Active Learning & its Applications

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS by Research
in
Computer Science and Engineering

by

Priyam Bakliwal
201407643
priyam.bakliwal@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2017

To

My Mother and My Grandmother

whose unconditional love and support is the driving force of my life.

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Active Learning & its Applications" by Priyam Bakliwal, has been carried out under my supervision and is not submitted elsewhere for a degree.

05/07/17
Date

Adviser: Prof. C.V. Jawahar

Acknowledgements

My journey in IIIT-Hyderabad has been a wonderful experience. As I submit my MS thesis, I wish to extend my gratitude to all those people who helped me in successfully completing this journey.

I would first like to thank my thesis advisor Prof. C.V. Jawahar for all the guidance and support. He has been a source of continuous inspiration throughout my MS journey. His guidance has not only helped me become a good researcher but also a better person. Thank you for pushing me beyond what I thought were my limits. I could not have imagined having a better advisor and mentor for my MS.

I would also like to extend my special gratitude towards Dr. Manish Shrivastava, Dr. Manoj Kumar Chinnakotla and Dr. Vineeth N Balasubramanian for accepting to work with me. It was both an honour and a great privilege to work with them.

I don't have enough words to thank Arpita Das and Deepa Jagyasi. Without them, I could not have survived in IIIT-H. Its truly priceless to find friends with same mental disorder as you. Thank you Arpita for being my support system and tolerating my stupidity in all possible situations.

Working at CVIT was great fun. Thank you Yasaswi, Aditya, Suriya, Saurabh Sir, Praveen Sir, Jobin, Suranjana, Vijay, Govinda, Swetha, Anand Sir and all other CVITians for the wonderful interactions and fun work sessions. I am grateful to Mr. R. S. Satyanarayana, Varun, Rajan and Silar for all the support. Special thanks to Soumyajit, Shalki, Samruddhi, Nancy, Harish and Sumit for sharing your knowledge and for those cheerful experiences in IIIT.

This journey would not have been possible without my friends and their constant motivation - Somesh, Swati, Chanki, Manish, Vidhu, Gaurav, Roohi, Lucky and Surya. I thank Sumit, Abhishek, Avinash, Ramanand, Pinaki and Shoumill for being such amazing friends. Survival in Hyderabad would be very difficult without you people.

I could not have accomplished it without the support and understanding of my parents and my family. They have given me continuous encouragement throughout my years of study and through the process of researching and writing this thesis. Last, but not the least, thanks to IIIT community for giving me an inspiring environment and loads of opportunities to grow.

Abstract

Active learning also known as query learning is a sub-field of machine learning. It relies on the assumption that if the learning algorithm is allowed to choose the data from which it learns, it will perform better with less training. Active Learning is predominantly used in areas where getting a large amount of annotated data for training is not feasible or extremely expensive. Active learning models aims to overcome the annotation bottleneck by asking queries in the form of unlabelled instances to be labelled by a human. In this way, the framework aims to achieve high accuracy using very less labelled instances resulting in minimization of annotation cost

In the first part of our work, we propose an Active Learning based Image Annotation model. Automatic image annotation is the computer vision task of assigning a set of appropriate textual tags to a novel image. The aim is to eventually bridge the semantic gap of visual and textual representations with the help of these tags. The advantages of the proposed model includes: (a). It is able to output the variable number of tags for images which improves the accuracy. (b). It is effectively able to choose the difficult samples that needs to be manually annotated and thereby reducing the human annotation efforts. Studies on Corel and IAPR TC-12 datasets validate the effectiveness of this model.

In the second part of the thesis, we propose an active learning based solution for efficient, scalable and accurate annotations of objects in video sequences. We focus on reducing the human annotation efforts with simultaneous increase in tracking accuracy to get precise, tight bounding boxes around an object of interest. We use a novel combination of two different tracking algorithms to track an object in the whole video sequence. We propose a sampling strategy to sample the most informative frame which is given for human annotation. This newly annotated frame is used to update the previous annotations. Thus, by collaborative efforts of both human and the system we obtain accurate annotations with minimal effort. We have quantitatively and qualitatively validated the results on eight different datasets.

Active Learning is efficient in Natural Language documents as well. Multilingual processing tasks like statistical machine translation and cross language information retrieval rely mainly on availability of accurate parallel corpora. In the third section we propose a simple yet efficient method to generate huge amount of reasonably accurate parallel corpus using OCR with minimal user efforts. We show the performance of our proposed method on a manually aligned dataset of 300 Hindi-English sentences and 100 English-Malayalam sentences.

In the last section we utilised Active Learning for model updation in cQA system. Community Question Answering(cQA) platforms like Yahoo! Answers, Baidu Zhidao, Quora, StackOverflow *etc.* provides experts to give precise and targeted answers to any question posted by a user. These sites form huge repositories of information in the form of questions and answers. Retrieval of semantically relevant questions and answers from cQA forums have been an important research area for the past few years. Considering the ever growing nature of the data in cQA forums, these models cannot be kept stagnant. They need to be continuously updated so that they can adapt to the changing patterns of Questions-Answers with time. Such updation procedures are expensive and time consuming. We propose a novel Topic model based active sampler named *Picky*. It intelligently selects a smaller subset of the newly added Question-Answer pairs to be fed to the existing model for updating it. Evaluations on real life cQA datasets show that our approach converges at a faster rate, giving comparable performance to other baseline sampling strategies updated with data of ten times the size.

Contents

List of Figures

List of Tables

Chapter 1

Image Annotation

Automatic image annotation is the computer vision task of assigning a set of appropriate textual tags to a novel image. The aim is to eventually bridge the semantic gap of visual and textual representations with the help of these tags. This also has applications in designing scalable image retrieval systems and providing multilingual interfaces. Though a wide varieties of powerful machine learning algorithms have been explored for the image annotation problem in the recent past, nearest neighbor techniques still yield superior results to them. A challenge ahead of the present day annotation schemes is the lack of sufficient training data.

1.1 Introduction

With the outburst of social media there is a tremendous increase in unannotated raw image data getting archived everywhere. One often needs textual descriptions for these images to build scalable and semantically meaningful access methods. This needs new approaches for scalable and automatic image annotation. Automatic image annotation (here after referred to as simply image annotation) aims at assigning a set of appropriate textual tags to a new test image without explicitly understanding (eg. object detection, image categorization) the images. (See Figure 1.1 for an example.) Since there are many possible tags for a single image, the problem is very different from that of image classification/categorization. Annotation is essentially a multilabel classification problem. Usually, the annotation data sets have a large vocabulary of tags/labels and the objective is to pick and predict the most appropriate subset.

All the image annotation schemes are essentially based on machine learning. Many powerful methods (eg, based on CRF, HMM, SVM) were tried for this task in the past. However, in 2008, Makadia *et al.* [32] demonstrated that a simple nearest neighbor method can yield superior results on the popular data sets. In later years, Guillaumin *et al.* [31] as well as Verma and Jawahar [48] extend this method. They used metric learning [31, 48] and also refined the nearest neighbor computation process [48]. Even many of the later attempts with modern machine learning schemes [26, 49] did not yield results that are

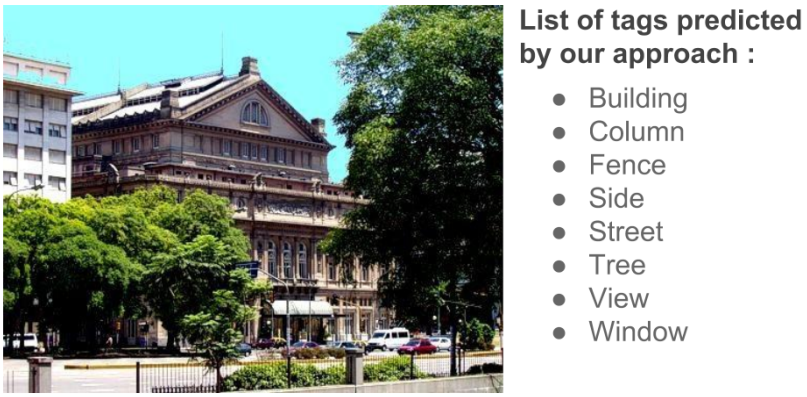


Figure 1.1: Automatic image annotation task. Result of our approach on an image from IAPRTC-12 data set. Our method predicts a set of appropriate tags with minimal amount of training data.

superior to the two pass nearest neighbor (2-PKNN) [48] over the popular databases such as Corel and IAPRTC-12.

1.2 Active Learning for Image Annotation

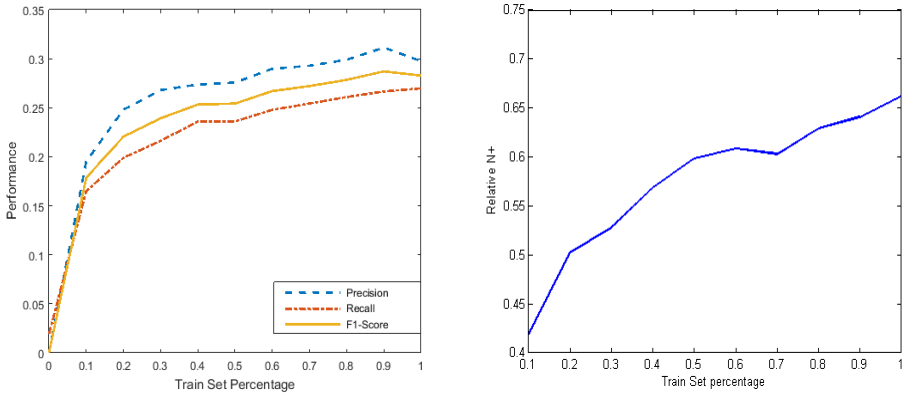
We propose an active Learning based image annotation model. We leverage the image-to-image and image-to-tag similarities to decide the best set of tags describing the semantics of an image.

Our objective is to obtain higher performance with minimal amount of training data. We achieve this with the help of active learning and automatically selecting images that are worthy of human labeling.

As we demonstrate in the next section, performance of the previous algorithms [32, 48] heavily depends on the number of labeled examples available for training. However practically, it is extremely difficult to get labeled examples. This can be overcome using active learning, where only a selected set of images is manually labeled to improve the performance. A general active learning process consists of two stages: (i) Learning algorithm and (ii) Sample selection algorithm. The performance of an active learning model highly depends on the sample selection scheme. As pointed out in [52] the most common criteria for sample selection are uncertainty, diversity, density and relevance.

We select the harder examples for manual annotations automatically. We use uncertainty for the sample selection. We handle the class imbalance problem by selecting a subset of training examples with similar frequency of labels rather than complete training set. This ensures that the most labeled classes do not dominate the results. We use a KNN classifier for its simplicity. We refine the 2-PKNN [48] scheme further for the active learning by enabling this scheme to predict variable number of tags for each of the images. For each image, we retain all the labels that satisfy a minimum threshold. An image is then considered for active learning based on the prediction score.

The advantages of the proposed model includes:



(a) (b)

Figure 1.2: Dependency of performance on number of train images. Note that the curves are not saturating and the annotation data needs many more examples.

1. It is able to output the variable number of tags for images which improves the accuracy.

2. It is effectively able to choose the difficult samples that needs to be manually annotated and thereby reducing the human annotation efforts.

We perform our experiments on two benchmark datasets, Coral and IAPRTC-12 and show that even with only 10% of train data and a proper selection of the additional examples to annotate, we can achieve a performances that is typically obtained by 80% of the training data. We report experimental results demonstrating the utility of our approach.

1.3 Proposed Approach

The focus of this work is to reduce the amount of training data used in annotation task with simultaneous increase of performance. We achieve this by formulating the problem in an active learning framework. The performance of a typical annotation algorithm depends on two factors: (a) feature representations and the underlying similarity metric, (b) size and quality of the training data set. It is self evident from Fig: 1.2 that performance of the annotation is directly proportional with size of the training set, i.e., larger the training data, better the performance. However, many of these training examples are "redundant" and do not add any value to the learning task. Identifying these examples can save the manual efforts which is among the primary goals of the current work.

Consider the training set $\mathcal{T} = \{(I_1, L_1), \ldots, (I_t, L_t)\}$ where, $I_1, \ldots, I_t$ are the images and $L_1, \ldots, L_t$ are its respective label sets. Let $\mathcal{Y} = \{y_1, \ldots, y_l\}$ denotes the vocabulary of labels. Given an unannotated image J, our aim is to assign multiple labels L_j associated with it.

Joint Equal Contribution (JEC) [32] treats image annotation as a retrieval problem. The technique uses a greedy algorithm to find image labels, from its nearest neighbors, found using low-level image features. The method is quite simple and intuitive which strongly claims that a simple combination of basic distance measures defined over commonly used image features can effectively serve as a baseline method for multilabel image annotation tasks. Despite its simplicity, at the time of its proposal, JEC held the state of the art on all benchmark annotation datasets [32]. However, it fails to consider class imbalance and weak labeling issues explicitly. The former problem is tackled by 2-PKNN algorithm [48] by using annotation performance in terms of mean recall. It uses 2-phase nearest neighbor model to predict image annotations. Given an unseen image, the algorithm identifies its semantic neighbors, in the first phase, for all the labels. And in the second phase the selected samples are used to predict the tags.

In our proposed method, for identification of semantic neighbors, we pick K images for each semantic label in the vocabulary that are most similar to J. In this way we ensure that each label appears at least K times in the training data. Let T_{Jx} be the set of K images, that are most useful in predicting the score of label y_x for image J. These neighbors incorporate image-to-label similarities. Once all T_{Jx} are determined, we merge them to form a final subtrain set specific to image J. It can be easily seen that this setting addresses the class imbalance issue by choosing each label to appear at least K times in train data. With this train data, we apply a weighted nearest neighbor algorithm to assign importance to the labels based on image similarity. In this way, we determine the scores for each label y_x for the image J.

Most of the previous works [31, 48] keep a static size of tags to be predicted (generally 5). This helps to retain higher precision. However, in this algorithm we dynamically determine this number. For each test image our algorithm predicts the score for every label in the dictionary. We assign all the labels that satisfy the 'score threshold'. For calculating the score threshold, we randomly sample a set of train images and use our algorithm to find the scores for each label. Later, scores for all the labels present in train ground truth is used to calculate the mean score (S_m), this is then used to calculate the score threshold, to determine the presence of label in an image. The final threshold can be calculated as:

$$\tau = S_m - \epsilon \tag{1.1}$$

where, ϵ is the tolerance parameter that decides tradeoff between precision and recall. Large value of ϵ leads to smaller τ and thus more number of labels will be assigned to the image leading to higher recall.

For an unseen test image J, the 2PKNN algorithm is used to determine score for each of the labels. All labels L_i with score S_{Ji} greater than τ are kept as predicted labels for the image J.

For active learning we have to decide the images that need to sampled from the test set. As mentioned earlier we use uncertainty principle to select these images to maximize the performance. We take mean of the scores of the predicted tags to decide the prediction confidence for the image.

$$\theta_J = \frac{1}{n} \sum S_{Ji}, \forall S_{Ji} \geq \tau \tag{1.2}$$

We greedily select $X\%$ of images with the lowest mean score from the total test set. These are the images used for active learning which are combined with existing training set. We recalculate the scores for remaining test images and assign its tags based on the updated scores. This summarizes one iteration of active learning. Thus, with every iteration we improve the model and predict the tags with higher accuracy. Algorithm 1 explains the proposed method algorithmically.

Algorithm 1 Active learning algorithm for Image Annotations

Input: Trainannotations, DistanceMatrix, τ, X

Output: AssignedTestLabels

for $i = 1$ to numOfTestImages **do**
  Select Train Subset.
  Calculate the scores using KNN and Train Subset.
  for $j = 1$ to numOfLabels **do**
    if ($S_{ij} \geq \tau$) **then**
      Add j to AssignedTestLabels and
    end if
  end for
  $meanScore_i = mean(Score(AssignedLabels))$
end for
Choose X images with lowest meanScore
Ask for user annotations for these images.
Add them to train set.
for $i = 1$ to numOfRemainingTestImages **do**
  Recalculate Train Subset and AssignedTestLabels
end for

1.4 Experiments and Results

1.4.1 Data sets, representation and evaluation measures

We have used two popular image annotation data sets namely, Corel and IAPRTC-12. Corel data set was first used in [35] and since then it has been one of the benchmark data sets in image annotations. IAPRTC-12 data set was first used for cross-lingual retrieval in [20]. In this data set each image is

Data set	Total number of Images	Number of Labels	Number of Train Images	Number of Test Images
IAPRTC-12	19627	291	17665	1962
Corel	4999	268	4500	499

Table 1.1: Details about the data sets used for the experiments of Image Annotation task

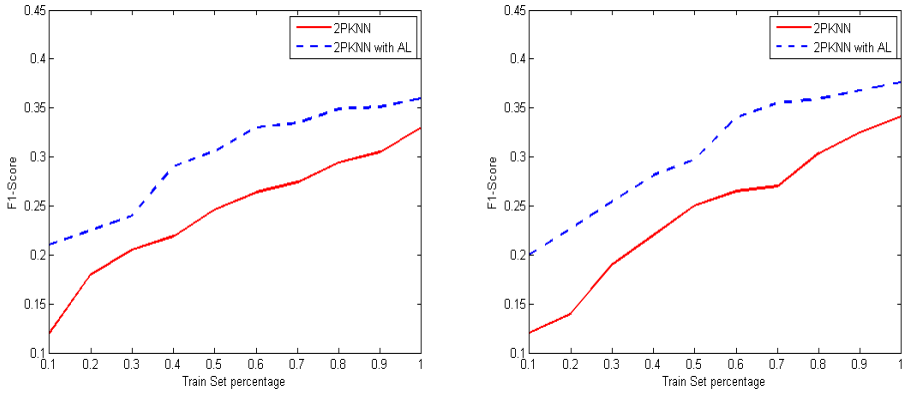
	Corel 5K				IAPR TC-12			
Method	P	R	F1	R-N+	P	R	F1	R-N+
JEC[32]	27	32	29.3	53	28	29	28.5	85
TagProp[31]	31	37	33.7	56	48	25	32.9	78
KSVM[49]	32	42	36	68	47	29	36	92
2PKNN [48]	39	40	39.5	68	49	32	38.7	94
2PKNN with AL	**45**	**46**	**45.5**	**80**	**56**	**32**	**41**	**97**
2PKNN[48]	35	32	33	63	43	29	34	93
2PKNN with AL	**36**	**34**	**35**	**67**	**47**	**31**	**37**	**95**

Table 1.2: Performance comparison among different image annotation methods. 2PKNN with AL (this work) uses 10% of test data for active learning. The top section shows performance calculated on full test data and the bottom section shows performance for only 90% test data, excluding 10% of actively learned test data.

described in detail; but for image annotation task, only nouns are extracted as labels. Table 1.1 describes the basic characteristics of the data sets used for experimentation.

For performance analysis of this algorithm, we have used 15 distinct descriptors as used in [31]. These include both global and local features. SIFT and Hue based descriptors covers local features of an image whereas GIST and histogram based descriptors encode the overall characteristics of the image. Distances between the features are calculated following the earlier research [31]. L1 measure is used for colored histograms, L2 for GIST and χ^2 for the SIFT and Hue descriptors.

To compare the result with earlier methods, we have used similar evaluation measures as in [48]. Given an unseen image we predict the label set using the score and the threshold. The evaluation measures include 1) Average precision per label 2) Average recall per label 3) Average F1-score per label and 4) Normalized N+ Score. If for a label l_x there are i_g images in the ground truth and i_p images



(a) (b)

Figure 1.3: Change of F1-Score with increase in number of train images for both Corel and IPAR TC-12 datasets.

are predicted for it then, for label l_x, precision P_x and recall R_x are defined as:

$$P_x = \frac{i_g \cap i_p}{i_p} \qquad \text{and} \qquad R_x = \frac{i_g \cap i_p}{i_g} \tag{1.3}$$

The average of these precision and recall values over $\mathcal{Y}$ gives us mean precision and recall for the data set. To analyze the trade-off between precision and recall we also calculate mean F1-Score as:

$$F1 = \frac{2 * P * R}{(P + R)} \tag{1.4}$$

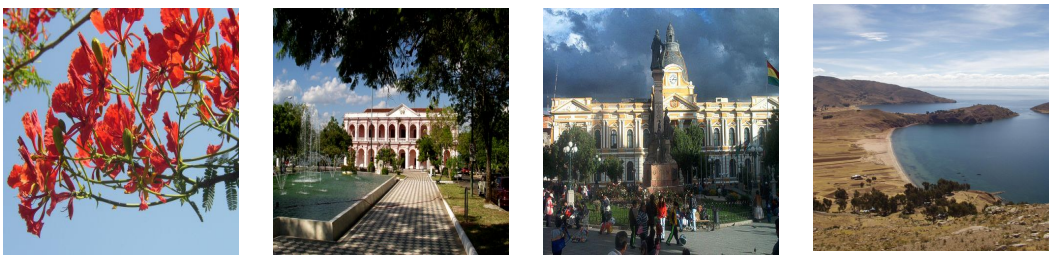
N+ is the number of labels that are assigned correctly to at least one image. But, instead of using absolute value for N+, we are using relative N+, i.e.

$$R_{N+} = \frac{N+_{Score}}{Numberof Labels} \tag{1.5}$$

1.4.2 Empirical Results

To compare the image annotation performance with other methods, we use the predefined training and test sets used in the past [31, 32, 48]. The summary or our results as well results for previous models are summarized in Table 1.2. We have used active learning along with 2PKNN algorithm to show that there is a significant performance gain even with using only 10% of data for active learning.

Now we study the performance variation with the size of the training set. It is a known fact that the performance of nearest neighbor algorithms increases with increase in training data. We have calculated the performance change with gradually increasing the size of training data for both 2PKNN algorithm and active learning algorithm. The active learning percentage was kept as 10. It is clear from Fig 1.3 that learning rate is comparatively high for Active learning method.



Tree	Tree, Fountain	Building, Center, Clock, Flag, Column, Fence, Lot, Statue, People, Square, Window	Bay, Beach, Cloud, Slope, Stone, Tree, View
Tree, Flower, Fruit, Sky, Life	Building,Fountain, Tree, Footpath, Paving	Building, Carpet, Lot, Stair, Tree	Stone, Jetty, Lake, Summit, Mountain
Tree, Flower, Sky	Tree, Building, Fountain	Building, Column, Carpet, Lot, People, Stair, Tree, Fence	Bay, Jetty, Mountain, Tree, Beach, Cloud, Lake, Stone, Summit

Figure 1.4: Qualitative Results are shown on IAPR TC-12 dataset. The second row respresents tags present in ground truth. Third and fourth rows represents tags predicted by 2-PKNN and our algorithm respectively. Prediction of 'Sky' and 'Flower' in image 1 shows that we handle weak labeling. Also the prediction of 3, 3, 8 and 9 tags for first, second, third and fourth image respectively shows the effectiveness of varying length annotations.

In this experiment we gradually increased the quantity of test images picked for active learning. Also, to show that we effectively pick up most uncertain samples, we randomly sample the same quantity of test images and added to the train set to calculate performance. Fig: 1.5 clearly depicts that the sample selection strategy of our proposed algorithm outperformed with extremely better results.

Use of fixed length annotation method faces disadvantages in the cases where annotation length is either comparatively less or more than the mean annotation length. We have shown in Fig 3 that we are able to predict better tags for images with as low as 1 tag as well as for images with 11 tags. It is also evident from $Image_1$ that we are handling missing label issue.

1.4.3 Discussions

We have clearly shown with our experiments that the active learning algorithm we propose has multiple advantages in terms of performance gain as well as reduction in requirement of train data. On

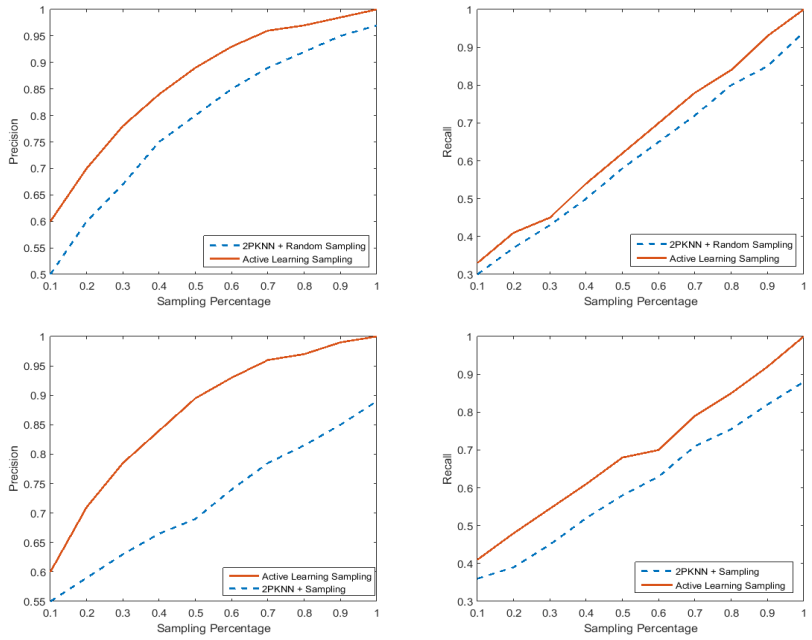


Figure 1.5: Performance graphs to depict effective selection of active learning samples. The first two images are for Corel dataset and later once are for IAPRTC-12 dataset. We have used random sample selection for 2PKNN algorithm and the selected samples are added to the train set vs our selection strategy to compare the performances.

one hand with active learning we reduce the time and cost required for getting train data and on the other hand with dynamic decision of tag length and greedy selection of most uncertain images we improve the performance.

1.5 Summary

We have proposed an active learning based image annotation model. This model combines nearest neighbor approach along with active learning to annotate an unseen image. One of the most important issues in auto image annotation is deciding the annotation length, and our algorithm gives a simple and effective solution to this issue . Also, it is clear from the results that the algorithm is effectively able to pick hard samples from the test data, so as to dramatically improve the overall accuracy of the test samples even with extremely less actively annotated images. Thus with minimal user efforts, we are able to outperform extremely well. Currently, we have used fixed thresholds for deciding the annotation length. In future, we are planning to learn the thresholds from training data. Also, we can use weighted thresholds based on the properties of the predicted annotation labels.

Chapter 2

Generating Annotations for Tracking

Visual tracking is a rapidly evolving field of computer vision. The aim of tracking is to estimate the location of the object in each frame of the image sequence . Tracking is a part of many higher-level problems of computer vision, such as motion analysis, event detection and activity understanding. Availability of highly accurate labeled data is required for for both training and evaluation of many of such applications. Recent computer vision solutions use machine learning. Effectiveness of these solutions relies on the amount of available annotated data which again depends on the generation of huge amount of accurately annotated data. Generation of such data is expensive both in terms of time and effort.

2.1 Introduction

With increase in use of surveillance cameras and decrease in cost of storage and processing of surveillance videos, there is a huge availability of unlabeled video data. This data can be utilized in many high level computer vision tasks such as motion analysis, event detection and activity understanding. Computer vision models that do video analysis [61, 62] require accurately annotated data for both training and evaluation. However, annotating massive video sequences is extremely expensive and may not be feasible.

The use of tracking algorithms to generate annotated data lack in terms of detection accuracy and reliability making them unsuitable for critical applications like surveillance systems, transport, sports analysis, medical imaging, etc. Most of the recent algorithms [19, 46, 66] use the appearance model as a prerequisite for the success of a tracking system. It is extremely challenging to design a robust appearance model which can be adaptive to all the working conditions like partial/full occlusion, illumination changes, motion blur, shape changes, etc.. These methods do give us a significant improvement in tracking output but they are still not reliable enough to be used for generation of annotated data.

There have been many attempts in the past to generate annotated data from videos. However, these methods are not often used for large industrial scale annotations because they usually lack annotation consistency and accuracy. In most cases [27], human annotators mark the object of interest in a video

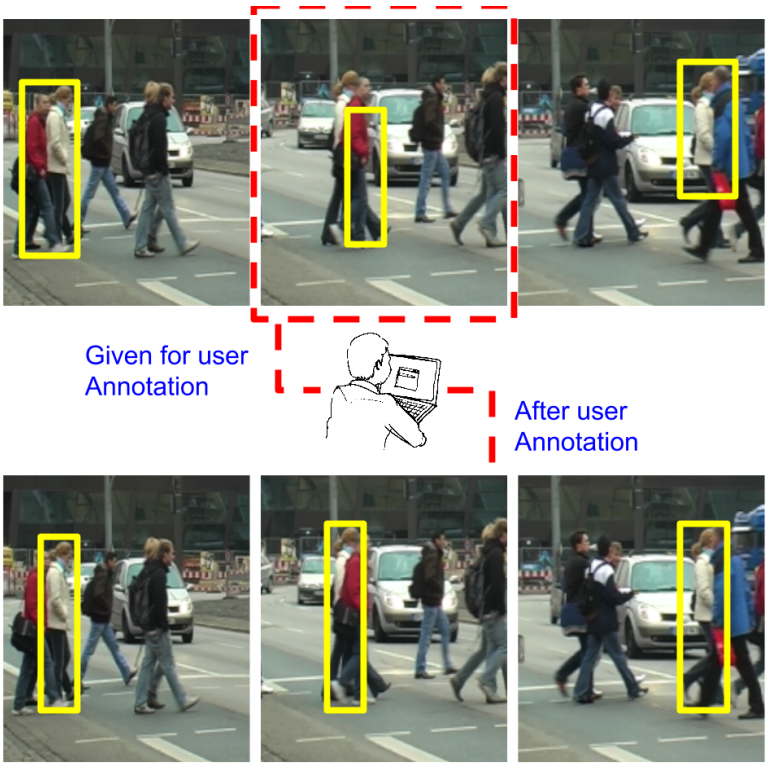


Figure 2.1: Use of Active Learning for object tracking in video sequences. Top row shows the tracking output on 3 frames of TUD-Crossing(4) sequence. The tracker fails to track the person in white jacket accurately. Our sampling technique selects the most informative frame (shown in red rectangle) for user annotation. The proposed algorithm ensures more accurate tracking with minimal user efforts. Bottom row shows better predictions for entire sequence with only one user annotation.

sequence. As pointed out in [50], manual annotations, involve a huge cognition load, and is subjected to inefficiency and inaccuracies. Some efforts that use crowd sourcing to increase the number of annotations, mainly for building large corpora [13, 34, 41], suffer from inconsistent annotations as most workers are poor annotators. In video annotation, the marking consistency of an annotator is extremely important as it becomes difficult to capture the marking variation in shape and extent of object within neighboring frames. Thus, crowd sourcing mandates robust quality control protocols.

Due to extremely high cost of human annotation for large video datasets, much of the research efforts have been dedicated towards leveraging the use of unlabeled data. Many algorithms developed recently are using semi-supervised learning [15, 29], or weakly-labeled data [46], which is faster to annotate. All of these algorithms aim at reducing the number of annotations needed.

Some works focus directly on making the annotation process itself more efficient for the human annotators. In [25] the authors propose a method for categorizing images using binary queries rather than asking annotators to select the category from some predefined list. However, such systems are not suitable for annotating image sequences.

The video annotation framework proposed by Vondrick and Ramanan [51] is based on the video annotations using active learning. In this system annotations are derived by tracking results and active learning is used to intelligently query the human annotator for corrections on the tracks. Angela *et. al.* [1], uses an incremental learning approach which continuously updates an object detector and detection thresholds, as an user interactively corrects annotations proposed by the system. In their work, the learning approach is paired with an active learning element which predicts the most difficult images. Their solution is purely based on detection and does not consider the tracking. However, our approach incorporated tracking into detection, making it more robust while ensuring minimal annotation effort.

Works in similar lines includes [7, 21, 56] which utilize active learning for video indexing and annotation. However, they do not incorporate the power of existing efficient tracking algorithms to create a robust and accurate framework for real time object detection in video sequences.

2.2 Proposed Approach

We propose an active learning based solution for efficient, scalable and accurate annotations of objects in video sequences. In this paper, we focus on reducing the human annotation efforts with simultaneous increase in tracking accuracy to get precise, tight bounding boxes around an object of interest. We use a novel combination of two different tracking algorithms to track an object in the whole video sequence. We propose a sampling strategy to sample the most informative frame which is given for human annotation. This newly annotated frame is used to update the previous annotations. Thus, by collaborative efforts of both human and the system we obtain accurate annotations with minimal effort. Using the proposed method, user efforts can be reduced to half without compromising on the annotation accuracy. We have quantitatively and qualitatively validated the results on eight different datasets.

Contributions: In this work, we use tracking algorithms to detect the objects in multiple frames and active learning is used to improve the correctness of the tracks. We propose *(i)* an effective tracking algorithm and *(ii)* an adaptive key-frame strategy that use active learning to intelligently query the annotator to label the objects at only certain frames which are most likely to improve the performance. The proposed active learning strategy can also be used in other computer vision tasks.

We propose a framework that can easily incorporate various tracking algorithms, making it more generalized. Multiple tracking algorithms (2 in our case) are combined efficiently to produce a reliable and accurate track for the object.

One of the major contributions of this method is consideration of neighborhood in selection of key frames. Also, we have used 'Query by Committee' strategy for key frame selection. Consideration of temporal neighborhood makes sure that with each user annotation, the tracking is best updated for neighboring frames as well. The advantages of our method includes easy incorporation of tracking algorithms, automatic detection of key frames thereby drastically reducing human efforts and scalable annotation process. This makes our approach suitable for annotations of large video datasets.

We performed experiments on objects of multiple datasets and show that the user efforts for doing annotation can be reduced up to 50% when using the proposed active learning strategy without compromising on tracking accuracy. We also show that with the same amount of user efforts the proposed method achieves an improvement of up to 200% for tracking task. We report experimental results on 8 different datasets consisting of more than 2500 frames and 17 objects. The consistent improvement in all scenarios demonstrate the utility of our approach.

2.3 Tracking Algorithms

We employ three tracking algorithms in this work. The most simple uses bi-linear interpolation which does not consider object characteristics and predicts tracks using initialization only. The other two algorithms are the modification of two state of the art tracking methods, Weighted Multiple Instance Learning Tracker (WMILT) [58] and Discriminative Scale Space Tracker (DSST) [10]. WMILT uses weighted instance probabilities to detect object of same size in other frames. On the other hand, DSST uses discriminative correlation filters based on a scale pyramid representation to track the object. DSST algorithm is scale invariant while the WMILT algorithm works for objects with not much scale change.

2.3.1 Bi-linear Interpolation

Interpolation is the basic approach to the problem of object tracking. In simple terms, the linear interpolation of two known points given by the coordinates (x_0, y_0) and (x_1, y_1) is the straight line between these points. In the problem of video annotation, the main criteria is how to decide the key frames. In this approach the user is asked to annotate every n^{th} frame and rest of the frames are simply tracked using interpolation. There is a trade-off between tracking accuracy and annotation cost. Smaller value of n leads to better track but higher annotation cost.

2.3.2 Bidirectional WMILT

Weighted Multiple Instance Learning Tracker (WMILT) [58] integrates the sample importance into the learning procedure. A bag probability function is used to combine the weighted instance probability. The algorithm weighs the positive instances according to their importance to the bag probability, it assumes that the weight for the instance near the target location is larger than that far from the target location.

The algorithm relies on positive and negative samples. The positive samples and negative samples are separated into two bags. The initialized target is labeled as positive. The contribution of each positive sample is calculated using a monotone decreasing function with respect to the Euclidean distance between the locations of sample and target. In this way the tracker integrates the sample importance into the learning procedure.

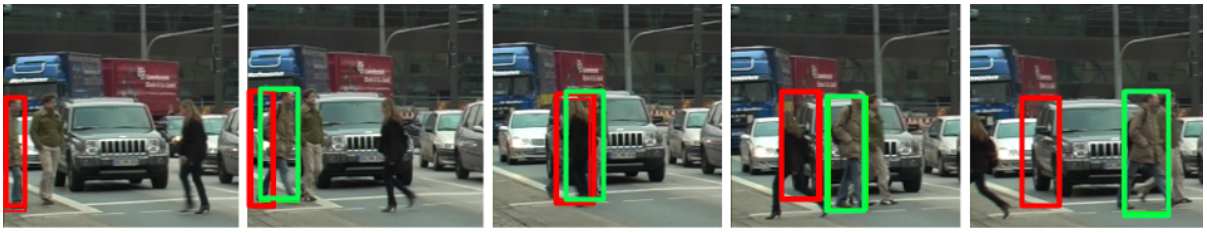


Figure 2.2: The behavior of proposed method 'Collaborative Neighborhood Tracker' in case of occlusion. First frame is the initialization frame and red bounding boxes are the initial tracking output. The algorithm selects middle frame as the key frame. After user annotation, the track updates (shown in green). Clearly, the tracking algorithm is handling occlusion well and the key frame selection is improving the overall track.

Intuitively, all the instances in the negative bag are very far and completely dissimilar to the target. Therefore, the algorithm treats all negative instances to contribute equally to the negative bag. Finally, a bag log-likelihood function is used to find the instance for which the probability is maximized. The algorithm efficiently integrates the sample importance into the learning procedure to detect the similar sized target in rest of the image sequence. We have used the algorithm to detect the object location both in backward as well as forward image sequence resulting in higher tracking accuracy.

2.3.3 Bidirectional DSST

The DSST algorithm [10] extends discriminative correlation filters [5] to multi-dimensional features for visual object tracking. We utilize this method to predict the target locations in both the temporal directions of the video sequences, so as to improve the prediction.

The algorithm uses HOG features along with image intensity features. An image is represented as d-dimensional feature map from which a rectangular target patch is extracted. An optimum correlation filter is found by minimizing the cost function. We build a 3-dimensional scale space correlation filter for scale invariant visual object tracking. The filter size is fixed to $M \times N \times S$, where M and N are height and width of the filter and S is the number of scales. A feature pyramid is constructed from a rectangular area around the target and the pyramid is centered at the target's location and scale. A 3-dimensional Gaussian function is then used to get the desired correlation output.

This correlation filter is used to track the target both in previous and next frames of the image sequence. Given a new frame, a rectangular cuboid of size $M \times N \times S$ is extracted from the feature pyramid. Similar to above, the cuboid is centered at the predicted location and scale of the target. We compute the correlation scores and the new target location and scale is obtained by finding the maximum score.

2.4 Active Learning Baselines

Generally, annotating massive videos is extremely expensive. There are hundreds of hours of surveillance video footage of cars and pedestrians which will require a lot of human effort to annotate. Currently video annotations are done typically by having paid users on Mechanical Turk labeling a set of key frames followed by linear interpolation [55].

2.4.1 Interpolation with key frame selection

We extend the interpolation based tracking by adding dynamic key frame selection strategy. As discussed earlier, the interpolation method is highly dependent on key frame selection interval n. The optimum value of n vary from object to object. For example, consider an object moving at a constant pace for few frames and then change speed during later frames. Such cases makes it hard to find a single optimum value of n for even one object.

A slight modification in naive linear interpolation approach can significantly reduce the human efforts. We have designed a tool that initially asks the user to annotate first and last frame of the video sequence. the tool calculates the object track using linear interpolation. It also gives flexibility to the user to decide which frame to annotate next so as to improve the tracking accuracy. This avoids the problem that occurs due to fixing the n for a given object.

2.4.2 Uncertainty based Active Learning

One of the simplest and most intuitive key frame selection strategy is uncertainty sampling. In this method, the algorithm queries the frames about which the tracker is least certain. in this approach we use both the tracking algorithms, *viz.*, WMILT and DSST, separately. For WMILT, we use classifier probability as the measure to define uncertainty. Whereas, to calculate uncertainty for DSST, we consider both tracker's translation and scale correlation confidence scores. The frame with minimum tracker's score / confidence, is considered as the next frame for user annotation. The tracker's output is updated after every user annotation.

2.5 Collaborative Tracking

We propose a new collaborative approach to improve tracking accuracy while ensuring minimal user efforts. We use the tracking algorithms described in section 2 and combine them in a novel way to get an enhanced hybrid tracking algorithm. The DSST being scale invariant algorithm is complimentary to WMILT algorithm which detects similar sized objects. Hence, a combination of both gives a higher tracking accuracy.

2.5.1 Collaborative Tracker

We have collaborated the two trackers (WMILT and DSST) into a new tracker named 'Collaborative Tracker'. We represent the target as a bounding box enclosing its spatial extent within a video frame. The bounding box is represented as a 4-dimensional vector representing top-left and bottom-right corner coordinates of the target. Let the predicted bounding boxes of above two trackers be P_w and P_D. Then the collaborated output is given by:

$$P_C = \epsilon P_W + (1 - \epsilon) P_D \tag{2.1}$$

where ϵ ($0 \leq \epsilon \leq 1$) is the weight assigned to the individual tracker outputs. The value of ϵ is fixed at the start of the annotation. If the size of object across the whole track is constant and does not vary much, the value of ϵ is greater than 0.5 else it is less than 0.5.

We also propose an adaptive frame sampling scheme which uses active learning to intelligently asks the human annotator to annotate the target only in few specific frames that are likely to improve the performance. This approach is based on the fact that for any tracking algorithm, not all the objects/videos can be treated equally.

The performance of any tracking algorithm can vary significantly depending upon the scenario in the video. Some objects are comparatively easier to annotate automatically. For example, the frames in which a person is standing. In such cases, only one frame initialization might give required track. However, the more complex scenarios require a lot more annotation efforts to get the desired track. Thus, the proposed key frame sampling scheme helps to utilize the annotation efforts on more complex objects (or frames) that are visually ambiguous, such as occlusions or sudden change of appearance (see Fig 2.2).

Suppose at time t, the task is to figure out the frame that the user should annotate next. We utilize the difference in opinion principle to determine the next frame. The center (C) of the predicted bounding box (P) is given by:

$$C = \frac{P_1 + P_3}{2}, \frac{P_2 + P_4}{2} \tag{2.2}$$

where P_1 and P_2 are coordinates of top left corner of the bounding box and P_3 and P_4 are coordinates of bottom right corner. For a frame i, we determine the difference in center predictions D_i^t for all three algorithms at time $t-1$ using:

$$D_i^t = C_{ic}^{t-1} \oplus C_{id}^{t-1} \oplus C_{iw}^{t-1} \tag{2.3}$$

where,

$$a \oplus b \oplus c = dist(a, b) + dist(b, c) + dist(c, a) \tag{2.4}$$

$dist(x, y)$ is the Euclidean distance between x and y. Next, we determine the frame that best helps in improving the tracker output for all other frames. We select the most useful key frame as the frame with largest center difference as per 'Query by Committee' strategy. The key frame f^t at any time t is found using:

$$f^t = \arg\max_i (D_i) \tag{2.5}$$

These key frame annotations are used to track the object using different tracking algorithms and ultimately each track adds up to the accuracy of the final output. Intuitively, the track of an object at a particular frame is more accurate when the initialization is done in the near by frame. Therefore, for every frame we use the tracking output for the iteration where (initialization) key frame is closest. The tracker output for frame i at time t (P^t_{Ci}) is calculated using Eq2.1 as:

$$j = \arg\min_{k=1}^{t-1} |f^k - i| \tag{2.6a}$$

$$P^t_{Ci} = \epsilon P^j_W + (1-\epsilon) P^j_D \tag{2.6b}$$

The algorithm finds out most uncertain frame and asks the user for its annotation. The correction of this frame results in overall improvement of the tracking accuracy. Therefore, with every iteration the tracker improves and the algorithm updates the object positions, thereby making the prediction more accurate.

2.5.2 Collaborative Neighborhood Tracker

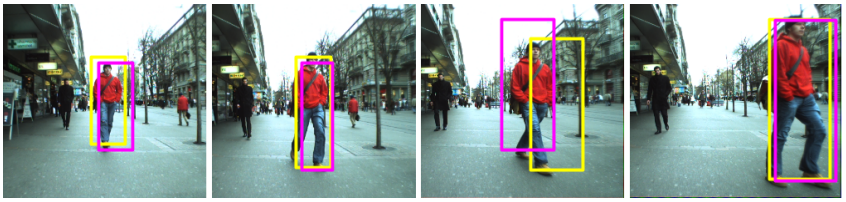
In this approach, we consider the uncertainty of temporal neighboring frames to determine the next key frame. The intuition behind this is that every user annotation should improve the object location in the whole video sequence and not just the current frame (See Fig 2.3). Thus, we consider the temporal neighborhood center difference along with the current frame's center difference to decide the next key frame to be given for user annotation. This makes our sampling scheme robust. In this approach, the key frame selection is done using:

$$f^t = \arg\max_i (\sum_{j=1}^{T} \eta e^{-|j-i|} D_j) \tag{2.7}$$

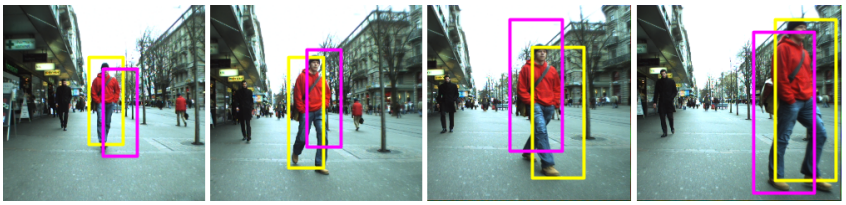
where η is a normalization constant and T is the total number of frames in the sequence. All the frames in the sequence are considered in the neighborhood of every key frame, more closer the neighboring frame to the key frame the greater is the impact of its center difference. Similar to Collaborative Tracker, the Collaborative Neighborhood Tracker is expected to become more accurate with each user annotation. In this case, the improvements are expected to be more because the selection of key frame is based on collective uncertainty of temporal neighborhood making the selection process more informative. Finally, Eq 2.6 is used to give the final track of the object.

2.6 Experiments

We have performed multiple experiments on eight different publicly available datasets and show the effectiveness of the proposed algorithm in various scenarios.



(a)



(b)

Figure 2.3: Importance of consideration of neighborhood frames while key frame selection. The output of two different trackers are shown in separate color bonding boxes. To decide next key frame to annotate there are two ways: (a) Based on tracker disagreement of candidate frame. (b) Based on tracker disagreement of candidate neighborhood. Clearly, second scenario is better to be given to user for annotation. User annotation of its third frame will update the tracking output of neighbors as well resulting in better track and more reduction in error.

2.6.1 Datasets and Evaluation Measures

We have used 17 sequences from standard tracking datasets like ETH-Bahnhof, ETH-Jelmoli, ETH-Sunnyday, TUD-Campus, TUD-Crossing, TUD-Stadmitte, David, Couple, etc. to evaluate the performance of the proposed technique. The video sequences pose several challenges such as illumination changes, size and pose changes, motion blurs, partial and full occlusions etc. to tracking algorithms. There exists an abundance of performance measures in the field of visual tracking. The following notations are used:

◇ **Input Format:** $FrameNum$**,** x_1**,** y_1**,** x_2**,** y_2

◇ **Output Format:** $FrameNum$**,** x_3**,** y_3**,** x_4**,** y_4

We have selected the following criteria to measure the performance and provide comparisons among different tracking algorithms.

Average Error: Average Error is the mean of difference between each side of the bounding box generated by the tracker $\{(x_3, y_3), (x_4, y_4)\}$ and the ground truth $\{(x_1, y_1), (x_2, y_2)\}$.

$$AvgError = (|x_1 - x_3| + |x_2 - x_4| + |y_1 - y_3| + |y_2 - y_4|)/4 \tag{2.8}$$

Edge Error: An edge error occurs if the difference between an edge of the bounding box generated by the tracker and the ground truth is more than 5 pixels, i.e., if $|e_i^{Tracker} - e_i^{GT}| \geq 5$ then edge error

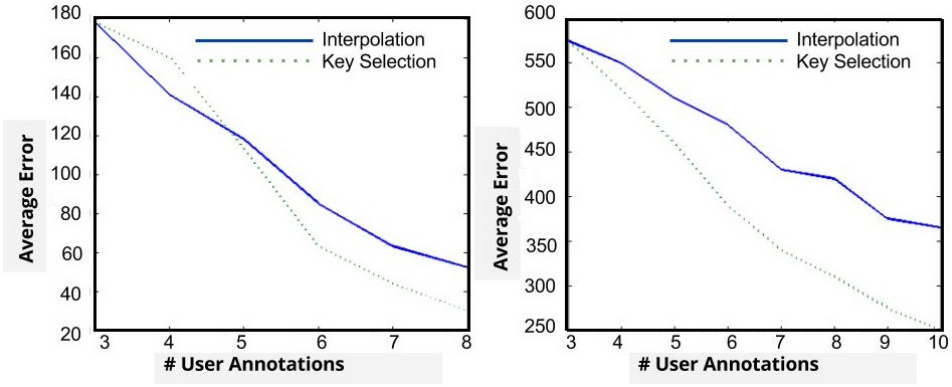


Figure 2.4: Change in Average Error with the number of user annotations for Liner Interpolation and Key Frame Selection(M1) for TUD-Campus(2) and TUD-Crossing(8).

is 1 else 0 (where, $e_i^{Tracker}, e_i^{GT}$ are edges of tracker and ground truth). The edge error is then summed up for all 4 edges of the bounding box.

Centroid Error: Centroid Error is the Euclidean distance between centroids of bounding boxes of the tracker and the ground truth. Centroid error is calculated as:

$$CentroidError = \sqrt{(\bar{c_x} - c_x)^2 + (\bar{c_y} - c_y)^2} \tag{2.9}$$

where, $(\bar{c_x}, \bar{c_y})$ are ground truth centroid coordinates and (c_x, c_y) are tracker centroid coordinates.

2.6.2 Comparison of various Active Learning Strategies

The main aim of video annotation framework is to generate the track for different objects with minimal user interaction. In this section we describe several experiments to show the effectiveness of the proposed algorithm. We have referred Key Frame Selection as M1, Uncertainty (WMILT) as M2, Uncertainty (DSST) as M3, Collaborative Tracker (proposed approach) as M4 and Collaborative Neighborhood Tracker (proposed approach) as M5. Also for datasets we have used notations, TC for TUD-Campus, TCr for TUD-Crossing, EJ for ETH-Jelmoli, ES for ETH-Sunnyday and EB for ETH-Banhof. As mentioned earlier the value of ϵ depends on the variations in the size of object across the whole track. We have used different values of ϵ for each object in the dataset. For objects such as ETH-Banhof(2,3) where the variation in object size is much we have used lower values (0.20, 0.22), where as, for objects like Couple and David the value of ϵ is higher (0.75,0.80).

In this experiment we compare the traditional annotation technique with the active based method. Existing video annotation framework [55] typically have users labeling frames at regular intervals followed by linear interpolation. Fig 2.4 shows the decrease in average error with increase in number of user annotations for Linear Interpolation and Key Frame Selection(M1). Clearly, the decrease in error shows that the active learning based solution is better than traditional annotation technique.

In this experiment, we show that the proposed tracker is suitable for mission critical applications like automotive surveillance. For such applications limb precision is very important. The limb pre-

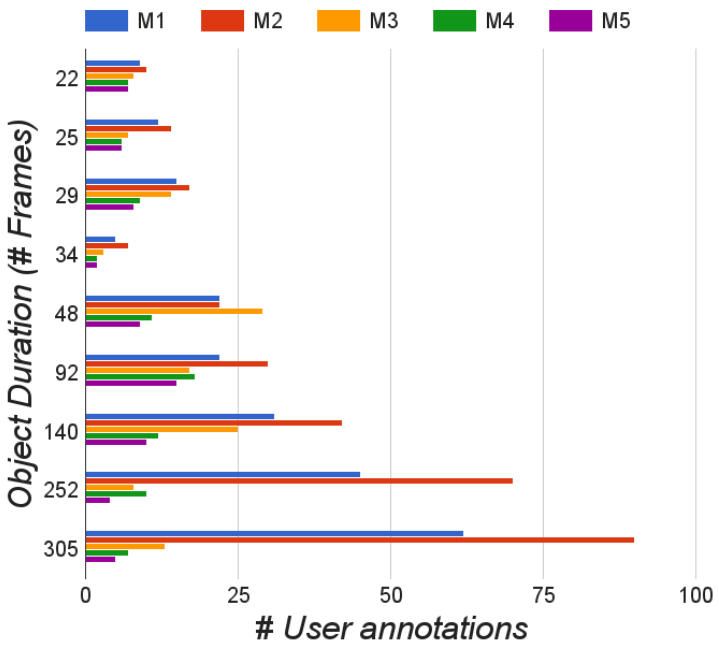


Figure 2.5: The number of user annotations required by objects of different length to achieve average error less than 5 pixel per frame. Clearly, M4 and M5 (proposed methods) requires significantly less user efforts especially for objects with longer video sequences.

cision of an algorithm can be captured accurately using the edge error. Thus, the aim is to calculate the number of user annotations required by different algorithms to get an 'Edge Error' less than 1 per frame. A better algorithm should achieve this error rate with minimum possible user annotations. Table 2.1 shows the number of user annotations required to get an edge error less than 1 per frame for different active learning algorithms. We observe that our proposed method 'Collaborative Neighborhood Tracker' (M5) consistently outperforms all other approaches. This is due to the incorporation of temporal neighborhood information into key frame sampling scheme. Notice that, 'Collaborative Tracker' (M4) which lacks neighborhood information performs better than M1, M2 and M3 confirming that our hybrid tracker is more accurate than individual trackers.

Fig 2.2 shows the behavior of proposed method (M5) in case of occlusion. The algorithm intelligently selects the key frame so as to improve the overall track. Clearly, the tracking algorithm is improving after every user annotation.

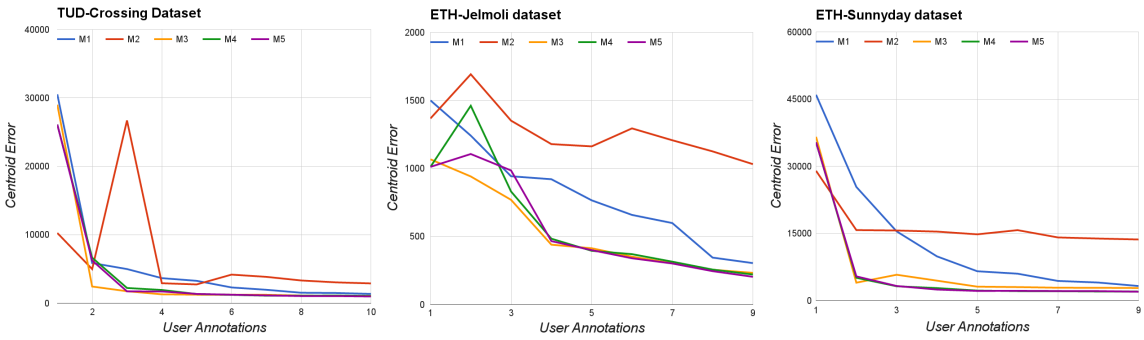
Another important measure to decide the effectiveness of any object detection framework is the 'Average Error'. The annotation algorithm having least average error after a certain number of user annotations is the better one. Table 2.2 shows the average error achieved by different annotation algorithms with same number of user interactions. Clearly, the proposed method (M5) is performing better than other annotation algorithms. We are able to achieve nearly half error with same user efforts then the other approaches.

Objects	Active Learning Approaches				
	M1	M2	M3	M4	M5
TUD-Campus(2)	12	15	10	**6**	7
TUD-Crossing(1)	7	13	5	3	**2**
TUD-Crossing(5)	10	20	15	13	**12**
TUD-Crossing(8)	16	22	19	15	**12**
ETH-Banhof(2)	12	18	8	11	**7**
ETH-Banhof(3)	12	20	7	7	**5**
ETH-Jelmoli(1)	9	14	3	**2**	**2**
ETH-Jelmoli(2)	8	12	4	**2**	**2**
ETH-Jelmoli(5)	9	17	5	6	**4**
ETH-Sunnyday(2)	9	14	6	5	**4**
ETH-Sunnyday(5)	24	20	18	16	**15**
ETH-Sunnyday(12)	8	15	9	7	**5**
ETH-Sunnyday(34)	7	10	8	7	**6**
Couple	41	28	11	9	**7**
David	12	15	11	9	**8**
Total	196	253	139	118	**98**

Table 2.1: The number of user annotations required to get an edge error less than 1 per frame. The value in the parenthesis indicates the object ID. The best results are reported in bold.

Objects	Active Learning Approaches				
	M1	M2	M3	M4	M5
TUD-Campus(2)	205	206	175	159	**157**
TUD-Crossing(1)	176	224	147	148	**105**
TUD-Crossing(5)	485	815	525	**398**	426
TUD-Crossing(8)	908	937	816	713	**701**
ETH-Banhof(2)	485	887	427	398	**381**
ETH-Banhof(3)	288	305	209	164	**128**
ETH-Jelmoli(1)	85	224	77	70	**56**
ETH-Jelmoli(2)	104	314	99	88	**80**
ETH-Jelmoli(5)	206	447	266	193	**187**
ETH-Sunnyday(2)	222	487	286	199	**184**
ETH-Sunnyday(5)	3277	1889	1756	1487	**1401**
ETH-Sunnyday(12)	418	725	300	263	**233**
ETH-Sunnyday(34)	195	400	153	171	**146**
Couple	2099	1204	1644	1140	**934**
David	3962	1742	921	1258	**880**
Total	13122	10811	7807	6852	**6005**

Table 2.2: Average error(rounded to nearest integer) achieved by different annotation algorithms after 5 user annotations. The value in the parenthesis indicates the object ID. The better algorithm should achieve less error in same number of user annotations.



(a) (b) (c)

Figure 2.6: Change in 'Centroid Error' with increasing user annotations for (a) TUD-Crossing, (b) ETH-Jelmoli and (c) ETH-Sunnyday datasets. Clearly, error for our proposed algorithms 'Collaborative Tracker' and 'Collaborative Neighborhood Tracker' is decreasing faster than other annotation algorithms.

Centroid Error measures the precision of the center of the bounding box. For every good annotation algorithm the centroid of trackers output should be as close as possible to the center of the ground truth. In this experiment we have measured the centroid error precision for different active learning strategies on three datasets namely TUD-Crossing, ETH-Jelmoli and ETH-Sunnyday. We have aggregated the centroid errors for all the object after each user annotation to check the convergence of these algorithms. Fig 2.6 shows the change in 'Centroid Error' with increasing user annotations for different datasets. Clearly, error for our proposed algorithms 'Collaborative Tracker' and 'Collaborative Neighborhood Tracker' is decreasing faster than other annotation algorithms.

Another major concern while doing large scale video annotation is the scalability of the annotation algorithm. The annotation cost increases significantly for videos with larger duration. For these experiments we have taken objects (multiple datasets) of varied length and calculated the number of user annotations required to get a satisfactory track (Average error less than 5 pixels per frame). From fig 2.5, the performance difference is high for objects that are present for larger number of frames which shows that the proposed method is effective for both short as well as long video sequences. This shows that the proposed approach is highly scalable.

Thus, from above experiments it is clear that using the proposed approach user efforts required for video annotations can be reduced to 50%. Also, the method is scalable and robust to challenges like occlusion.

2.7 Summary

In this work, we propose an efficient and accurate method to effectively annotate huge video sequences with minimal user efforts. The approach is suitable for generating large annotated datasets for mission critical applications like surveillance and autonomous driving. We effectively utilize the active learning

approach to decide the best selection of key frames. This makes our approach scalable to generate huge annotations for large scale surveillance and automotive related videos with substantial reduction in human efforts. We have verified that using the proposed approach, annotation efforts can be reduced to half while maintaining the track quality.

Chapter 3

Parallel Corpora Generation

Multilingual processing tasks like statistical machine translation and cross language information retrieval rely mainly on availability of accurate parallel corpora. Manual construction of such corpus can be extremely expensive and time consuming.

3.1 Introduction

Parallel corpus is an inevitable resource for many language processing tasks like Statistical Machine Translation(SMT) and cross-lingual information retrieval. Such tasks require an *aligned parallel corpus* where each sentence in a source language is aligned to the corresponding translated sentence(s) in target language. The task of creating a sentence aligned parallel corpus is expensive and time consuming since it involves the task of manual translation. Major sources for creating parallel corpus are Parliamentary proceedings like Europarl corpus[28], parallel sentences from web and translations of books/documents.

India is a multilingual, linguistically dense and diverse country with rich resources of information [8]. Though Monolingual corpora are available, availability of parallel corpus is very limited in quantity for language pair other than Hindi-English. Indian parliament proceedings are available only in Hindi and English and not in any other languages. But there are numerous amount of books that are translated in more than one language which are not digitized but can be used as a reliable source to generate parallel sentences. In this work, we are trying to leverage the Optical Character Recognition systems for digitizing the books in English and their respective translations in other Indian languages. For solving the problem of sentence alignment, various methods have been proposed over the past three decades like [17]. Since our data is OCR-generated data, existing algorithms failed to fetch a good level of accuracy since the text to be aligned is noisy.

To the best of our knowledge, two main algorithms have been proposed for sentence alignment in noisy data. The first work *Bleualign* [42] proposed MT based method for aligning sentences from OCR-generated parallel texts which are noisy. They used MT system to initially translate the texts and then used BLEU score[36] to calculate the sentence similarity which is the base for alignment. Following this method, [18] proposed a new scoring function that discriminates parallel and non-parallel sentences

based on the ratio of text covered by bilingual phrase-pairs from a Moses phrase table. The first approach requires an MT system with a reasonable performance [42] which in our case is only possible for Hindi-English pair. The second method needs the access to bilingual-phrase pairs where for Indian languages have only limited number of sentences in the parallel corpus to create phrase tables.

The SMT systems are very sensitive towards the quality of training data. We have not come across any work in the past that have a mechanism to detect the failures of alignment algorithm. We propose an Active Learning based solution that does validations along with text alignment. The key idea is, if an algorithm is able to detect its failures and give that to a human in the form of queries, one can significantly reduce the amount of human effort while consistently maintaining the output quality.

3.2 Challenges for Data Creation & Sentence Alignment

These days the accuracy of OCR systems are very good. But still multiple errors occur while reading text due to font style difference, picture quality of book *etc.* Additional 1-to-many beads are introduced in our corpus by sentence boundaries being mis-recognized because of OCR or tokenization errors. There are several errors added in the form of spelling mistakes. Sentence alignment is further complicated by image captions, footnotes or advertisements that are not marked as such, and consequently considered part of the running text of the article. These text fragments typically occur at different positions in the two language versions, or only in one of them. They can be very disruptive to sentence alignment algorithms if they are not correctly recognized as deletions (1-to-0 or 0-to-1 beads), since a misalignment may cause consecutive sentences to be misaligned as well.

3.3 Proposed Approach

In this work we present a simple yet efficient method to generate huge amount of reasonably accurate parallel corpus with minimal user efforts. We utilize the availability of large number of English books and their corresponding translations in other languages to build parallel corpus. Optical Character Recognition systems are used to digitize such books. We propose a robust dictionary based parallel corpus generation system for alignment of multilingual text at different levels of granularity (sentence, paragraphs, etc). We show the performance of our proposed method on a manually aligned dataset of 300 Hindi-English sentences and 100 English-Malayalam sentences.

The proposed approach is a recursive alignment algorithm to align text at multiple levels (sentence, paragraph, *etc.*). This method is a self updating validation algorithm that can predict when the alignment is done wrong. We show that the proposed framework can be used for precise alignment of multilingual sentences with minimal human effort.

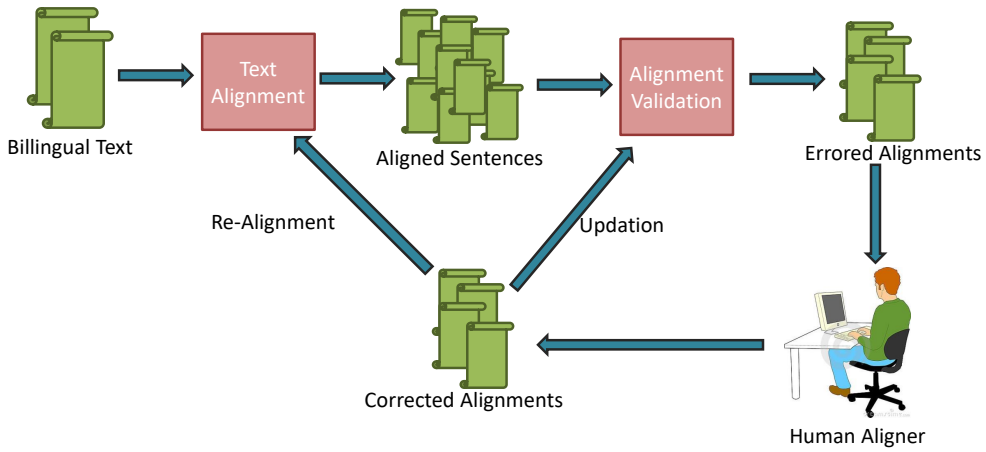


Figure 3.1: Block diagram of Align-Me framework. Given multilingual texts, an alignment algorithm is used to align the text. These aligned sentences are validated using length heuristics. Possible erroneous alignments are given to the user for corrections. These corrected alignments are used for updation of validation heuristics. In this way Align-Me aligns multilingual documents precisely with minimal user efforts.

3.4 Align Me

Align Me is an interactive framework that generates parallel corpus for two different languages given the parallel text (OCR data in our case) and a bilingual dictionary. As shown in Fig 3.1, the framework uses two separate algorithms: 'Alignment Algorithm' which align the sentences of the corpora and the 'Validation Algorithm' which detects where the former algorithm is failing. The sentences for which the alignment algorithm fails are given to the user for correction. Based on user corrections, the Validation algorithm updates itself for better prediction of the failures of the alignment algorithm.

We used the bilingual mappings released publicly by Indian Institute of Technology, Bombay [22] for the initial alignment of text. These are dictionaries that contains root words of one language mapped to all its possible translationsin the other languages. There are 242 such dictionaries containing mappings of most of the Indian languages like Assamese, Bengali, Kannada, Gujarati, *etc.* Given the OCR generated parallel text T_{l1} and T_{l2} for language L_1 and L_2, we first find out all the words of language L_1 that occur exactly once in the T_{l1}. Further, We use a dictionary D_{l1-l2} to filter out the words from W_{l1} whose corresponding mapping in L_2 has occur only once. In this way we have a set of candidate aligned words C_{aw} in T_{l1} with their corresponding words in T_{l2}.

$$c_{aw} = \{(w_{l1}, w_{l2}) \mid \quad freq(w_{l1}) = freq(w_{l2}) = 1 \quad and \quad (w_{l1}, w_{l2}) \in D_{l1-l2}\} \tag{3.1}$$

It is observed that there exist a few erroneous items in word mappings found by Eq 3.1. Thus, we added another measure to validate the former mapping technique. We assume that the displacement of a word

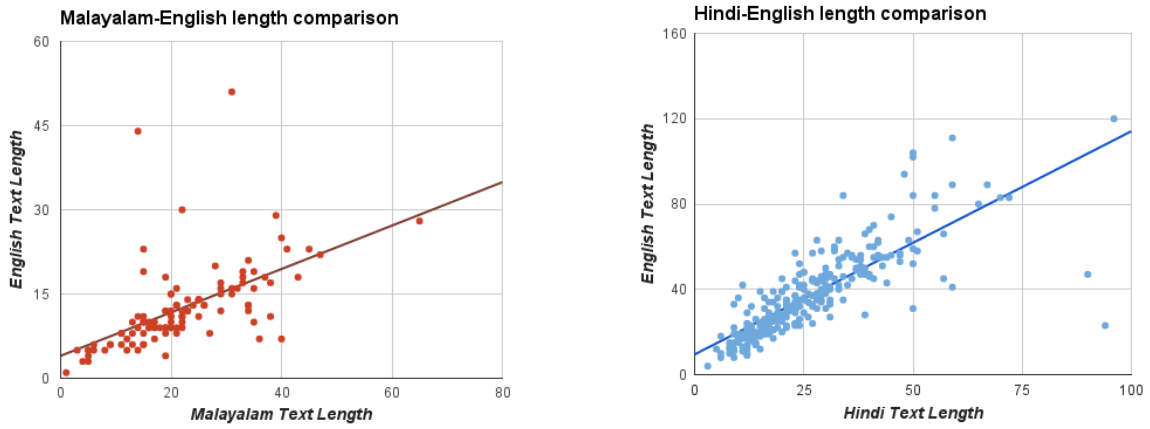


Figure 3.2: Comparison of number of words of 100 English-Malayalam sentences and 300 English-Hindi sentences. The figure shows that the count of words follow a nearly linear mapping.

and its translation should not be large. We check that the relative position of two words w_{l1} and w_{l2} in the corresponding texts T_{l1} and T_{l1} should not differ more than a threshold τ.

$$f_{aw} = \{(w_{l1}, w_{l2}) \mid (w_{l1}, w_{l2}) \in C_{aw} \quad and \quad |(pos(w_{l1})/len(T_{l1}) - pos(w_{l2})/len(T_{l2}))| < \tau\} \tag{3.2}$$

where $pos(x)$ gives the position of a word in the text and $len(y)$ gives the length of the text. We consider the final word alignments f_{aw} as the correct alignments and use them as anchors to split the text. The next division of the text is done from the next separator. We use language specific sentence separators like "|", "?", "!" in Hindi and ".", "?", "!" in English.

Fig 3.2 shows that in spite of one-to-one or many-to-one mapping between sentences of two languages, the number of words in corresponding sentences mostly follow a linear mapping. This fact is used by our validation algorithm, we train a 'Linear Regressor' for the number of words present in the corresponding aligned texts of L_1 and L_2.

$$N_2 = a + b \times N_1 \tag{3.3}$$

where, N_1 and N_2 are number of words in aligned text of L_1 and L_2. We use the above trained Regressor to predict N_2 given N_1 for all the sentences aligned by the algorithm. The sentences where predicted number of words differs from that of original number of words by a certain threshold, are given to user for correction.

After the user corrections the Regressor is updated. These aligned texts are again given to the aligning algorithm for obtaining finer alignments. After each iteration we obtain finer annotations and an updated and more accurate Regressor.

Figure 3.3: The above table shows the qualitative performance of Align-Me. The top row depicts the output of first iteration and bottom row depicts the output of second iteration. One can get aligned sentences at different levels depending on the requirement.

3.5 Experiments & Results

To create the test data we digitized four books using OCR systems namely 'George Washington Man And Monument' and its Hindi translation and Kerala assembly Budget-speech of the year 2015 and its Malayalam translation. Due to the difference in writing styles of two authors, there is a huge difference between number of sentences present in the books and their respective translations. We have tested on 492 Hindi sentences and its corresponding 356 English sentences. We have aligned them manually to get 300 English-Hindi sentences. For English-Malayalam text we have used 140 Malayalam sentences and 165 English sentences. We created 100 English-Malayalam aligned sentences to validate the performance of proposed approach.

The approaches proposed in the past used various evaluation measures. Dan [33] used block error to evaluate alignments. Chaudhary *et. al* [9] proposed a sentence based evaluation using Precision, Recall and F1-Measure. For the first level alignment of Hindi-English text we are getting 85.2% precision and 78% recall and for Malayalam-English text we are getting 96% precision and 85% recall.

To show the effectiveness of 'Active Learning' in the alignment task, we have used 'Word Level Error' than 'Sentence Level Error'. Even if a single word of a sentence have a mis-alignment, all the other words of that sentence are said to be aligned erroneously. We calculate 'Word Error Percentage' for both the languages as:

$$WordErrorPercentage = \frac{NumberofMisalignedWords}{TotalNumberofWords} \tag{3.4}$$

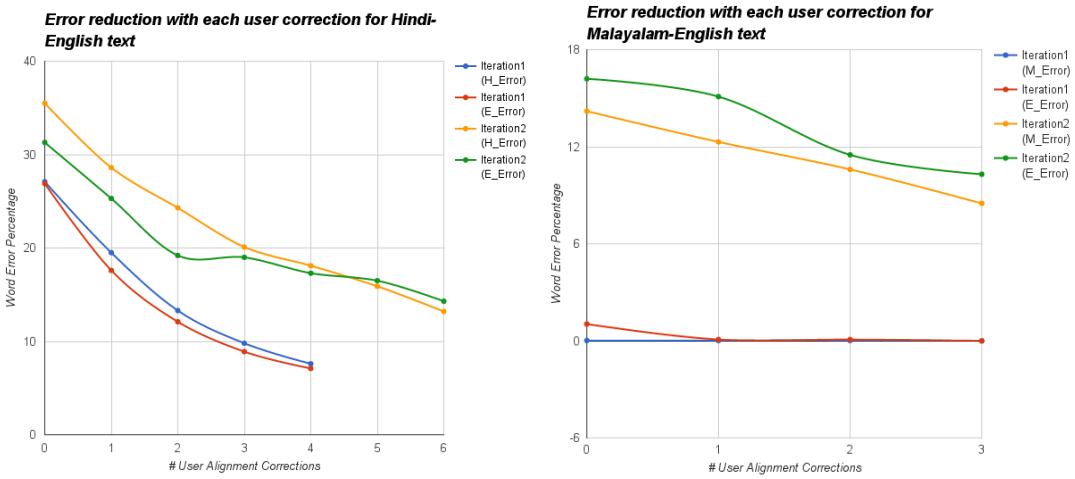


Figure 3.4: The above graph shows the reduction of 'Word Error Percentage' with every user annotation. We have calculated word errors for all the languages. 'H_Error', 'E_Error' and 'M_Error' are word errors for Hindi, English and Malayalam respectively. The error graph shows the fall of error for two iterations. It is evident that the validation algorithm is able to correctly determine the mis-aligned samples.

In Fig 3.4 we show that our algorithm is able to detect correctly, the mis-aligned texts to be queried to the user. The figure shows the reduction in error with every user correction for two iterations on same text.

Fig 3.3 shows that Align-Me is effectively able to detect aligned texts of different modularities. With each iteration finer alignments are done. We also show that the proposed framework is immune to OCR system introduced errors. In the second iteration of Malayalam alignment, the algorithm handled 1-to-many beads introduced due to mis-recognition of sentence boundaries by OCR systems.

3.6 Summary

In this work, we proposed Align Me as an efficient framework for generating huge corpus of parallel text using minimal user efforts. Our framework uses multilingual dictionaries to align the texts initially. At every step, the verification of the alignments is done using a validation algorithm which uses length based heuristics to determine possible mis-alignments. Experimental data depicts that length based heuristics work really well in cases where there are possible errors in the text-to-be-aligned. These heuristics also perform exceedingly well in cases when the number of sentences in both the languages vary by a huge count. In this approach, the human effort is reduced to a great extent as the framework queries only the misaligned sentences to the human annotator. The proposed approach can be utilized for generation of huge corpus for languages like Malayalam-English, Marathi-English, Hindi-Kannada *etc.* where there is huge paucity of aligned data. The performance of the method remains consistent even if the input data is noisy; this proves the high degree of robustness that the method offers.

As part of future work, we would like to use the proposed framework for generation of parallel corpus for other Indian languages as well. We are also trying to incorporate other factors like BLEU score for detection of mis-alignments.

Chapter 4

cQA Model Updation

The interaction between knowledge-seekers around the world is made easy by the Community Question Answering(cQA) platforms like Yahoo! Answers, Baidu Zhidao, Quora, StackOverflow *etc.* Experts provide precise and targeted answers to any question posted by a user. These sites form huge repositories of information in the form of questions and answers. Retrieval of semantically relevant questions and answers from cQA forums have been an important research area for the past few years. Several models have been built in the past to bridge the "lexico-syntactic" gap in the cQA content. However, considering the ever growing nature of the data in cQA forums, these models cannot be kept stagnant. They need to be continuously updated so that they can adapt to the changing patterns of Questions-Answers with time. Such updation procedures are expensive and time consuming.

4.1 Introduction

Community Question Answering (cQA) services provide users a platform to interact with "experts" on various topics of interest. The precise answers provided by experts in cQA system reduces the effort of the users to surf through several web pages. Users can also post opinion based questions which other users/experts can answer based on their personal experiences.

There are two major issues associated with these forums. Firstly, the users often ask repetitive questions due to which the experts end up wasting time in answering similar questions. Secondly, few questions remain unanswered for a long time ("Question Starvation") making the users wait indefinitely for an answer. The retrieval of semantically relevant questions and answers effectively solves the above stated problems. Semantic relatedness is easier to detect in Questions-Answers (QA) when they share common words and have similar structural arrangement. However, the challenge is to determine those pairs that differ syntactically and lexically but express the same meaning. For example: "What is the secret of happiness?" and "Contentment is the path to follow for a happy life." vary a lot structurally but possess semantic relation.

Several techniques have been proposed in the literature for similar QA retrieval in cQA forums. Approaches such as classical IR based retrieval models [39], Language Modeling for Information Retrieval

(LMIR) [57], Translation Models [23, 54, 63], Topic Models [6, 24, 60], Deep Learning Approaches [11, 38, 65] *etc* have proven themselves successful. However, cQA is a very active community with hundreds of QA pouring in daily. The continuous addition of data would demand frequent updation of the previous model. It is computationally very expensive to train a new model with the entire dump of data or update an existing model with thousands of new pairs each time. In our proposed method, we leverage Active Learning to make the task of model updation computationally efficient.

The key idea behind Active Learning is to make a machine learning algorithm achieve greater accuracy with fewer training samples.Active Learning based solutions typically prove to be helpful in domains where there is a scarcity of annotated data. These solutions focus on intelligent selection of a smaller set of unlabeled instances which if labeled allow the learning algorithm to learn faster. For example, addition of questions "What are good places to hangout in New York?", "What are the tourist attractions in New York?" and their respective answers would not be of much help to a cQA model as both pairs are semantically similar. Thus, we use Active Learning to select an optimal set of QA pairs from the newly added data that is sufficient to update the model.

4.2 Related Work

Semantic relatedness is not captured efficiently by the classic retrieval models, such as BM25 [39], LMIR [57], as they focus on finding textual similarities between the queries. Researchers have also utilized translation based models for solving this problem. Xue *et al.* [54] added the query likelihood language model to improve the performance of the word based translation model. Jeon *et al.* [23] estimated the translation probabilities by utilizing the similar nature of the archived answers. Zhou *et al.*[63] used phrase based translations considering the QA pairs as parallel corpus. However, Zhang *et al.* [60] used lexical and user behavior heterogeneity to point out that considering the QA pair as parallel is not an effective method.

Later, topic modeling [6, 24, 60] assumed that there are few common hidden topics that the questions and answers share. These techniques matched questions on both topic and term levels.

Zhang *et al.* [59] learnt the word representations and question categories together and integrated the learnt representations into traditional language models. Zhou *et al.* [64] used Fisher Kernel to create fixed size representations of questions with varied lengths. The model embedded the questions with the metadata "category".

Following the trend of deep learning based solutions, Das *et al.* [11] worked on retrieval of similar questions utilizing a deep structured topic model. They combined paired Convolutional Networks and Topic Models to retrieve semantically related questions. Qui *et al.* [38] introduced Convolutional Neural Tensor Network (CNTN), which combined semantic matching and sentence modeling. This model changed the word tokens into vectors with the help of a look-up layer and encoded the questions and answers to fixed-length vectors with convolution and pooling layers. Zhou *et al.* [65] used a deep neural network to map the QA pairs to a common semantic space. Semantic similarity between questions

and answers is determined by cosine similarity of their semantic vectors. Das *et al.* [12] proposed Siamese Convolutional Neural Network (SCQA) which uses shared parameters to learn the similarity metric between QA pairs.Most of these works focus on creating a model for finding the semantic similarities between Question-Answer and Question-Question pairs, but none of these works demonstrate techniques to keep the model up-to-date.

4.3 Proposed Solution

We propose a novel Topic model based active sampler named *Picky*. It intelligently selects a smaller subset of the newly added Question-Answer pairs to be fed to the existing model for updating it. Evaluations on real life cQA datasets show that our approach converges at a faster rate, giving comparable performance to other baseline sampling strategies updated with data of ten times the size.

In this work we propose a Topic Model based Active Sampler called "Picky" which serves as a query selection strategy for updating cQA models. We build a base model with a significantly small amount of entire data and keep on updating it with the newly arriving data. We use the base model to predict the scores of the new QA pairs. The pairs are added to a candidate set if the model is not confident, *i.e.*, the model cannot predict them as positive pairs with high confidence. We apply topic modeling on the candidate set to represent each QA pair as a distribution of topics. We cluster the candidate set based on topics and select pairs from each topic cluster depending on the density of that topic. These QA pairs are treated as the final set of data points needed to update the cQA model. Topic modeling ensures that an even distribution of new data points are taken from each topic cluster. It also avoids selection of outliers making the updating model stable and robust.

The major contribution of our work are as follows:

1. We propose an intuitive query selection strategy for stable updation of models in cQA with freshly arriving data.

2. Our approach intelligently selects datapoints to be fed to the model which saves training time. It can also be used for reducing annotation expenses by reducing the data needed for re-training.

3. We overcome the drawback of uncertainty sampling (selection of outliers) using density based method. Model confidence is used as a measure to find uncertainty and topic clusters are used for density estimations.

4. Results from real-life cQA dataset validates that our approach converges at a faster rate with much lesser amount of data.

Active Learning also known as "query learning" [43] relies on the assumption that if the learning algorithm is allowed to choose the data from which it learns, it will perform better with less training data. Active Learning is predominantly used in areas where getting a large amount of annotated data for training is not feasible or extremely expensive. For example, accurate labeling of speech utterances

is extremely time consuming and requires trained linguists. Information extraction systems require extensive training on labeled documents having detailed annotations. Active Learning models aims to overcome the annotation bottleneck by asking queries in the form of unlabeled instances to be labeled by a human. The framework aims to achieve high accuracy using very less labeled instances resulting in minimization of annotation cost.

Over years people have tried different query selection strategies like *Uncertainty Sampling* [2, 30], *Density Sampling* [16, 53], *Query-By-Committee* [4, 47], *Expected Model Change* [44], *Expected Error Reduction* [14, 40], *etc* to calculate the informativeness of unlabeled instances in an Active Learning scenario. Poursabzi *et al.* introduced Active Learning with Topic Overviews (ALTO) [37], which used density sampling to help humans annotate documents.

The data in cQA systems is ever-increasing, repeatedly training a model on such incremental data is computationally expensive. The key hypothesis of this work is to utilize Active Learning systems to select the most informative points from the newly added data, thereby significantly reducing the amount of data needed to feed the model to keep it up-to-date.

"*Picky*" uses a combination of Uncertainty and Density sampling to select the data points needed to update the model. The pairs for which the model prediction score is very less are treated as "Uncertain Pairs". Though these Uncertain Pairs contain information about changing data distribution, it also contains a significant amount of random and promotional content. To filter erroneous data, the Uncertain Pairs are divided into clusters of different topics. More samples are selected from dense topics and less samples from sparse topics (Density Sampling). Combination of Uncertainty and Density sampling ensures selection of QA pairs depicting the change in data distribution and avoidance of outliers.

4.4 *Picky* - A Topic Model based Active Sampler

Picky as shown in Fig 4.1 combines uncertainty and density based sampling to select the most informative data points. *Picky* is a generic technique which can be applied in any domain irrespective of the data set and algorithm used to build the learning model.

In this paper, we validate this sampling technique to retrieve semantically similar questions and answers in cQA. In cQA systems, QA pairs generally belong to different categories like sports, lifestyle, family and relationships *etc.* These high level categories have multiple subcategories. Each subcategory contains a set of related questions-answers and the number of questions per subcategory is small. It is possible that a newly added question may belong to an unseen subcategory. Thus, only a subset of the categories is known during the time of training. The aim of the proposed approach is to find out samples which are likely to belong to such unseen categories generating the need for model updation. *Picky* is an intuitive approach to keep updating these models with unseen data with minimal computational cost and effort.

We use the deep semantic model CDSSM [45] as the training algorithm to learn the similarities between questions and answers. CDSSM [45] maps raw textual features into vectors in the semantic space.

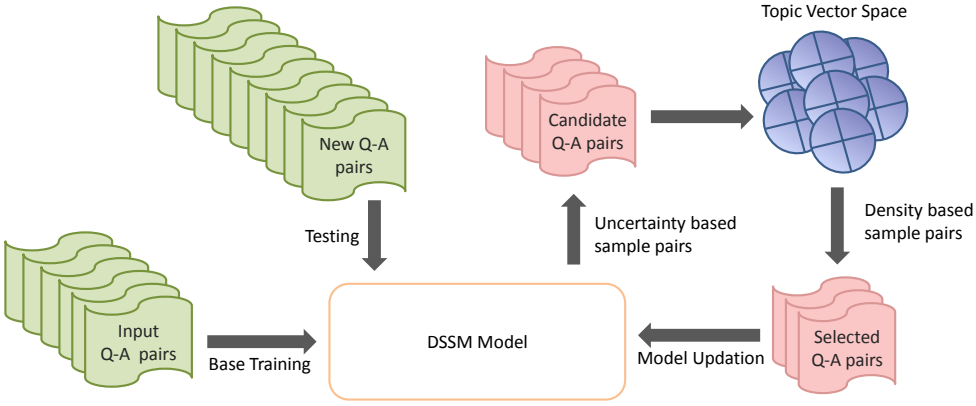


Figure 4.1: Block diagram of *Picky*. Initially, CDSSM model is trained with input QA pairs. Later with every iteration the previous model is used to test the new pairs. Based on model uncertainty, candidate set is created which is projected to a topic vector space using LDA. Final query pairs are selected from this space using Density sampling.

These models work with clickthrough data consisting of query and the set of clicked documents. The input to these networks are not high dimensional term vectors of document but letter tri-gram based vectors which reduces dimensionality and take care of out-of-vocabulary words and spelling errors. For example, "pen" is represented as (#pen#) where # is the delimiter used, letter 3-grams would be #pe, pen, en#. CDSSM It uses Convolutional Neural Network which consider words at a contextual level and project each word within a context to a local contextual feature vector. On the local feature vector, it uses a max pooling layer to extract the crucial local features to form a fixed-length global feature vector. Affine transformations and element-wise non-linear functions are then applied to the global feature vectors to extract highly non-linear and effective features.

A base CDSSM is trained initially with available QA pairs with the objective to maximize the conditional likelihood of the answers given the questions or to minimize the loss function in equation 4.1.

$$L(\Lambda) = -log \prod_{(Q,A^+)} P(A^+|Q) \tag{4.1}$$

where Λ denotes the set of parameters of CDSSM, Q is the question, A^+ is the set of answers given for the question(best answer or answers with few user votes). For negative training pairs we feed CDSSM with A^- which contain random answers from other questions.

However, the base model gets outdated with time due to the the changing distribution of data. It needs to be updated using the newly added QA pairs. *Picky* uses the base CDSSM model's uncertainty as an indicator of the need for model updation. For each batch of newly added QA pairs, we test them

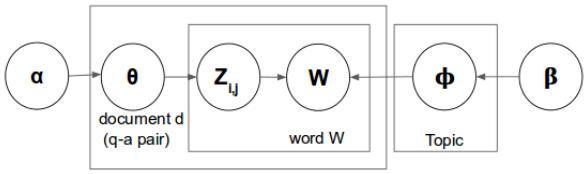


Figure 4.2: Graphical model of generative LDA.

using the available trained model. We formulate a candidate set QS_c for model updation, which contain all the pairs for which the model score is less than a threshold, depicting the CDSSM model's inability to predict that the pairs in QS_c are related.

The pairs in QS_c are uncertain but feeding the entire set for updating the model is not necessary. The idea is to pick less but more informative pairs which when sent to update the model, will make the model quickly adapt and learn the patterns in newly arriving data. To form the final updation set, we focus on the entire candidate topical space rather than individual instances of candidate set. This will ensure that the final selection is less prone to outliers. The least certain instances according to model are not "representative" of other instances in the distribution. Updating the model with only these instances is unlikely to improve accuracy on the complete data. We overcome this problem by modeling the candidate set distribution explicitly during data point selection. The main idea is that informative instances should not only be those which are uncertain, but also those which are "representative" of the unseen categories.

We use Latent Dirichlet Allocation (LDA) [3] as a topic modeling approach to learn the distribution of candidate set QS_c. LDA transform the question answer pairs which the base model cannot predict confidently, to latent topic space. It is used to discover latent semantic structure in the collection of uncertain QA pairs. The latent semantic structure aims to push the words with similar semantics into the same topical space. Therefore, the topics can capture more semantic information than raw term features. LDA is a generative probabilistic topic modeling approach which is fed with the uncertain pairs generated from the CDSSM model that includes hidden variables as topics. In LDA, the latent variables are represented as polynomial distribution over words in the dataset. Therefore, the QA pairs in QS_c could also be represented as a polynomial distribution over topics.

The observed variables are the bag of words per QA pair and the hidden random variables are the topic distribution per pair, the distribution of the vocabulary per topic and the topic for a word in a particular QA pair. The graphical model of LDA can be seen in figure 4.2. QA pairs in QS_c are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. The generative process of LDA used to find the posterior probability of the hidden variables given the value of observed variables is explained in [11].

The posterior probability is used to obtain the topic vector distribution of each QA pair. QS_c can be projected into K dimensional topic vector space. The output of topic modeling is a $M \times K$ sparse matrix T which determines the probability of each candidate pair belonging to one or more than one topics.

$$T = [P(a_{i,j})]_{M*K} \tag{4.2}$$

where $P(a_{i,j})$ is the probability of the i^{th} candidate QA pair to belong to j^{th} topic.

We use $P(a_{i,j})$ to determine the informativeness of the candidate pair. For each topic we select candidates which are most probable to lie in the topic space, thus capturing the semantics of the topic. We select N_j number of candidates from a j^{th} topic by using the density of that topic. A denser topic will add more candidates to the final updation set. Thus, for a j^{th} topic:

$$N_j \propto |T_{(1,..M),j}| \tag{4.3}$$

The proportionality constant of Equation 4.3 is determined by the total number of pairs we want to add to the final set. The number of topic K in which QS_c is divided is determined by the size of QS_c. The final set is an optimal representation of the possible new categories added in each batch. As, the size of final set is very less in comparison to the total added pairs, there is a huge reduction in time as well as computational complexity of updating.

4.5 Experiments and Discussion

In this section, we first describe the dataset and evaluation measures used in our experiments. We also demonstrate the effectiveness of the proposed algorithm using various experiments.

4.5.1 Dataset Details

We collected Yahoo! Answers dataset from Yahoo! Labs Webscope. The dataset has about 4 million questions. Each question in the dataset contains title, description, best answer, most voted answers and meta-data like categories, sub categories etc. Each question-answer pair was pre-processed by lower-casing, stemming, stop-word and special character removal. We trained the base model training using randomly selected 1 million question-answer pairs. We consider 2 million data as the new incoming data. We divided these 2 million samples into 4 equal sizes.

We used the remaining 1 million data as unbiased test data to evaluate the performance on different models. We use Precision, Recall and F1-Score as evaluation measures.

4.5.2 Results and Discussion

The experiments we conducted on Yahoo! Dataset show that *Picky* exhibits good performance with reduced computational cost and expenses. *Picky* can be used in any challenge where frequent model updation is required. In the first experiment, we trained the Base CDSSM Model with 1 million samples. We added 0.5 million randomly selected samples with each iteration and updated the model using this

Training Model & Train-Set Size	Precision	Recall	F1-Score
CDSSM (Base Model) (1.0M)	0.811	0.582	0.678
Base + Random (1 + 0.5M)	0.701	0.727	**0.713**
Base + *Picky* (1 + 0.05M)	0.691	0.733	0.712
Base + Random (1.5 + 0.5M)	0.874	0.612	0.719
Base + *Picky* (1.05 + 0.05M)	0.892	0.652	**0.753**
Base + Random (2.0 + 0.5M)	0.672	0.764	0.715
Base + *Picky* (1.1 + 0.05M)	0.685	0.751	**0.716**
Base + Random (2.5 + 0.5M)	0.659	0.727	0.691
Base + *Picky* (1.15 + 0.05M)	0.628	0.799	**0.703**

Table 4.1: The table compares performance of CDSSM base model with addition of random data points and *Picky* selected data points. The base model is trained with 1 million training pairs. For $Base + Random$ models, we keep on updating the previously trained model with randomly selected 0.5 million samples with each iteration. For $Base + Picky$ models, we use 0.05 million informative pairs to update the the previously trained model iteratively. With $Picky$ we are able to achieve comparable performance with one-tenth of data. Threshold used for determination of positive model prediction is 0.5. The results of better performing algorithm are boldfaced.

additional data. Let us call the new model as $Base + Random_{addition}$. We, apply *Picky* on these 0.5 million samples and extract 0.05 million informative pairs from it and use them to update the base model. Let us call the new model as $Base + Picky_{addition}$. For this experimental setting, we divided QC_s into 200 Topics. We continue the same method for 4 iterations and keep on updating the previous model with the random and *Picky* extracted samples. To compare the performances we have used an unbiased test set of 1 million positive pairs and 3 million negative QA pairs. Table 4.1 shows that in each iteration the performance of $Base + Random_{addition}$ and $Base + Picky_{addition}$ are comparable in spite of the huge difference in number of samples used for updating the model. This proves that *Picky* is correctly sampling the data points that the model needs to be updated with.

We observed that the performance measures are not following any specific pattern. This is due to that fact that Yahoo! Answers contain a lot of noisy and promotional content embedded within the data. One more important observation is that the Yahoo! Answers dump we have used for proving the effectiveness of *Picky*, is shuffled in nature. It is not a time-series data. The use of shuffled data for training and updating minimizes the probability of addition of new data pairs belonging from very different sub-categories. For a dataset arranged according to timeline, the questions coming in 2016 would be much different from that of 2005. If the base model is trained with database till 2010 and

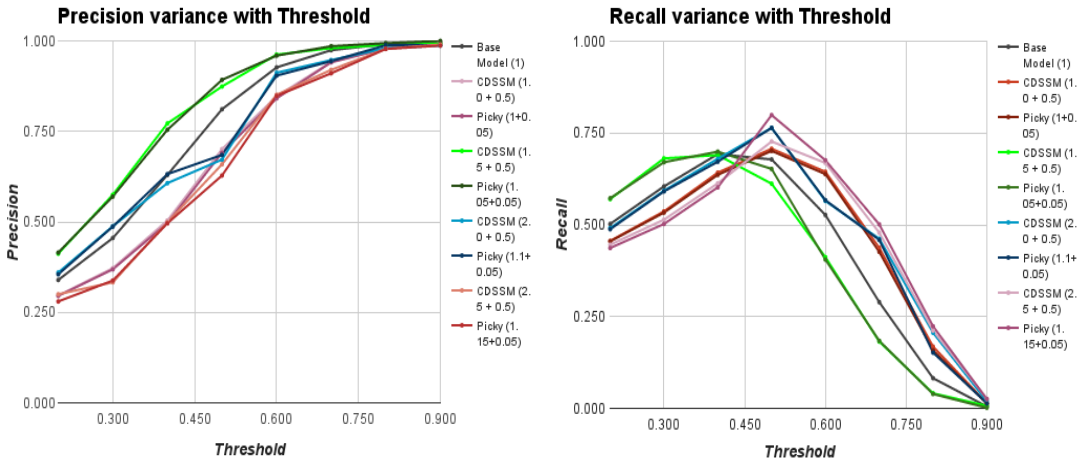


Figure 4.3: Change of Precision and Recall with increase in threshold for different models trained using Random Samples and *Picky* based samples.

for updating we would add new data from 2011-2016, the model would learn much more new sub-categories and changes in pattern of incoming data. The testing on this new model would surely reflect huge improvement in results. Though, the present results are depicting the effectiveness of the data point selection strategy of *Picky*, we would obtain better performance if we would have used time-stamped un-shuffled data.

We also compared the performance of the models with *Random* selection and *Picky* based selection of data points for different values of threshold. The model outputs a confidence score that denote whether the QA pair is semantically similar in nature. We keep a threshold on the score to evaluate the performance of the models. As shown in Figure 4.3, with the increase in threshold, the precision increases and the recall first increases then starts decreasing. Therefore, we have chosen the threshold as 0.5 to keep a balance between precision and recall values. The figure depicts that the models updated with *Picky* samples are consistently outperforming the models updated with *Random* samples for all values of threshold.

We also conducted an experiment to prove that *Picky* selects more informative data points than random selection. We use a base CDSSM model trained on 1 million QA pairs. We then randomly select 0.2 million QA pairs from remaining 2 million samples and update the base model with this randomly selected 1.2 million data. We separately update another model by adding 0.2 million pairs actively selected using *Picky* to base model. As it can be observed from Figure 4.4, the Active selection of data points show better performance than random selection of same number of data points.

Table 4.2 compares the effectiveness of *Picky* over other sampling baselines. We use a base CDSSM model trained on 1 million training pairs. In each iteration we add 0.05 million QA pairs using *Picky*, Uncertainty Sampling and Random Sampling. We show that the proposed framework is able to distinctly outperform the other sampling strategies. While analyzing the results we found out that there are many QA pairs containing unreliable and promotional data. The base model predicts such pairs uncertainly by

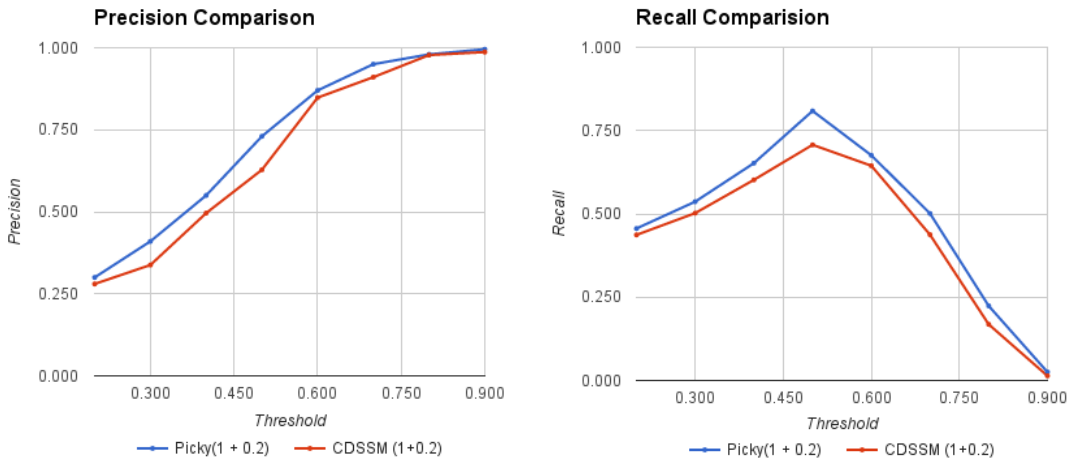


Figure 4.4: Comparison of informativeness of equal number points selected by *Picky* and Random Sampling.

giving them low semantic similarity score. In Uncertainty Sampling these erroneous pairs are added for retraining resulting in drastic decrease in performance. *Picky* considers such pairs as outliers, therefore model updation is not affected much by such data. Table 4.2 also shows that the results does not vary much for Random Sampling with the addition of new samples . This is because the base model is trained with a huge number of QA pairs and very less samples are added in each iteration. Random selection of such samples does not capture the change in the distribution of data thereby showing no increase in the performance of the model.

4.6 Summary

We proposed Topic Modeling based Active Sampler *Picky* as an efficient framework for updating a model used for retrieval of semantically relevant questions and answers in cQA systems. *Picky* employs the fact that QA pairs can be meaningfully represented in a topic vector space. It utilizes this distribution to shortlist the data pairs which are smaller in number but informative enough to represent an entire topic space. The shortlisted data points can update the model on behalf of other points in that topic space. Experiments on large scale real-life "Yahoo! Answers" dataset revealed that *Picky* shows comparable performance to the models updated with data of 10 times the size.

The approach proposed is re-usable and scalable in nature. It can be used for model updation for any amount of data over a infinite period of time. Since this approach help in extracting a few representative points out a large set of points, it significantly reduces the updation time and resources. Also, for supervised tasks where annotated data is scarce, *Picky* saves a lot of human effort and expense by choosing a small subset of informative data to be explicitly annotated.

As part of future work, we would like to enhance the sampling model by also utilizing the meta-data information like categories, user votes, ratings, user reputation of the questions and answer pairs for query selections. Also, we would like to experiment with other deep neural architectures such as

Train-Set Size	*Picky*		Uncertain		Random	
	Precision	Recall	Precision	Recall	Precision	Recall
1.05 M	**0.691**	**0.701**	0.583	0.492	0.651	0.659
1.10 M	**0.874**	**0.652**	0.552	0.448	0.663	0.641
1.15 M	**0.672**	**0.764**	0.598	0.486	0.648	0.603
1.20 M	**0.659**	**0.799**	0.614	0.551	0.646	0.695
1.25 M	0.650	**0.831**	0.631	0.503	**0.672**	0.645

Table 4.2: The table compares the effectiveness of *Picky* over other sampling baselines. We use a base CDSSM model trained on 1 million training pairs. In each iteration we add 0.05 million QA pairs using *Picky*, Uncertainty Sampling and Random Additions. We show that the proposed framework is able to distinctly outperform the other sampling strategies. Threshold used for determination of positive model prediction is 0.5. The results of better performing algorithm are boldfaced.

Recurrent Neural Networks, Long Short Term Memory Networks, *etc.* to form a robust and reusable framework.

Related Publications

Conference

1. Priyam Bakliwal, Guruprasad M. Hegde, C. V. Jawahar. **Collaborative Contributions for Better Annotations.** The 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP, VISIGRAPP), 2017.

2. Priyam Bakliwal, Devdath V V, C. V. Jawahar. **Align Me : A Framework to Generate Parallel Corpus Using OCRs & Bilingual Dictionaries.** The 26th International Conference on Computational Linguistics(COLING), 2016.

3. Priyam Bakliwal, C. V. Jawahar. **Active Learning Based Image Annotation.** The Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2015.

Bibliography

[1] Y. Angela, G. Juergen, L. Christian, and G. Luc, Van. Interactive object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 12

[2] P. Bakliwal and C. Jawahar. Active learning based image annotation. In *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, 2015. 35

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 2003. 37

[4] M. Bloodgood and K. Vijay-Shanker. A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. In *ACL Conference on Computational Natural Language Learning*, 2009. 35

[5] a. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 14

[6] L. Cai, G. Zhou, K. Liu, and J. Zhao. Learning the latent topics for question retrieval in community qa. In *International Joint Conference on Natural Language Processing*, 2011. 33

[7] M. Chatterjee and A. Leuski. CRMActive: An active learning based approach for effective Video annotation and retrieval. *International Conference on Multimedia Retrieval*, 2015. 12

[8] S. Chaudhury, D. M. Sharma, and A. P. Kulkarni. Enhancing effectiveness of sentence alignment in parallel corpora: Using mt heuristics. 2008. 25

[9] S. Chaudhury, D. M. Sharma, and A. P. Kulkarni. Enhancing effectiveness of sentence alignment in parallel corpora: Using mt heuristics. *ICON*, 2008. 29

[10] M. Danelljan, G. Haumlger, F. Shahbaz Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. *The British Machine Vision Conference*, 2014. 13, 14

[11] A. Das, M. Shrivastava, and M. Chinnakotla. Mirror on the wall: Finding similar questions with deep structured topic modeling. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2016. 33, 37

[12] A. Das, H. Yenala, M. Shrivastava, and M. Chinnakotla. Together we stand: Siamese networks for similar question retrieval. *Association for Computational Linguistics*, 2016. 34

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 11

[14] P. Donmez, J. G. Carbonell, and P. N. Bennett. Dual strategy active learning. In *European Conference on Machine Learning*. Springer, 2007. 35

[15] R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. *NIPS*, 2009. 11

[16] L. Fu and R. Grishman. An efficient active learning framework for new relation types. In *International Joint Conference on Natural Language Processing*, 2013. 35

[17] W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 1993. 25

[18] L. Gomes. First steps towards coverage-based document alignment. 2016. 25

[19] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. *PETSW*, 2007. 10

[20] M. Grubinger. Analysis and evaluation of visual information systems performance. 2007. 5

[21] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann. Inter-active learning of ad-hoc classifiers for video visual analytics. *VAST*, 2012. 12

[22] IIT. Bilingual mappings (http://www.cfilt.iitb.ac.in/downloads.html). Bombay. 27

[23] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *ACM International Conference on Information and Knowledge Management*. ACM, 2005. 33

[24] Z. Ji, F. Xu, B. Wang, and B. He. Question-answer topic model for question retrieval in community question answering. In *ACM International Conference on Information and Knowledge Management*. ACM, 2012. 33

[25] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Breaking the interactive bottleneck in multi-class classification with active selection and binary feedback. *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 11

[26] M. M. Kalayeh, H. Idrees, and M. Shah. Nmf-knn: Image annotation using weighted multi-view non-negative matrix factorization. 2014. 1

[27] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato. A semi-automatic tool for detection and tracking ground truth generation in videos. *VIGTAW*, 2012. 10

[28] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, 2005. 25

[29] Lee, Y. Jae, and K. Grauman. Learning the easy things first: Self-paced visual category discovery. *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 11

[30] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer, 1994. 35

[31] J. V. M. Guillaumin, T. Mensink and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. 2009. 1, 4, 6, 7

[32] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *European conference on computer vision*. Springer, 2008. 1, 2, 4, 6, 7

[33] I. D. Melamed. A geometric approach to mapping bitext correspondence. *arXiv preprint cmp-lg/9609009*, 1996. 29

[34] S. Oh and et. al. A large-scale benchmark dataset for event recognition in surveillance video. *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 11

[35] N. d. F. P. Duygulu, K. Barnard and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. 2004. 5

[36] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002. 25

[37] F. Poursabzi-Sangdeh, J. Boyd-Graber, L. Findlater, and K. Seppi. Alto: Active learning with topic overviews for speeding label induction and document labeling. In *Association for Computational Linguistics*, 2016. 35

[38] X. Qiu and X. Huang. Convolutional neural tensor network architecture for community-based question answering. In *International Joint Conference on Artificial Intelligence*, 2015. 33

[39] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al. Okapi at trec-3. *NIST Special Publication SP*, 1995. 32, 33

[40] N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *International Conference on Machine Learning*, 2001. 35

[41] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 2008. 11

[42] R. Sennrich and M. Volk. Mt-based sentence alignment for ocr-generated parallel texts. 2010. 25, 26

[43] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010. 34

[44] B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, 2008. 35

[45] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. Learning semantic representations using convolutional neural networks for web search. In *ACM International Conference on World Wide Web*, 2014. 35

[46] D. Thomas, A. Bogdan, and V. Ferrari. Localizing objects while learning their appearance. *ECCV*, 2010. 10, 11

[47] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2001. 35

[48] Y. Verma and C. V. Jawahar. Image annotation using metric learning in semantic neighborhoods. 2012. 1, 2, 4, 6, 7

[49] Y. Verma and C. V. Jawahar. Exploring svm for image annotation in presence of confusing labels. 2013. 1, 6

[50] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 2013. 11

[51] C. Vondrick and D. Ramanan. Video annotation and tracking with active learning. *NIPS*, 2011. 12

[52] M. Wang and X. Hua. Active learning in multimedia annotation and retrieval: A survey. 2011. 2

[53] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *European Conference on Information Retrieval*. Springer, 2007. 35

[54] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. In *ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2008. 33

[55] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations, 2009. 15, 19

[56] Z. J. Zha, M. Wang, Y. T. Zheng, Y. Yang, R. Hong, and T. S. Chua. Interactive video indexing with statistical active learning. *Transactions on Multimedia*, 2012. 12

[57] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 2004. 33

[58] K. Zhang and H. Song. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognition*, 2013. 13

[59] K. Zhang, W. Wu, F. Wang, M. Zhou, and Z. Li. Learning distributed representations of data in community question answering for question retrieval. In *ACM International Conference on Web Search and Data Mining*, 2016. 33

[60] K. Zhang, W. Wu, H. Wu, Z. Li, and M. Zhou. Question retrieval with high quality answers in community question answering. In *ACM International Conference on Conference on Information and Knowledge Management*, 2014. 33

[61] D. Zhong and S.-F. Chang. Structure analysis of sports video using domain models. *International Conference on Multimedia and Expo.*, 2001. 10

[62] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. *IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 10

[63] G. Zhou, L. Cai, J. Zhao, and K. Liu. Phrase-based translation model for question retrieval in community question answer archives. In *Association for Computational Linguistics: Human Language Technologies*, 2011. 33

[64] G. Zhou, T. He, J. Zhao, and P. Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In *Association for Computational Linguistics*, 2015. 33

[65] G. Zhou, Y. Zhou, T. He, and W. Wu. Learning semantic representation with neural networks for community question answering retrieval. *Knowledge-Based Systems*, 2016. 33

[66] H. Zhou, Y. Yuan, and C. Shi. Object tracking using sift features and mean shift. *Computer vision and image understanding*, 2009. 10