figure 3 shows, the linear model based on the high dimensional approximation is not necessarily worse than the original model. In fact in some categories, the liblinear model even works slightly better. Note that we do not tune the parameters

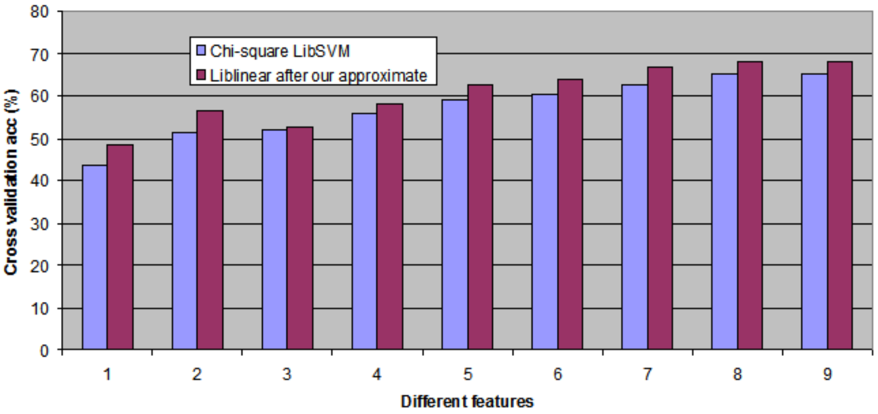

Fig. 3. Comparing the performance using LibSVM and Liblinear with our kernel approximation.

of both models in this toy experiments. In the future, we plan to improve our model with heterogeneous kernel learning methods [15].

2.5 One-vs-All Ensemble SVM with Data Sampling

IBM Multimedia Analytics and Retrieval System (IMARS) [6] has been developed and applied previously for semantic classification of unstructured images and videos. The general framework is depicted in Fig. 4, and is primarily based on generating collections of 1-vs-All SVM classifiers.

Using the IMARS learning paradigm, we explored several variants of the pipeline to better understand the contribution of each to system performance:

- **Feature Fusion:** Early (vector concatenation) or Late (Unit Model score averaging).
- **Data Sampling:** Random subsamples of negative examples, or using the entire dataset.
- **Number of Bags:** Changing the number of Unit Models that are trained for a particular feature by choosing different subsamplings of example data.
- **Feature Sets:** Varying sets of features used for modeling.
- **Kernel Type:** A single RBF kernel with kernel parameter -4, C=100, and sigmoid feature normalization was used for most model learning; however, we performed one experiment with a Chi^2 kernel, similar parameters, to understand any effect the variation of kernels might have.

1-vs-All scores of multiple Ensemble Models were converted to Multiclass labels by choosing the max over all the classifier scores.

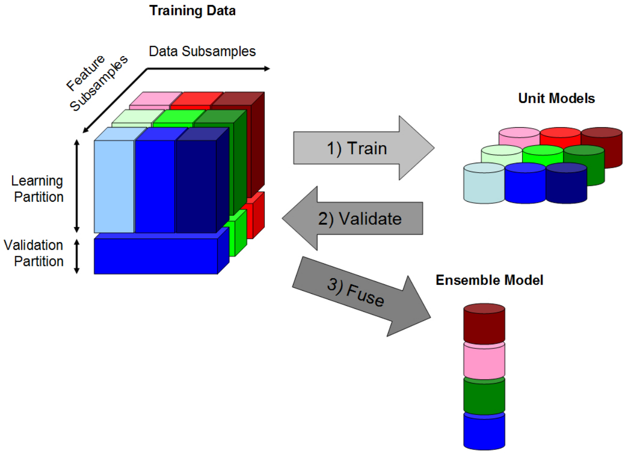

Fig. 4. IBM Multimedia Analytics and Retrieval System (IMARS) Flowchart. Training data is seperated into a Learning and Validation partition, used for model learning and model selection, respectively. Within the Learning partition, data is segmented across features and data subsamples. These segments are referred to as "bags." Within the Validation partition, data is only segmented across features. Unit Models are trained for each bag within the Learning partition using a 1-vs-All SVM. The performance of each Unit Model is assessed on the Validation data partition, and Unit Models are selected for inclusion in the Ensemble Model based on which models boost Ensemble Model performance on the Validation data the most. If no Validation partition is definied (0%), all Unit Models are fused into an Ensemble Model. Fusion is performed by weighted averaging, with weigts based on average precision.

Table 3. Multiclass Accuracy of 1-vs-ALL SVM

Exp.	Fusion	B. Size	# D. Bags	# F. Bags	F. Set	Avg. Dim.	Dataset	VS %	MACC	P/C
1	Early	100	1	1	7	10000	2	0%	51.6%	1.0
2	Early	100	1	1	5	16020	2	0%	50.0%	0.605
3	Early	100	1	1	2	26020	2	0%	55.2%	0.411
4	Early	100	1	1	3	44920	2	0%	55.9%	0.241
5	Late	100	1	15	3	3134	2	20%	42.3%	0.174
6	Early	200	1	1	2	26020	2	0%	59.1%	0.110
7	Early	500	2	1	3	44920	2	20%	62.6%	0.005
8	Early	1000	1	1	3	44920	2	0%	64.2%	0.003
9	Late	1000	1	10	4	3854	1	0%	52.7%	0.003
10	Early	2000	1	1	3	44920	2	0%	66.6%	0.001
11	Early	All	1	1	3	44920	2	0%	68.0%	0.0001

2.6 Results

One-vs-All Ensemble SVM with Data Sampling 1-vs-ALL experiments are depicted in Table 3. Experiment index number, feature fusion type (Early or Late), number of examples for each of positive and negative categories per bag, number of bags across data, number of bags across features, the feature set used, the average dimensionality across feature bags, the proportion of data used for Validation Split (VS), multiclass accuracy scores, and relative performance scaled by computational complexity (P/C) are shown. Computational complexity is computed simply as the running time required to compute a kernel matrix.

Experiment 8 was reproduced using a Chi^2 kernel, yielding MACC of 64.4%, and was not listed in this table.

In the first experiment, we established a baseline with a very small data sampling rate (100 examples in each of positive and negative) and only two SIFT features utilized. In the second experiment, we examined the performance of our other features, excluding SIFT. In the third, we used both sets, and in the fourth we added the newer FourierPolarPyramid feature vector. In the fifth experiment, we examined the effect of using late fusion, instead of early fusion. In the sixth experiment, we began to study the effects of increasing the number of examplars. In the seventh and eigth experiments, we examined the effect of increasing data sampling either by using larger data bag sizes, or a larger number of bags. The ninth experiment represented our submission NCFC_ORIG_2_EXTERNAL_SUBMIT.txt; this dataset used a restricted number of features and late fusion, due to time constraints. In the last two experiments, we finally examine the effects of using larger amounts of data, up to the limit of our dataset.

In summary, our experiments yield a number of notable observations:

1. With early fusion, adding more features improves performance, SIFT contributing the most (see Exp. 1-4).
2. Adding additional data improves performance (Exp. 4, 6-11) .
3. Early fusion of features outperforms late fusion with averaging, and provides a better P/C ratio (Exp. 4-5).
4. Bagging along the data dimension reduces overall performance, but improves P/C (Exp. 7-8).
5. Best performance is acheived using early fusion of all features and all data (Exp. 11).
6. Chi^2 kernels and searching over SVM parameters holds the potential to boost performance further.

Multiclass SVM Multiclass SVM experiments are depicted in Table 4. Three different aspects were explored in the experiments: fusion type, dataset size, and aggregation type. From the results in the Table emerges that:

1. Early and kernel fusion are comparable, and perform better than late fusion
2. a larger training set leads to a better classification performance (Dataset 2 is better than Dataset 1)
3. Averaging seems to be the best aggregation strategy.

Table 4. Multiclass SVM Experiments with Mean Accuracy performance

Exp.	Fusion	Aggregation	Feature Set	Dataset	Mean ACC
12	Early	-	2	2	**69.7%**
13	Early	-	1	1	66.0%
14	Kernel	Avg	2	2	69.7%
15	Kernel	Avg	1	1	66.1%
16	Late	Avg	2	2	68.2%
17	Late	Prod	2	2	67.8%
18	Late	Max	2	2	65.2%
19	Late	Avg	1	1	62.8%
20	Late	Prod	1	1	62.2%
21	Late	Max	1	1	57.9%

2.7 Official Submissions

In Figure 5 are reported the official runs submitted to the ImageCLEF 2012 website, in comparison to all other official submissions (43 in total). IBM achieved the top three visual only classification performances, as well as the best overall accuracy.

The submissions were all purely visual, and corresponded to the following experiments (in decreasing order of performance): 12, 13 with Kernel approximation fusion (as described in Section 2.4), 13, 11, 12 with Kernel approximation fusion, 18, 21. After the submissions we further improved the worst performing ones through better normalization, parameter selection, and further debugging, yielding to the performances reported in Tables 3 and 4.

Fig. 6(b) shows the confusion matrix of the best performing run. Overall the matrix presents a strong diagonal. However, some clear mis-classifications are evident. In particular, “GSYS - System overview” resulted to be the hardest class to categorize, being quite often confused with “GFLO - Flowcharts”. Looking at the appearance of the images in such classes, it is evident that based on visual features alone they are in most cases indistinguishable. This confusion might be mitigated by exploiting the textual information associated with, or in, the images. We plan to follow this direction in future experiments.

We expect that extracting textual information and combining it with our strong visual modeling will boost classification performance, given the complementary information of those two representations, and also looking at the performances of other groups.

In conclusion, appropriately modeling the visual appearance can provide strong modality classification performance, even without text analysis. In our experiments we found that adding more features and training data lead to models with better classification performance. Early fusion and kernel fusion seem to be the best combination strategies. Our kernel approximation provides a principled and efficient framework to perform such fusion, while significantly increasing the efficiency of the computation of the best performing CHI2 kernel.

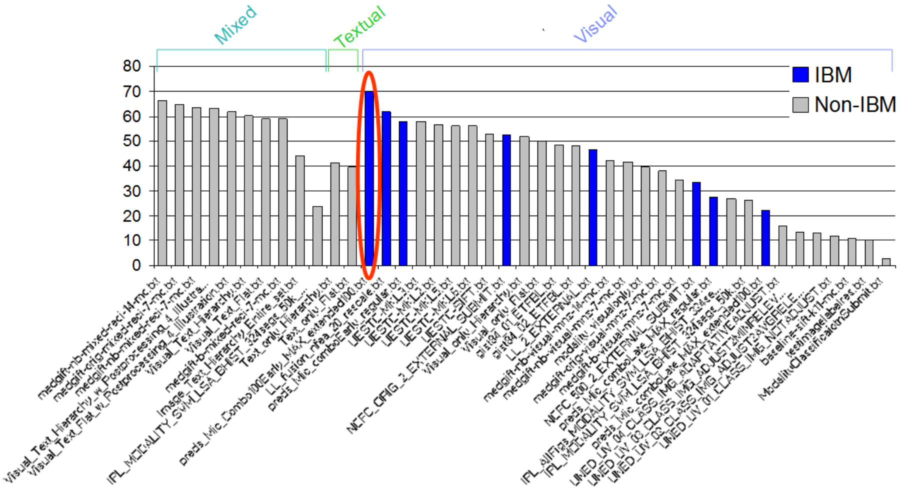

Fig. 5. Distribution of mean accuracy scores across all the submissions to ImageCLEF 2012. Runs are divided by type (visual only, textual only, mixed). IBM submitted only visual runs (in blue). IBM's runs achieved the top three visual only performance. IBM's top run (Multiclass SVM trained from extended data, Experiment 12, circled in red) performed even better than the mixed runs, thus achieving the best overall classification accuracy.

3 Case-Based Retrieval Task

In this section, we give an overview of the application of our methods to case-based medical image retrieval and present the results of our submitted runs.

3.1 Techniques and Approaches

Techniques used for case-based retrieval task are mainly based on methods from Information Retrieval (IR) and Natural Language processing (NLP), including rule-based and machine-learning techniques. In order to facilitate the classification of a large dataset and to retrieve the most relevant documents for a given query, an IR system applies various NLP methods to construct a semantic view of each document indexed via relational database or text index. This semantic view is summarized by a set of relevant keywords (i.e., index terms) as a signature of this document.

In the biomedical domain, the terminology is very important because the words used in the document are related to medical terms that can refer to the same concept with different semantic interpretation (i.e., senses) based on the textual context. In addition to the NLP techniques for reducing the size of the relevant keywords by eliminating stopwords and stemming words, our system also applies semantic similarity methods to improve the understanding of textual terms and remedy potential ambiguity among medical concepts. The focus of

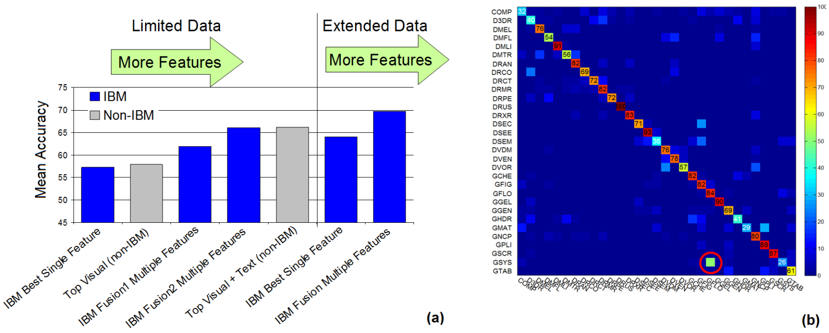

Fig. 6. (a) IBM runs grouped according to training dataset, number of descriptors adopted, and comparison with top performing non-IBM submissions. IBM's run with kernel approximation fusion trained on the original (limited in size) ImageCLEF dataset performs comparably to the top Mixed non-IBM run and largely better than other non-IBM visual only submissions. With additional training examples, IBM's system was able to significantly outperform all other runs, including Mixed ones. (b) Confusion matrix of the best IBM run (Experiment 12). "GSYS - System overview" resulted to be the hardest class to categorize, being quite often confused with "GFLO - Flowcharts".

the semantic similarity is to find the strength of the semantic relatedness or the semantic connections between textual terms. The taxonomic proximity between terms measures the degree of overlapping between contextual word vectors using Information Content (IC) based measures. To reduce computational complexity, the semantic relatedness is applied within an ordered window to find relevant terms between adjacent terms in the document.

3.2 Retrieval Framework

To build our retrieval framework based on the approaches we described above, we use the YTEX (Yale cTAKES extensions) system for computing the semantic IC-based measures and the cTAKES (clinical Text Analysis and Knowledge Extraction) as a NLP system based on the Unstructured Information Management Architecture (UIMA) that combines rule-based and machine-learning. The cTAKES system uses the OpenNLP Maximum Entropy package for sentence detection, tokenization (words), part-of-speech (POS) tagging. It performs the name entity recognition of biomedical from the Unified Medical Language system (UMLS) Metathesaurus, and other biomedical source such as Systematized Nomenclature of Medicine, Clinical Terms (SNOMED CT).

To classify the medical articles, our system used the YTEX semantic similarity to identify and disambiguate medical terms before storing the annotations on the documents in the relational database, and indexing relevant medical concepts that can facilitate the topic case query matching. For each case-based query, we first executed the NLP pipeline to extract relevant medical concepts and gen-

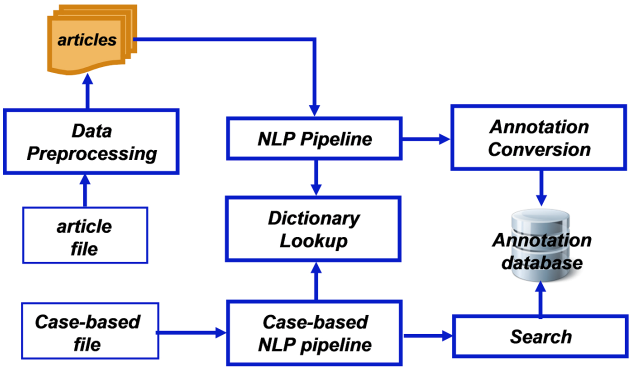

Fig. 7. Case Based System Overview

Table 5. Case-Based Results

RunID	Run Type	MAP	GM-MAP	bpref	P10	P30
ibm-case-based	Textual	0.0484	0.0023	0.0439	0.0577	0.0449

erate an SQL expression by combining the concepts using logical OR operator (meaning all of the concepts to be optional). To limit the number of query results and select the most relevant annotations, we define the weight measures for sorting and ranking them based on the cosine distance between medical concept vectors.

3.3 Experimental Results

Fig. 7 describes the processing flow of the system. The data processing splits the whole article file into multiple individual files in order to distribute them across multiple nodes. Each article is stored in the database as an annotation including its most relevant medical concepts. The case-based NLP pipeline processes each case-based file to find relevant medical concepts. Finally, the search engine composes a SQL logical expression and ranks the result set retrieved from the annotation database.

Due to time constraint, we only submitted one run for case-based retrieval task, shown in Table 5. First, we experimented with the semantic similarity approach, and found good correlation among medical concepts within the text corpus with appropriate senses. However, by matching only the medical concepts, the results are not as good as could be if we had used additional lexical databases.

3.4 Conclusion

The IC-based measures applied to adjacent words in a defined text window improves the semantic relatedness performance and is less expensive than computing all the word-pairs in the corpus. However, we also observed low system performance when the SQL expressions are complex and the number of concepts is high. In the future, we would like to improve the semantic similarity methods by incorporating the medical concepts with the lexical semantic analysis. We also want to have a better matching measures to improve the accuracy.

References

1. D. Lowe: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60, 2, pages 91110, (2004).
2. Aude Oliva and Antonio Torralba: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vision, 42(3):145 175, (2001).
3. T. Ahonen, A. Hadid, and M. Pietikainen: Face recognition with local binary patterns. ECCV, pages 469-481, (2004).
4. Cands, Emmanuel and Demanet, Laurent and Donoho, David and Ying, Lexing: Fast Discrete Curvelet Transforms. Multiscale Modeling and Simulation, 5 (3). pp. 861-899, (2006)
5. Shiraishi J, Katsuragawa S, Ikezoe J, Matsumoto T, Kobayashi T, Komatsu K, Matsui M, Fujita H, Kodera Y, and Doi K.: Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. AJR 174; 71-74, 2000
6. Yan R., Fleury M.O., Merler M., Natsev A., Smith J.R. Large-Scale Multimedia Semantic Concept Modeling using Robust Subspace Bagging and MapReduce. ACM Multimedia 2009
7. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
8. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
9. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
10. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)
11. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
12. National Center for Biotechnology Information, `http://www.ncbi.nlm.nih.gov`
13. Vedaldi A., Zisserman, A.: Efficient Additive Kernels via Explicit Feature Maps, IEEE Trans. Pattern Analysis and Machine Intelligence, 34(3), 2012
14. Kimeldorf, G.,Wahba, G.: Some results on tchebychefan spline functions. Journal of Mathematical Analysis and Applications 33 (1971) 82-95.
15. Cao, L., Luo, J., Liang, F., Huang, T. S.: Heterogeneous Feature Machines for Visual Recognition IEEE Proc. Int'l Conf. Computer Vision (ICCV), 2009
16. Ho, C.-H. Lin, C.-J.: Large-scale Linear Support Vector Regression, http://www.csie.ntu.edu.tw/c̃jlin/liblinear/
17. Chang, Chih-Chung and Lin, Chih-Jen: LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology,2(3), pages 1-27, (2011)