Stock Recommendation Application

Naveen Jetty, Narendra Ambati, Sarath Chandra Bandaru, Suresh Yarra
Computer Science
University of Missouri – Kansas City

I. Abstract

The Stock market data is very large and it is very interesting in analyzing the type of data being generated. Every market data keeps changing. Tracking the data and comparing with other market is bit tricky. Our Stock Recommendation Application provides the stock data from past five days for the selected company. We also recommend the other companies against the selected company with its data. The basic search is provided for retrieving the results using Natural Language Processing. The Text you entered in analyzed in Natural language and the results are shown. We also started to use TF and TF/IDF for finding company stock related in some articles. This is yet to be integrated in our Stock Recommendation System.

This is a web application which recommends stocks with Highest similarities to clients' favorites. Client view the Real-time info of a specific stock and select their best-likes through drop down buttons and search bar. The System crawls NASDAQ stock info from Yahoo Finance in a real-time manner and calculates similarities using Algorithm. The system recommends 5 stocks for every client favorite and presents the information of these stocks in a table in UI.

II. Introduction

The stock market, based on real time transactions, is a new and important component of the present day financial system. Recently, interest on share trades in the stock exchange market is growing tremendously and expectations point to an even faster growth. Considering the scenario that the move of stock market plays an important role for companies to obtain investments, take strategic decisions and expand its activities on a global level, many specialists are motivated to take advantage of the observed behavior of shares to help avoid instability within a company, or further, to predict its behavior in the near future. Though it is possible to use the historical data of stock market to do prediction for investment, accurate predictions are still very hard to make due to the disorganized nature of stock markets, which might depend on external events like governmental actions, company crisis and many other factors.

Our project suggested a way that, instead of trying to predict one particular stock's movement with respect to the whole market, it might be easier to predict a stock A's movement based on stock B's movement, which turned out to be the main motivation of our stock recommendation system. Based on a customer's specific choices, our system is able to recommend a number of similar stocks (both short-term results and long-term results are provided). Besides, stock trend figures and real-time stock value are all available on our system.

The rest of the report is structured as follows. Section III describes the system structure of our project. Section IV shows in detail how Algorithm works in our calculation of similarity. Section V gives overview of the software package used in this system. Section VI shows the experimental results obtained by our recommendation and Section VII presents the conclusions and references.

III. System Overview

System Structure

We designed a stock recommendation system which recommends stocks with highest similarities to users based on their favorites. The system uses a J2EE framework, it has one UI page, several serve lets, a recommendation algorithm and a mysql database. Figure 1 shows an overview of project structure. The project components are explained in detail in the following:

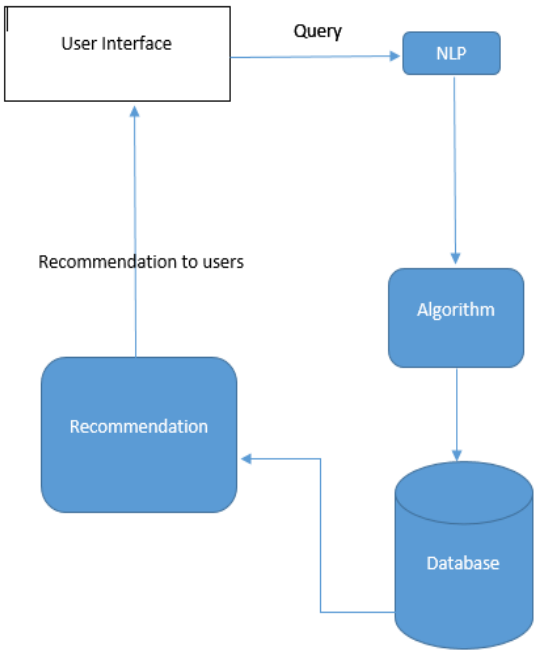


Figure-1: System Architecture

UI and Servlets:

The UI helps clients choose their favorites stocks through drop-down buttons and search bar. They can view the real-time trend of every stock by clicking the plot button. The drop down buttons helps them make a multiple selections based on that suits their requirement. After client makes their own choices, the UI will send client favorites' to the back-end and present recommendation results from back-end in a table. Our UI is written in html and is decorated with CSS and Bootstrap. It has a group of JavaScript functions to interact with clients in a dynamic manner. Servlets connect UI and java code block of the project. It passes the input from UI to a java program and hands out the return data from java program to the UI. The project contains four servlets, which is used to calcite similarity, calculate distance between two stocks and extract stock information from Yahoo Finance.

NLP

The user query first processed by NLP, we are using Stanford coreNLP technology, SentimentAnnotator technique we followed. It gives high positive, positive, neutral, negative and high negative five values for each word and gives whether the total sentence is either positive or negative. We are taking that result from NLP and returning the results based on that. The result from SentimentAnnotator we are using in the algorithm.

TF: Term Frequency

TF is one of the information extraction technique. It extracts the information from a given set data. For this application we are implementing a search engine for a repository. For this, first we are running a vector functions to generate the corpus data. Once the corpus data is generated we calling that data in the index form. Building the repository by tf and tf/idf methods. Once the repository is built user can gives a query to search. We are also displaying the time taken to display the results.

Yahoo Finance API:

We crawls Nasdaq stock information from the Yahoo Finance API. The URL of this API is "http://finance.yahoo.com/d/quotes.csv?s=". We select bid, 50 day moving average, day high and day low as our attributes and store these information's into our database.

Java blocks:

The system has two java packages, which are: data model and similarity calculation model. The data models are the classes we used to store and process the stock information locally. For each client's best-like stock, the system builds up a CustomerStock object to store its symbol and a list which contains 5 recommended stocks with the highest similarities to it. Every recommended stock is stored in the recommendation list as a RecStock object. It contains fields such as its similarity to clients' favorites, its bid, moving average, day high and day low.

The similarity calculation model are the java programs we used to calculate similarity.

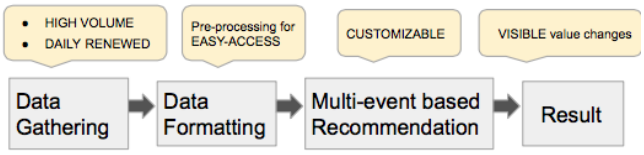


Figure 2: Flow map

Dataset

We use NASDAQ Stock Exchange Data as our dataset. It contains 2959 stocks in total. As mentioned above, we use Yahoo Finance dataset to acquire historical prices of each stock. The dataset is daily renewed with most recent value via Yahoo API. We choose MYSQL as our database because it is the most popular open source database and it is most deployed database in the web and cloud (with 80% Deployments integrated)

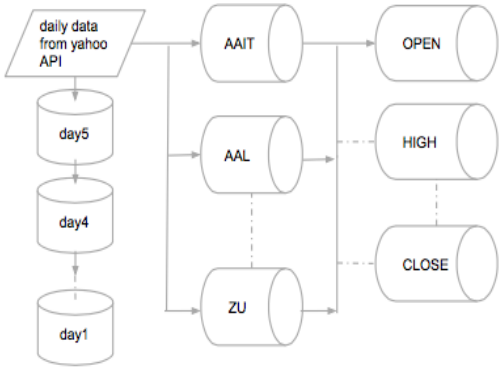


Figure 3: Dataset

IV. ALGORITHM

We created a model for maintaining stock information. Here the information implies of concurrent data. Concurrent data is like a sequence of data in specific time period (here time window is 5days). By using this concurrent information we can define stock move in a time window. We are doing similarity calculation by considering single event. Single event is like at a specific time (here on single day). Here on a single day we can predict negative/positive relation, which is extracted from Stanford CoreNLP.

Similarity calculation is done based on the calculation of distance between the bid values. These calculations involve the concurrent bid values and finding the average of them. These distance values fall in the range of 0 and 1. The lowest distance valued company is considered to be closely related to the company selected.

V. SOFTWARE PACKAGE DESCRIPTION

UI overview

The User interface of our stock recommendation system contains our project title, main page figure, three dropdown buttons and recommendation search bar. The functions of these components are explained in detail in the following.

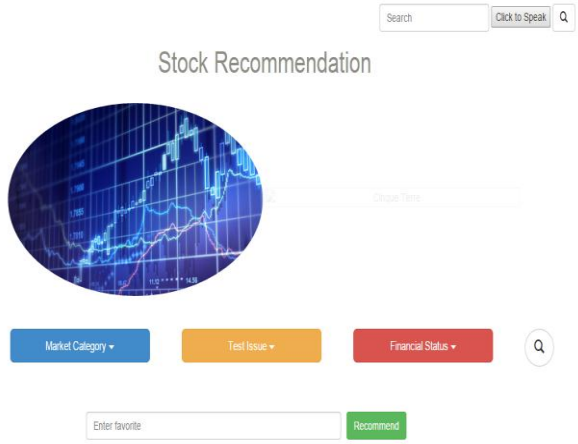


Figure 4: UI Overview

Drop down button

The first step of our recommendation process is to let the client choose his favorite stocks. Clients can make selections through the three drop-down buttons, which are market category, test issue and finical status. When client chooses different options, the content of the related search bar will also change. On the right side of the drop-down buttons there is a round search button, a table is extracted from our database which presents all the stocks that fit in the prerequisites.

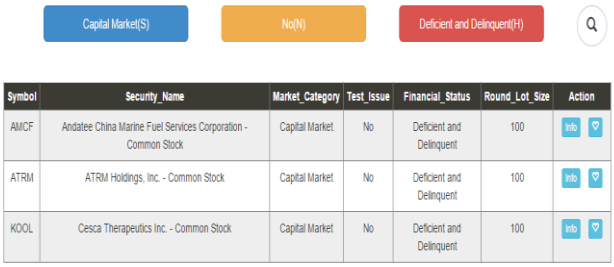


Figure 5: Drop Down Buttons

Recommendation

Suppose that the client has chosen two stocks as his favorites', after he clicking the recommend button, the system send the information to the back-end, which will calculate the similarities of all the stocks from Nasdaq stock market to the two stocks and presents the results in a table. The system recommends 5 stocks with highest similarities for every choice. The table also presents the general information for the recommended stocks such as similarity, today's bid, moving average, day high and day low.

Search Button

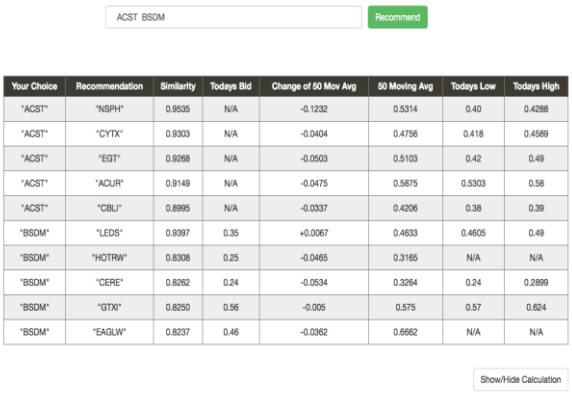
The search button at the top is used to search for a user query in natural language format. If you type 'The least selling stock' in the search bar, the query undergoes through NLP's SentimentAnnotator and from its result the stock details are displayed.

Figure 6: Search Query Results

VI. EXPERIMENT RESULTS

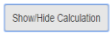
We are now able to use rather complex episodes for our stock market evaluation. Since it is hard for us to get access to commercial databases, we use stock information provided by yahoo finance to show that our similarity evaluation is reasonable.

We are showing the results for ACST and BSDM stocks, we recommending 5 recommendation stocks for the these two companies. We are also showing how we calculated the similarities.



Your Choice	Recommendation	Similarity	Todays Bid	Change of 50 Mov Avg	50 Moving Avg	Todays Low	Todays High
"ACST"	"NSPH"	0.9535	N/A	-0.1232	0.5314	0.40	0.4288
"ACST"	"CYTX"	0.9303	N/A	-0.0404	0.4756	0.418	0.4589
"ACST"	"EGT"	0.9268	N/A	-0.0503	0.5103	0.42	0.49
"ACST"	"ACUR"	0.9149	N/A	-0.0475	0.5875	0.5303	0.58
"ACST"	"CBLI"	0.8995	N/A	-0.0337	0.4206	0.38	0.39
"BSDM"	"LEDS"	0.9397	0.35	+0.0067	0.4633	0.4605	0.49
"BSDM"	"HOTRW"	0.8308	0.25	-0.0465	0.3165	N/A	N/A
"BSDM"	"CERE"	0.8262	0.24	-0.0534	0.3264	0.24	0.2899
"BSDM"	"GTXI"	0.8250	0.56	-0.005	0.575	0.57	0.624
"BSDM"	"EAGLW"	0.8237	0.46	-0.0362	0.6662	N/A	N/A

Figure 7: Recommendation Results for ACST & BSDM



Your Choice	Recommendation	Similarity	Bid Distance	Change of Mov Avg Distance	50 Moving Avg Distance	Daily change	Distance
"ACST"	"NSPH"	0.9620	0.0198	0.0277	0.0064	0.0189	0.0395
"ACST"	"CYTX"	0.9328	0.0020	0.0389	0.0406	0.0450	0.0720
"ACST"	"IRDMZ"	0.9271	0.0500	0.0530	0.0297	0.0000	0.0787
"ACST"	"BSDM"	0.9238	0.0250	0.0327	0.0592	0.0400	0.0825
"ACST"	"EGT"	0.9232	0.0200	0.0071	0.0086	0.0800	0.0832
"BSDM"	"IRDMZ"	0.9582	0.0250	0.0203	0.0295	0.0000	0.0437
"BSDM"	"CYTX"	0.9486	0.0230	0.0062	0.0186	0.0450	0.0542
"BSDM"	"CBLI"	0.9393	0.0229	0.0325	0.0475	0.0186	0.0647
"BSDM"	"BGMD"	0.9365	0.0152	0.0493	0.0267	0.0350	0.0678
"BSDM"	"ACST"	0.9276	0.0250	0.0327	0.0592	0.0300	0.0781

Figure 8: Calculation Results

From the above figure we can that the recommendation results for these two companies are accurate based on our dataset.

Apart from this we mentioned earlier about TF/IDF we implemented a search engine for the repository. Those results are shown below.

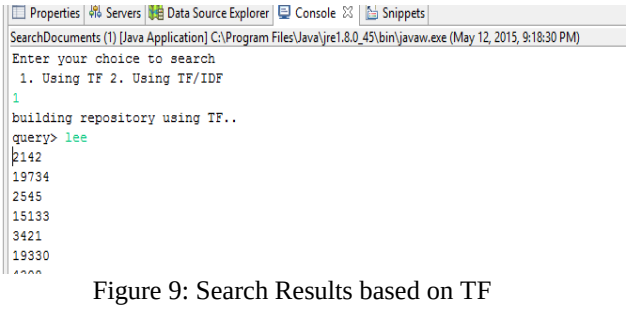


Figure 9: Search Results based on TF

The following diagram shows the search results based TF/IDF, the repository will built based on this and results the document numbers which having a given query.

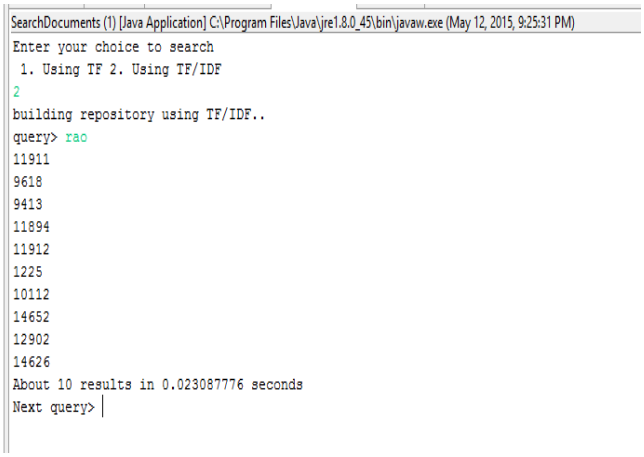


Figure 10: Search results based on TF/IDF

FUTURE WORK

Include TF and IDF in our stock recommendation application. We think UI can be more improved and results can be shown in pie chart, bar chart instead of just tables.

ACKNOWLEDGMENT

This Project is done as part of 'Knowledge discovery and Management' course schedule. We thank Dr. Lee for providing opportunity to work on NLP, TF and IDF Technologies.

REFERENCES

[1] H. Wu and R. Luk and K. Wong and K. Kwok. "Interpreting TF-IDF term weights as making relevance decisions". ACM Transactions on Information Systems, 26 (3). 2008.

[2] Tosey, P. & Mathison, J., (2006) "Introducing Neuro-Linguistic Programming Centre for Management Learning & Development, School of Management, University of Surrey

[3] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in the time-series database.

[4] Bandler, Richard; Grinder, John (1979). Andreas, Steve, ed. Frogs into Princes:Neuro Linguistic Programming. Real People Press. p. 8. ISBN 0911226192

[5] Yahoo API - https://code.google.com/p/yahoo-finance-managed/wiki/YQLAPI

[6] TF and IDF concepts - http://www.tfidf.com/