INTRODUCTION:- Recommender systems help individuals and communities address the challenges of information overload. Inform a tion filtering recommenders look at the syntactic and s e mantic content of items to determine which are likely to be of interest or value to a user.

Each of these technologies has a role in producing an e f fective recommender system. Information retrieval (IR) systems focus on allowing users to express queries to select items that match a topic of interest to fulfill a particular information need.

Information filtering(IF) systems require a profile of user needs or preferences. The simplest systems require the user to create this profile manually or with limited assistance.informatio filtering techniques have a central role in recommender systems. IF techniques build a profile of user preferences that is particularly valuable when a user encounters new content that has not been rated before.

Collaborative filtering (CF) systems build a database of user opinions of available items. They use the database to find users whose opinions are similar (i.e., those that are highly correlated) and make predictions of user opinion on an item by combining the opinions of other likeminded individuals.

Hybrid Recommender Systems Several systems have tried to combine information filte r ing and collaborative filtering techniques in an effort to overcome the limitations of each.

Data Set: The user ratings for this experiment were drawn from the MovieLens system (http://movielens.umn.edu) which has more than 3 million ratings from over 80,000 users. Fifty users were selected at random from the set of users with more than 120 movie ratings. For each user, three sets of movies/ratings were selected at random without replac e ment. The first set of 50 ratings, termed the training set , was set aside for use in training the personalized inform a tion filtering agents. The second set of 50 ratings, termed the correlation set was used when combining users, agents, or both together. The final set of 20 ratings served as the test set .

Metrices:Recommender systems researchers use several different measures for the quality of recommendations produced. Coverage metrics evaluate the number of items for which the system could provide recommendations. In many systems, coverage decreases as a function of acc u racy—the system can produce fewer accurate recomme n dations or more inaccurate ones.

Experimental Components Our hypotheses are based on four models of recommender system: • user opinions only, • individual IF agents, • combinations of IF agents, and • combinations of IF agents and user opinions.

User Opinions Only. Extensive research has already been performed on the problem of generating recomme n dations from a set of user opinions. Nearestneighbor collaborative filtering is already generally accepted to be the most effective mechanism for performing this task, and we therefore use it (Breese, 1998). In particular, we use the DBLens research collaborative filtering engine developed by the GroupLens Research project for expl o ration of collaborative filtering algorithms.

Individual IF Agents. Three types of IF agents, or filte r bots, were created and studied in this project: DGBots, RipperBot, and a set of GenreBots. Doppelganger Bots (DGBots) are personalized bots that create profiles of user preferences and generate predi c tions using IR/IF techniques (specifically, a modified TFIDF, (Salton and Buckley 1987) based upon the co n tent features of each movie.

Ripper bot was created using Ripper, an inductive logic program created by William Cohen (Cohen, 1995). We found that Ripper performed best when trained on a set of data limited to genre.

identifiers and the 200 most frequent keywords. Ripper also works best when asked to make binary decisions, so for each user we trained four Ripper instances, tuned to distinguish between 5/4321, 54/321, 543/21, and 5432/1 respectively.

The GenreBots consisted of 19 simple bots that rated each movie a 5 if the movie matched the bot's genre and a 3 otherwise.

Combination of Users and IF Agents. Because user ratings were incomplete, and because CF with 23 agents proved to be the most effective combination of IF agents, we used CF to combine the 23 agents and all 50 users. The method is identical to the CF combination of agents except that we also loaded the ratings for the other 49 users. Again, the database was cleared after each user.

Results H1: Collaborative Filtering better than Single Agents We hypothesized that collaborative filtering using the opinions of the 50-user community would provide better results than any individual agent. To compare these, we first identified the best individual agent. We evaluated the three DGBots, RipperBot, the 19 individual genr e Bots, and the personalized Mega-GenreBot (see table 1). Of these, only RipperBot, Mega-GenreBot, and the DGBot that used both cast and keywords were not dominated by other agents. RipperBot had the highest accuracy (lowest MAE) by far, but low ROC sensitivity (poor decision support).

H2:Many agents better than one.

H3:CF of users better than combination of agents.

H4:Combinations of agents and users is best overall.

DISCUSSION:-

The most important results we found were the value of combining agents with CF and of combining agents and users with CF. In essence, these results suggest that an effective mechanism for producing high-quality reco m mendations is to throw in any available data and allow the CF engine to sort out which information is useful to each user. In effect, it becomes less important to invent a brilliant agent , instead we can simply invent a collection of useful ones.