DOMAIN SPECIFIC SEARCH BY RANKING MODEL ADAPTATION USING BINARY CLASSIFIER

N. A. Anuja Jaishree¹, M. Krishnamurthy²

¹PG Scholar Department of CSE, KCG College of Technology, Chennai, Tamilnadu, India.
²Professor Department of CSE, KCG College of Technology, Chennai, Tamilnadu, India.

ABSTRACT

Domain-specific search focus on one area of knowledge. Applying broad based ranking algorithm to vertical search domains is not desirable. Broad based ranking model is built upon the data from multiple domains. Vertical search engines use a focused crawler that attempts to index only relevant web pages to a pre-defined topic. With Ranking Adaptation Model we can adapt an existing ranking model to a new domain. Binary classifiers classify the members of a given set of objects into two groups on the basis of whether they have some property or not.

Keywords: Ranking Adaptation SVM, Information Retrieval, Support Vector Machines, Learning to Rank, Domain Adaptation.

I. INTRODUCTION

Information retrieval is the process of getting information, relevant to the information needed, from a collection of information sources from the given data available. Searches are based on full text or other content-based indexing. Web search engines are the most visible IR applications. An information retrieval process is started from when a user enters a keyword or phrase or query in the search engine. Queries are words or statements that describe the information needed by the user. Information retrieval does not return a single object result from a database. A set of objects or results are returned to the user based on their level of relevancy. This relevancy in IR systems is calculated by a numeric score that defines how good a object in the collection matches for the query given by the user according to the value. The objects which are in the top ranking is then displayed to the user. This process will continue till the user gets the most desirable results.

In machine learning, Support Vector Machines are supervised learning models with associated learning algorithms that analyze the given data and recognize any interesting pattern
which is then used for classification and regression analysis of the data. The basic procedure of a SVM is that it takes a set of data as input and for each of the input predicts the two possible class of output it will belong to in a non probabilistic binary linear classifier method. From a given set of training examples where the data is marked to belong to either of the two categories SVM builds a training algorithm for constructing a new model which assigns new data into any of the category of the training set.

Learning to rank or machine-learned ranking (MLR) is a type of supervised or semi supervised machine learning problem which automatically constructs a ranking model from the training data. Training data is the data which is in some level of order between the objects in the given list. This order is calculated by a numeric or ordinal score or a binary judgment e.g. “relevant” or “not relevant” for each item.

In pairwise approach learning to rank problem is solved by a classification problem learning a binary classifier that determines which document is good in a given pair of documents. Example method: Ranking SVM: Ranking using click through logs i.e. search results which got clicks from users.

Ranking SVM is a typical method of learning to rank. There are two factors one must consider when applying Ranking SVM a learning to rank method to document retrieval. Ranking documents which appear in top of the results is very important for Information Retrieval. The training must be done such that the results are accurate. The number of relevant documents varies based on the query. The training model should be such a way that it should not be biased towards a query.

Three Important Processes in ‘Ranker’
- Retrieval–Finding documents from inverted index
- Matching–Calculating relevance score between query and document pair
- Ranking–Ranking documents based on relevance scores, importance scores, etc.

There is limited amount of labeled data and there are so many domains. It is often the case that plentiful labeled data exists in one domain (or coming from one distribution), but one desires a model that performs well on another related, but not identical domain. Hand labeling data in the new domain is a costly enterprise, and one often desires to be able to leverage the true, “external domain” data when a model is built for the new, “internal domain” data. There is need not to eliminate the annotation of in-domain data, but instead seek to minimize the amount of new annotation effort required to achieve better performance results. This is known as domain adaptation.

A vertical search engine is different from a common web search engine that it focuses on a particular domain of online content. They are even called specialty or topical search engines. The vertical content area is based on topicality, media type of the item or the genre of content. Some verticals are online shopping, electronic items, medicine, literature, science, tourism. Indeed.com and Yelp are two examples of vertical search engine. General search engines indexes large collection of data in the World Wide Web using a web crawler, however vertical search engines uses a focused domain crawler that indexes only the Web pages which are matching to a pre defined domain.

There are vertical search sites for individual as well as multiple vertical searches within a search engine. Vertical search provides benefits in broad based domains for ranking while the domain specific features terms of greater accuracy due to limited domain scope specific domain knowledge with taxonomy and ontology and support of specific unique user tasks.

Domain-specific verticals focus on a specific topic. Domain-specific search solutions focus on one area of knowledge, creating customized search experience, that because of the domain's limited corpus and clear relationships between the concepts, provide well relevant results for searchers using the search engine.
The objective is of this project is:
• To adapt ranking models learned for the existing broad-based search or some verticals, to a new domain, so that the amount of labeled data in the target domain is reduced while the performance requirement is still guaranteed;
• The ranking model to be effectively and efficiently adapted;
• Boost the model adaptation by domain specific features.

II. RELATED WORK

The broad based ranking model is built on the data obtained from multiple domains from the whole web and thus cannot be generalized for a specific domain with users special search requirements. The broad based ranking model only utilizes the vertical domains ranking features that are similar to the such as content specification of image, video and music cannot be utilized direct. Broad based ranking model provides a priori knowledge with Auxiliary Ranking Model. Size of auxiliary domain training set is large and complex. The broad-based and vertical search engines are generally based on text search mechanisms. The ranking model learned for broad based can be implemented to use directly to rank the documents for the verticals.

As many vertical search engines emerge and the amount of verticals increases drastically, a global ranking model, which is trained over a data set obtained from multiple domains, will not give a precise performance for each specific domain with special ontology, document formats, and domain-specific features [1]. Building one model for each vertical domain is both laborious for labeling the data and time consuming for learning the model. In this project the ranking model adaptation is proposed, to adapt the well learned models from the broad-based search or any other auxiliary domains to a new target domain. In model adaptation, only a small number of samples need to be labeled, and the computational cost for the training process is greatly reduced.

The statistical learning theory [8] learning algorithms are used for SVM which is a learning framework for machine language. The SVM learning algorithm efficiently optimizes the MAP [9] is used for calculating the precision of a document to a given adaptation query. The rank of the page is calculated using the click through data for optimizing the search engine [5]. Bias correction method [11] is useful for classifier evaluation under sample selection bias. Page Rank citation algorithm [7] determines the back links to a page and thus its popularity for a user. Applying learning to rank to document retrieval is discussed in adapting ranking SVM [3]. A novel application of P-R curves and average precision computations reveal interesting phenomena of IR evaluation methods [4].

III. RANKING MODEL ADAPTATION USING DOMAIN SPECIFIC SEARCH

The first objective is solved by proposing a ranking adaptability measure, that quantitatively calculates whether an existing ranking model can be adapted to the new target domain, and determines the level of performance for the adaptation.

The second problem is addressed by the regularization framework and a ranking adaptation SVM (RA-SVM) algorithm is proposed. The algorithm is a black box ranking adaptation model, which takes just the predictions from the existing ranking model for the trained data, rather than the internal representation of the model itself or the data from the auxiliary domains. The black-box adaptation property, achieves flexibility and efficiency. To resolve the third problem, assume that documents similar in their domain-specific feature space should have consistent in rankings. e.g., images should be similar to the visual feature space should be ranked into similar positions and vice versa. This is implemented by constraining the margin and slack variables of RA-SVM adaptively, so that similar documents are assigned with less ranking loss if they are ranked in a wrong order.
The proposed RA-SVM can better utilize both the auxiliary models and target domain labeled queries to learn a more robust ranking model for the target domain data. The utilization of domain-specific features can steadily boost the model adaptation. RA-SVM-SR is comparatively more robust than RASVM-MR. The adaptability measurement is consistent to the utility of the auxiliary model, and thus can be taken as an effective measure for the auxiliary model selection. The proposed RA-SVM is as efficient as directly learning a model in a target domain, however the including of domain-specific features doesn’t bring much learning difficulties in algorithms RASVM-SR and RA-SVM-MR.

IV. DESIGN

The ranking model adaptation starts with the input of the search query which compares with the database for the ranking adaptation domain ontology. From here the match type is seen and the requested web search done which gives the relevant search result as shown in Fig. 1.

The detailed proposed architecture for the ranking model describing ranking model adaptation domain ontology has the broad based ranking model to ranking adaptation SVM which gives the domain specific ranking model using the binary classifier in Fig. 2.
V. DATABASE DESIGN

The database designed for this project for the search engine and the members who can use the search engine. Database contains the repository of all the web pages and the detailed description of the traffic of the web pages. The database which is created is the repository from which the data which is relevant is retrieved according to the query given by the user.

There are two databases which acts as the repository for ranking of the web pages and one for the members

A. Visitors and Rank Table

In this table the keyword that is being recognized in the database is given as query. The name of the website and the link to the relevant query is given. Then the database through RASVM calculates the visitors to the page by the number of click through and the rank of a page is determined proportionately to the click in Table 1.

<table>
<thead>
<tr>
<th>QUERY/KEY WORD</th>
<th>SITE NAME</th>
<th>LINK</th>
<th>#VISITORS</th>
<th>RANK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antivirus</td>
<td>Norton</td>
<td><a href="http://www.norton.com">www.norton.com</a></td>
<td>23</td>
<td>5</td>
</tr>
<tr>
<td>Bank</td>
<td>SBI</td>
<td><a href="http://www.yahoo.com">www.yahoo.com</a></td>
<td>45</td>
<td>3</td>
</tr>
<tr>
<td>Games</td>
<td>Free Games</td>
<td><a href="http://www.bored.com">www.bored.com</a></td>
<td>75</td>
<td>2</td>
</tr>
</tbody>
</table>
B. Registered Members Table

Registered members table gives the user details for their login to use the search engine with their date of registration as shown in Table 2.

<table>
<thead>
<tr>
<th>ID</th>
<th>FIRST NAME</th>
<th>LAST NAME</th>
<th>EMAIL</th>
<th>USER NAME</th>
<th>PASS WORD</th>
<th>REG DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Riya</td>
<td>Sen</td>
<td><a href="mailto:rya@gmail.com">rya@gmail.com</a></td>
<td>Riya</td>
<td>l23rrtg3</td>
<td>12-01-14</td>
</tr>
<tr>
<td>2</td>
<td>Jeni</td>
<td>Kawa</td>
<td><a href="mailto:jeni@yahoo.com">jeni@yahoo.com</a></td>
<td>Jeni</td>
<td>Abtfg6</td>
<td>14-01-14</td>
</tr>
<tr>
<td>3</td>
<td>Marasi</td>
<td>lyer</td>
<td><a href="mailto:manasi@gmail.com">manasi@gmail.com</a></td>
<td>Marasi</td>
<td>Op2341</td>
<td>20-01-14</td>
</tr>
</tbody>
</table>

VI. RANKING ADAPTATION

Ranking Adaptation implements two ranking methods of SVM.

A. R-SVM

Ranking SVM one of the pair-wise ranking methods, that can adaptively sort the web-pages by their relationships how relevant to a specific query or keyword given by the user. A mapping function is required to define such relationship. The mapping function projects every data pair enquired and clicked web-page onto a feature space. The user’s click-through data combine with the features will denote page ranks for a specific query is considered as the training data for machine learning algorithms.

Ranking Method

Suppose C is a data set containing C elements C_i. r is a ranking method applied to C. Then the r in C can be represented as a C by C asymmetric binary matrix. If the rank of C_i is higher than the rank of C_j, i.e. r.C_i < r.C_j, the corresponding position of this matrix is set to value of "1". Otherwise the element in that position will be set as the value "0".

Information retrieval relevancy is measured by the following three measurements:

1. Precision
2. Recall
3. Average Precision

For a specific query to a database, let P_relevant be the set of relevant information elements in the database and P_retrieved be the set of the retrieved information objects. The above three measurements is represented as follows:

\[
Precision = \frac{|P_{relevant} \cap P_{retrieved}|}{|P_{retrieved}|}
\]

\[
Recall = \frac{|P_{relevant} \cap P_{retrieved}|}{|P_{relevant}|}
\]
B. Ranking Adaptation –SVM

The auxiliary domain and the target domain are related, their respective ranking functions should have similar shapes in the function space. Under such an assumption, auxiliary function actually provides a prior knowledge for the distribution of new function in its parameter space. RA-SVM is equivalent to directly learning Ranking SVM over the target domain, without the adaptation of auxiliary function. RA-SVM minimizes the distance between the target ranking function and the auxiliary function. RA-SVM is equivalent to learning only target domain.

Kendall’s Tau

Kendall’s Tau is the Kendall tau rank correlation coefficient, which is used to compare two ranking methods for the same data set. Suppose r1 and r2 are two ranking method applied to data set C, the Kendall’s Tau between r1 and r2 can be represented as follows:

\[
\tau(r1,r2) = \frac{P - Q}{P + Q} = 1 - \frac{2Q}{P + Q}
\]

Where P is the number of the same elements in the upper triangular parts of matrices of r1 and r2, Q is the number of the different elements in the upper triangular parts of matrices of r1 and r2. The diagonals of the matrices are not included in the upper triangular part.

VII. OPTIMIZED METHODS

The optimized methods check the relevancy of a page to the given query.

A. Precision@K
   - Set a rank threshold K.
   - Compute % relevant in top K.
   - Does not consider documents ranked lesser than K.

B. Mean Average Precision
   - Consider rank position of each relevance doc K_1, K_2, … K_R
   - Compute the calculation formula Precision@K in every K_1, K_2, … K_R documents.
   - Average precision = Average of P@K
   - MAP is Mean Average Precision across multiple queries/rankings in the given collection.

VIII. RANKING ADAPTATION WITH DOMAIN SPECIFIC FEATURE

This section specifies the consistent ranking feature for similar documents.

A. Margin Rescaling
   Margin rescaling determines rescaling the margin violation adaptively according to their similarities in the domain-specific feature space.
In a pair of dissimilar documents, similar ones with larger similarity will result in a smaller margin to satisfy the linear constraint, which produces less ranking loss in terms of a smaller slack variable similarity if the document pairs are ranked in a wrong order by the function f.

B. Slack Rescaling

Slack rescaling varies the amplitude of slack variables in the documents adaptively.

Slack rescaling is intended to rescale the slack variables according to their similarities in the domain specific feature space. When a pair of documents are dissimilar in the domain-specific feature space, by dividing the dissimilarity, the slack variables that control the ranking loss of the two documents are correspondingly amplified in order to satisfy the first linear equality of the pair of documents, and vice versa.

IX. RESULT ANALYSIS

Experiments performed over the datasets crawled from the search engine data base demonstrate the proposed ranking adaptation in the following graphs and charts.

To analyze the efficiency of the proposed RA-SVM-based methods, the learning time of different methods is compared by varying the adaptation query number in the webpage search engine settings. The Aux-Only does not take time learning a new ranking model, and the Tar-Only needs to be trained beforehand and then linearly combine it with Aux-Only, the comparison of Tar-Only, RA-SVM, RA-SVM-MR, and RA-SVM-SR with respect to Mean Average Precision (MAP) is taken.

The results are shown in Fig. 3, Fig. 4 and Fig. 5, it is observed that for small number of adaptation query number, the time of results returning of different algorithms are very similar. In large adaptation sets, even though Tar-Only is little better than RA-SVM methods, the standard deviation of different methods are not very significant.

This can be concluded that the proposed RA-SVM is quite efficient compared with direct training a model in the target domain. The RA-SVM-MR and RA-SVM-SR results show that the inclusion of domain-specific features doesn’t bring further learning complexity.

A. Mean Average Precision Graph

The following table in Table 3 shows the tabular data for interrelationship of MAP to Ranking Models. The Mean Average Precision of document to the query is calculated. The result is calculated based on 2, 4, 6 adaptation query number and how relevant the document was returned using Precision and Recall.

<table>
<thead>
<tr>
<th>QUERY</th>
<th>Tar-Only</th>
<th>Aux-Only</th>
<th>RA-SVM</th>
<th>RA-SVM-SR</th>
<th>RA-SVM-MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 QUERY</td>
<td>0.43</td>
<td>0.425</td>
<td>0.45</td>
<td>0.45</td>
<td>0.45</td>
</tr>
<tr>
<td>4 QUERY</td>
<td>0.431</td>
<td>0.425</td>
<td>0.452</td>
<td>0.448</td>
<td>0.454</td>
</tr>
<tr>
<td>6 QUERY</td>
<td>0.433</td>
<td>0.425</td>
<td>0.455</td>
<td>0.455</td>
<td>0.456</td>
</tr>
</tbody>
</table>

The graph in Fig. 3 shows the inter relationships between the Mean Average Precision Score and the various ranking methods.
The RA-SVM-MR gives more Mean Average Precision and hence a more relevant search query result.

B. Ranking Methods Comparison

The following tabular data represents the ranking methods comparison in Table 4.

<table>
<thead>
<tr>
<th>QUERY</th>
<th>Tar-Only</th>
<th>Aux-Only</th>
<th>RA-SVM</th>
<th>RA-SVM-SR</th>
<th>RA-SVM-MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 QUERY</td>
<td>1.5</td>
<td>1.1</td>
<td>1.6</td>
<td>1.5</td>
<td>1.7</td>
</tr>
<tr>
<td>4 QUERY</td>
<td>3</td>
<td>2.9</td>
<td>3.5</td>
<td>3.5</td>
<td>3.6</td>
</tr>
<tr>
<td>6 QUERY</td>
<td>4.5</td>
<td>4.2</td>
<td>5.5</td>
<td>5.4</td>
<td>5.7</td>
</tr>
</tbody>
</table>

The bar graph in Fig. 4 shows the comparisons of all the ranking methods:

RA-SVM –MR is quite efficient compared to other ranking methods.
C. Click through Vs Rank

The tabular representation in Table 5 shows click through of pages to ranking of a page.

<table>
<thead>
<tr>
<th>QUERY</th>
<th>Click through</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 QUERY</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>4 QUERY</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>6 QUERY</td>
<td>7</td>
<td>3</td>
</tr>
</tbody>
</table>

The graph in Fig. 5 shows the proportionality of the click through to pages to rank of the page.

![Click through vs. Rank](image)

The more the click through to a webpage increases by the user the more does it is ranked for relevancy of the page.

Thus the above analysis and results shows that RA-SVM-MR is a consistent ranking adaptation model for domain specific search.

X. CONCLUSION

In this project with the regularization framework based a Ranking Adaptation SVM algorithm is implemented, which performs adaptation in a black-box method where just the relevant predication of the auxiliary ranking models is needed for the adaptation rather than learning the ranking model itself.

Based on RA-SVM, two variations called RA-SVM margin rescaling and RA-SVM slack rescaling are proposed to utilize the domain specific features to further improve the ranking adaptation, by taking that similar documents should have consistent rankings, and restricting the margin and loss of RA-SVM adaptively according to their similarities in the domain-specific feature space.

Thus in this project ranking model adaptation for domain specific search using binary classifier is implemented.

The future work is the ranking adaptation of sponsored ads for search engines. Sponsored ads will be ranked according to the given domain vertical of a query and also return the top ranking ads based on relevancy and study the algorithms based on it.
REFERENCES


