Company Financial Relation Extractor

Chung-Ling Yao, cly264
Haoran Jia, hj742
Hongzhi Ren, hr777

Abstract

Our project, based on Stanford CoreNLP and Word2Vec tool, used supervised machine learning to train the model that extracts the four most common relations(i.e., acquisition, ownership, invest_in, merger) between companies from financial news articles of Bloomberg News website. After storing all the extracted financial relations into the database, we implemented the search function of one company or a pair. Moreover, on Cytoscape platform, to visualize data we designed a network which is able to display four different categories and the level of strength of relations.

Key Words

Stanford CoreNLP, Word2Vector, Relations Extraction, Supervised Machine Learning, Named Entity Recognition, Data Visualization, Financial News.

1. Introduction

The project extends Stanford NLP's relation extractor into financial area, and has done a lot of work to improve the accuracy of the model. We also developed a tool that can export graphical interpretation of the relations by processing financial news articles during a specific period. The relations between companies are always of great interests. However, the investors have to spend hours reading the news and reports to dig out information such as "Google acquires XYZ company" or "Yahoo invests $1,000,000 in ABC company". Our project focuses on the most common four types of relations between firms: "Acquire", "Invest_In", "Own" and "Merge". These four relations are different since "Acquire" is an action; "Invest_In" only means one company has made investment in another company; "Own" is a status that one company owns another one, and "Merge" means two companies are merged into a new one.

To develop the company relation extractor system, we have developed a structured method to overcome difficulties from each stage. The training technique we use is Stanford NLP Relation Extractor (http://nlp.stanford.edu/software/relationExtractor.shtml). The software edition we use is Stanford CoreNLP version 3.5.2., which requires JAVA 1.8 or higher.

The first step is to improve the NER's accuracy, where we used Word_to_Vector algorithm and Force_to_ORGANIZATION methods; The second step is to hand-tag over thousand of training sentences for our model. The third step involves model testing and model optimization. The final step is the company relations graphical interpretation software development.

2. Improve NER Model to Capture "ORGANIZATION"

The project first uses Stanford NLP Log-linear Part-Of-Speech Tagger (POS) to read text in English and assign parts of speech to each word (and other token), such as noun, verb, adjective, etc. The second step is to use Named Entity Recognition (NER) to label sequences of words in a text which are the names of things, such as person and company names, or gene and protein names. It comes with well-engineered feature extractors for Named Entity Recognition, and many options for defining feature extractors. Included with the download are good named entity recognizers for English, particularly for the 3 classes (PERSON, ORGANIZATION, LOCATION).In our project, we only use named entity "ORGANIZATION" since the four kinds of relations that we are looking at are all about companies (firms, corporations, funds, investment firms, etc).

However, the first problem here is: in the text, "apple" might refer to "apple, the fruit" or "Apple Inc., the company". Also, similar weird thing happens when we find the following sentence:

What is it like to work at Amazon?
(LOCATION)
"Amazon" should be tagged as "ORGANIZATION", but it is classified as LOCATION here!

The second problem is similar: "Apple Inc.", "Apple", "AAPL" all refer to a same entity—the Apple Inc. company. However, "Apple" and "AAPL" might not be tagged as ORGANIZATION. Consider the following sentence:

Discovery buys the company last year.
(PEOPLE)
"Discovery" is classified as "PEOPLE" because "buys" follows, which is a strong sign of "PEOPLE" entity.

To better solve the above problems, we used two methods: Force_to_Organization and Word_to_Vector.

Method1: Force_to_Organization

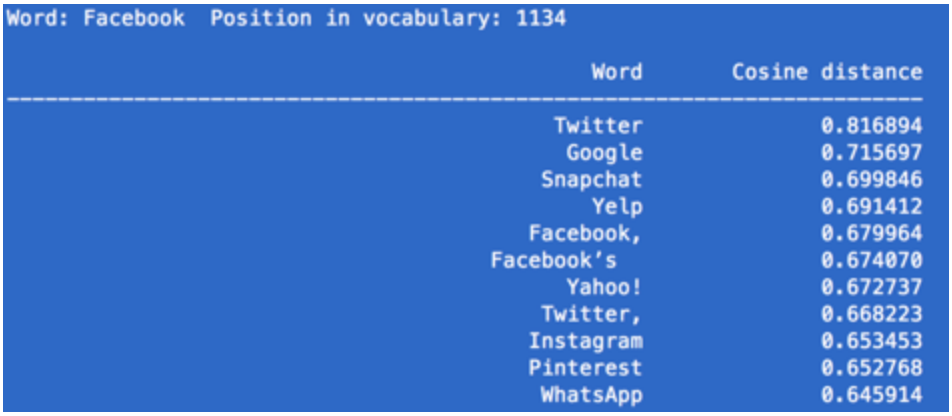
Since all of our texts are from financial news at Bloomberg.com, the probability of mentioning "apple, the fruit" would be very low, especially when the "Apple" is starting with upper case "A". Such simplified form of company names must be forced to be "ORGANIZATION", even they omit the types of business entity, such as "inc.", "INC", "PLC", "LP", "Lp", etc.

We have gathered over 20,000 names of major business entities in US. And provide different simplified form for each of them. These words will be forced to be ORGANIZATION. Thus, in our system, "AAPL", "Apple" and "Apple Inc." are all referring to the same thing.

Method2: Word_to_Vector

The word2vec tool takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words. The resulting word vector file can be used as features in many natural language processing and machine learning applications.

The basic idea is: if both entities are companies, then they must share similar word vectors, which means the "Cosine Distance" between these two words is close to 1. If we are not sure whether a certain word should be tagged as ORGANIZATION or other NERs, we can check the first 3 of its closest words. If three of these words are already been stored as ORGANIZATION in our data file, then this word will be tagged as "ORGANIZATION".



Word: Facebook Position in vocabulary: 1134

Word	Cosine distance
Twitter	0.816894
Google	0.715697
Snapchat	0.699846
Yelp	0.691412
Facebook,	0.679964
Facebook's	0.674070
Yahoo!	0.672737
Twitter,	0.668223
Instagram	0.653453
Pinterest	0.652768
WhatsApp	0.645914

Twitter, has Cosine Distance equal to 0.817, is very similar to Facebook.

Word: Pinterest Position in vocabulary: 21921

Word	Cosine distance
EHarmony	0.700784
Airbnb	0.652825
Facebook	0.652768
Dropbox	0.631220
Yelp	0.617475
Google	0.606942
Yahoo!	0.599597
Amazon.com	0.597794
Flipboard	0.593930
EBay	0.592872
Snapchat	0.583631

If EHarmony, Airbnb, Facebook are all recognized as ORGANIZATION, then "Pinterest" is highly likely to be tagged as ORGANIZATION as well.

The implementation of the Word_to_Vector is to first tag the text using POS. And then use the NER model to tag out entities. And then check every entity's similar words and correct those are suspicious misclassified words. These words will be added to the Force_to_Organization list. Then the process will be repeated again using the new Force_to_Organization list to change more entities' tag from LOCATION or PEOPLE to ORGANIZATION. Theoretically, the iteration will finally converge and no more tags need to be changed, and the process can stop then.

We use 2010-2014 whole year's news articles on Bloomberg News website (over 200,000 pieces of articles, over 135MB plain text file) to train the Word_to_Vector model. Since we have large amount of training text, the result from Word_to_Vector model would be more reliable.

3. Training Sentences Selection

The training sentences are first selected by machine and then by hand. The idea is also using Word_to_Vector to do the first filtering to pick the core words that best reflect the relations. For example, "acquire" is similar to "buy", "purchase", "acquiring" and so on. Sentences including such words will be selected and proof read by human.

Word: acquire Position in vocabulary: 1495

Word	Cosine distance
buy	0.779036
sell	0.707579
purchase	0.663008
acquiring	0.588173

Such words may reflect the same meanings as "acquire"

4. Training Sample Hand Tagging

After the sentences that contain our relations being picked and purified, we developed a JAVA program to pre-process the text and put POS tags and NER tags on it. The following job is to hand-tag the types of relations and pick out the two ORGANIZATIONs between which there is a relation in a sentence. A sample training sentence is shown as:

```
84  1  0  0  0  NNS medical-service 0  0  0
85  1  0  1  0  NN  company 0  0  0
86  1  Org 2  0  NNP/NNP/NNP/NNP Apria/Healthcare/Group/Inc  0  0  0
87  1  0  3  0  VBD be  0  0  0
88  1  0  4  0  VBN acquire 0  0  0
89  1  0  5  0  IN  in  0  0  0
90  1  OTHER  6  0  CD  2008  0  0  0
91  1  0  7  0  IN  by  0  0  0
92  1  Org 8  0  NNP/NNP/NN  Blackstone/Group/LP 0  0  0
93  1  0  9  0  .  .  0  0  0
94
95  8  2  Acquire
```

The last line means (Balckstone Group LP, at position 8) acquires (Apria Healthcare Group Inc, at position 2)

We have hand-tagged over 600 sentences with different relations and mix them with over 600 sentences without any relations inside of them.

5. Make Predictions

After tweaking the source code of Stanford CoreNLP and generating our own coreNLP.jar as external library, we are able to make more precise predictions of the test articles, and give out the desired plain text output extracting the company relations. The sample output shows as following.

```
Acquire
KKR & Co
Sedgwick Claims Management Services Inc.

Acquire
KKR
Mitchell International Inc.

Merge
Lampert
Kmart Holding Corp Austrian Merger Biolitec

Merge
Dixon & Bell
Troutman Sanders

Invest_In
Hitachi
Hiroaki Nakanishi

Merge
Google
Chrysler

Merge
Fiat SpA
Chrysler Group LLC

Merge
Fiat
Chrysler
```

For the first entry in the output above, it means that KKR & Co as a subjective has acquiring-related issues with Sedgwick Claims Management Services Inc. as an objective.

6. Evaluation

Before we build our output graphical software which presents the graphical interpretation of company relations, we have to first test our model. The test samples are splitted into two parts: the first part containing 100 sentences randomly picked from the corpus of Bloomberg Financial News in 2013 without any relations (negative sample); the second part containing 100 sentences randomly picked from the corpus of Bloomberg Financial News in 2013 with at least one relation inside of each of the sentences (positive sample).

Negative Sample Testing Results:

Our model gives 3 false positive (3 fake relations) out of the negative sample. The 3 sentences and their fake results are:

Chief Executive Officer John Chen unveiled a square-screened smartphone at an event in Toronto.
Fake Relation: John Chen Invest_In Toronto

Warnings this week about the pressures on China's economy by Finance Minister Lou Jiwei vindicated Luc De La Durantaye at CIBC Asset Management Inc. in Montreal.
Fake Relation: China Acquire Lou Jiwei

The Aussie and kiwi, like the loonie, have lost their appeal for Steve Lee, head of the foreign-exchange group at Nuveen Asset Management LLC in Chicago.
Fake Relation: Nuveen Asset Management LLC Merge Steve Lee

These sentences are very hard for our model to distinguish using statistical structures. However, LOCATION is can never considered in any of our 4 relations ("country" can never acquire or merge any other companies). Such country names must be eliminated in the output because there is no company in the world that has a name that is the same as a country's name. The easiest thing to do is to scan through a list of country names and eliminate such relations involving country names.

After implementing this simple process, the number of fake relations drops to 1, which makes our accuracy rate rise from 97% up to 99%.

Positive Sample Testing Results:

The potential mistakes that can be made for positive sample would be: misclassification (assign wrong relation to a sentence), fail_to_recognize (fail to extract the relation from a sentence), wrong_sequence (extract wrong subject and object, for example, "IBM acquires Amazon" and

“Amazon acquires IBM” are two different sequences). The priority of the above mistakes is: fail_to_recognize > misclassification > wrong_sequence.

Our model can successfully recognize 59 out of 100 test sentences. The reason for the lower accuracy is that: the sentences from Bloomberg Financial News are usually of long and complicated structures. Various relations, subordinate clauses, names, entities and tenses might be in a same sentences, which makes the training sample quite sparse. Also, there are many key-words/structures that reflect same relations. For example, “AA is going to **acquire** BB”, “AA **acquired** BB”, “BB **agreed to be acquired** by AA”, “AA filed a documentation about the **acquisition** of BB” and etc are all talking about the relation of “Acquire”. We need a lot of training sentences to capture all these features of one relation. However, hand-tagging is time consuming and we don’t have enough time to produce a well-formed training corpus.

Summary of Testing Results:
The average test accuracy of our model is 79%, after we have implemented various techniques to improve its accuracy rate.

7. Implementing Search Function

We used HashMaps and LinkedHashMapsto store the extracted relation entries with their mentioned times which represents the strength of the specific relation between two involved companies. Then we have two main search functions, the one is **searchOneCompany** which will output all the relation entries containing the searched one with mention times, the other is **searchByPair** which will output all the relation entries containing the pair of searched companies with the mention times.

Result entry structure:

company S, relation(activity), company K.

Company S as the subjective is keeping the relation or has the tendency of generating relation with company K as the objective.

For instance,

Google Inc, Own, Motorola

The above entry means Google Inc owns or is going to own or has the possibility to own Motorola.

The sample search results as following, the inputs are case-insensitive.

1. **searchOneCompany** (for example, BlackRock or Google)

```
Please enter a company name you want to search.
blackrock
The results for search the company blackrock as following.
BlackRock,Acquire,Scientific Active Equities = 4 times.
BlackRock,Own,Akbank TAS = 2 times.
BlackRock,Acquire,Barclays Global Investors = 1 times.
```

```
Please enter a company name you want to search.
google
The results for search the company google as following.
Google Inc,Acquire,Nest Labs = 1 times.
Google Ventures,Invest_In,Nest = 1 times.
```

From the displayed results above, we can know the BlackRock-related or google-related financial news and figure out the specific relations or activities with other companies.
Please note that coreference resolution has been implemented which has been showed in the search for “google”.

2. **searchByPair**

```
Please enter a pair of companies delimited by comma.
tumblr, Yahoo
The results for searching the pair of tumblr and Yahoo as following.
Yahoo,Acquire,Tumblr Inc = 1 times.
```

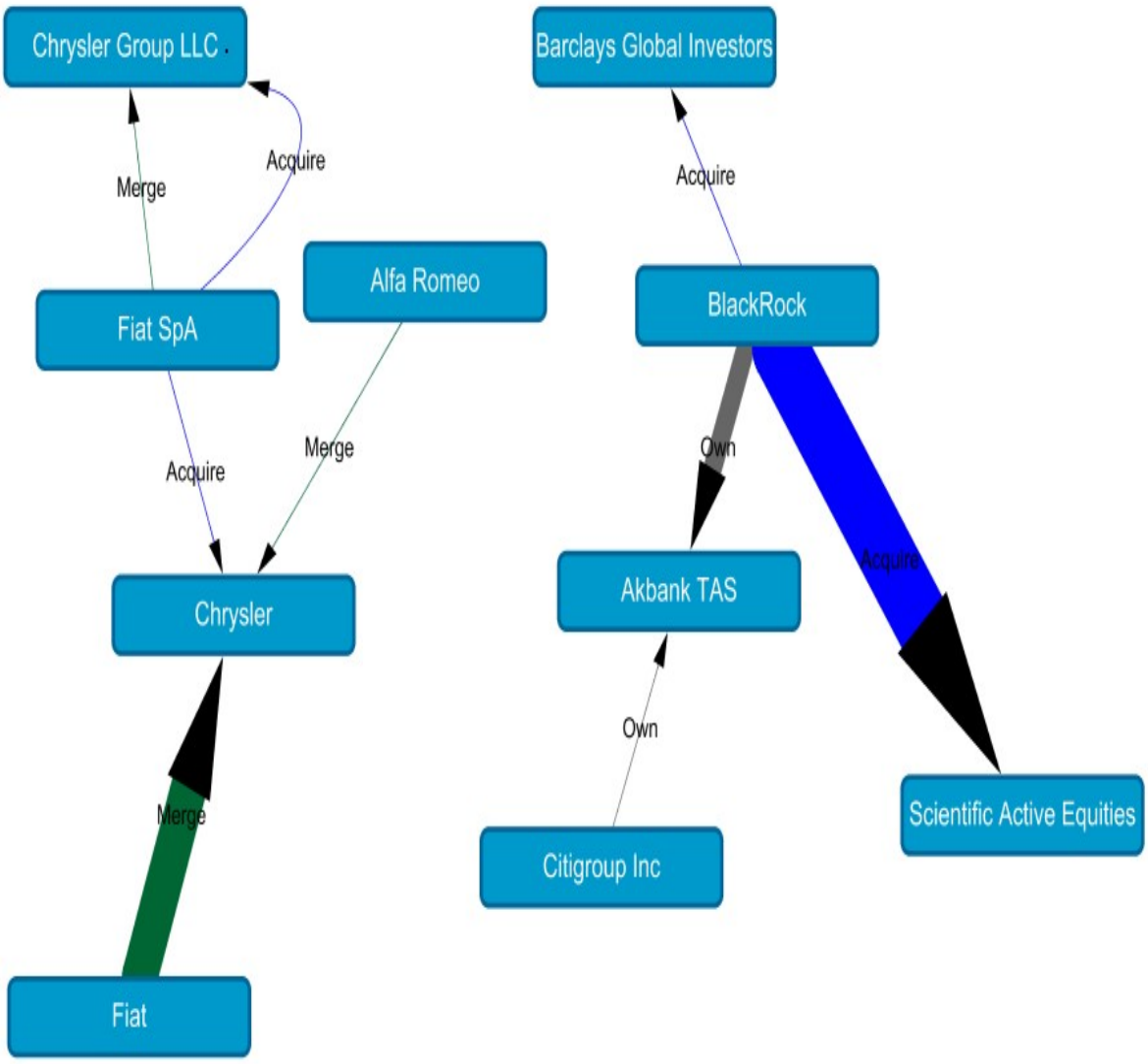
```
Please enter a pair of companies delimited by comma.
sina, alibaba
The results for searching the pair of sina and alibaba as following.
Alibaba,Acquire,Sina = 1 times.
```

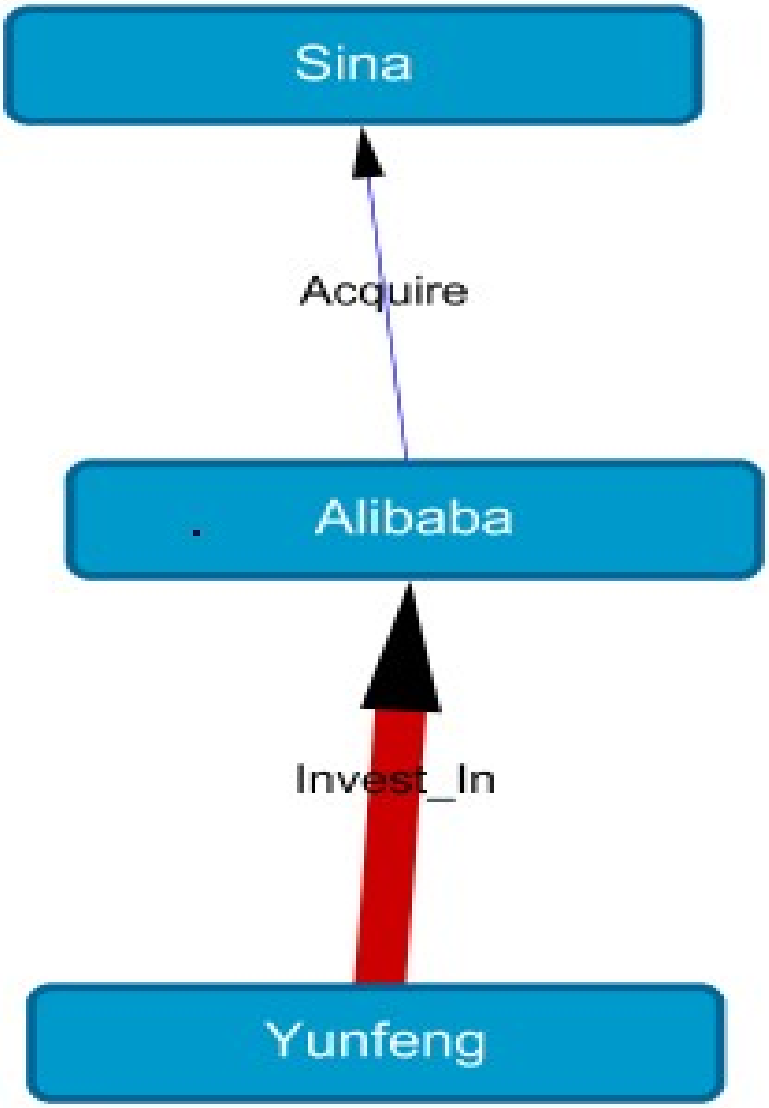
```
Please enter a pair of companies delimited by comma.
alibaba,sina
The results for searching the pair of alibaba and sina as following.
Alibaba,Acquire,Sina = 1 times.
```

From the displayed results above, we can know what happened and what relations are between the pair of companies.
Please note that the company order within the pair won’t influence the results and also coreference resolution has been implemented.

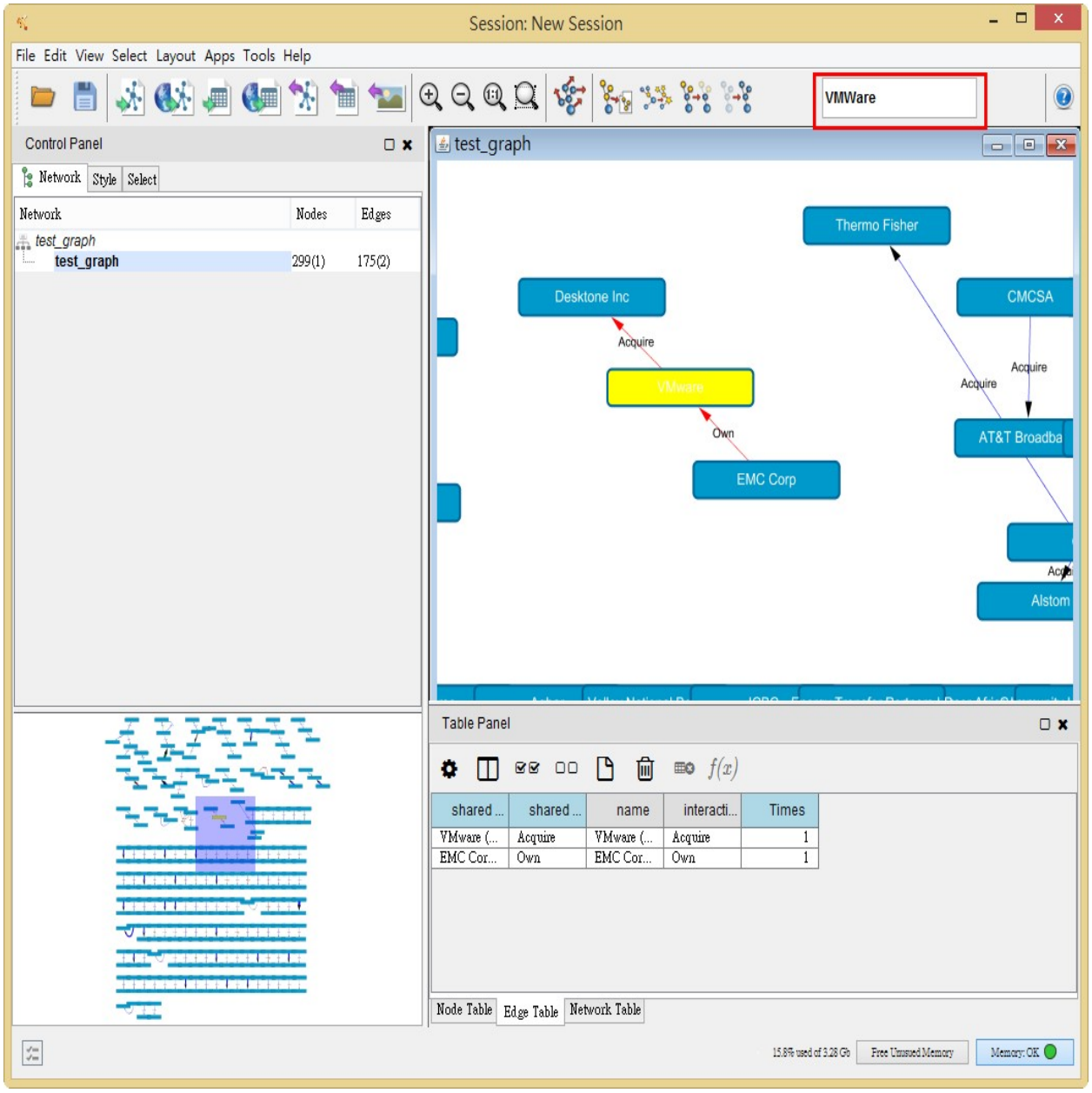
8. Generating Company-Relation Diagrams

We use Cytoscape, an open source software platform, to visualizing the company-relation interaction networks. Since four categories of relationships, i.e. acquire, own, merge and invest_in, have been defined, we used different patterns of edges to show the different relationships. Moreover, the more times the relation mentioned, the stronger the relation represents. So we used the thickness of the edges to show the strength of the relation. Meanwhile, in the huge network, you can seek the related information by searching the specific company as a node.





The company relations shown by Cytoscape (“Acquire” in blue arrows, “Invest_In” in red, “Own” in grey, and “Merge” in green). The thickness of arrows represent the strength of relations.



The search result shown by Cytoscape. Here we search "VMWare", and the node of VMWare becomes yellow. In the left bottom window shows the location of VMWare node in all company-relation interaction networks.

References

1. Information Extraction: Capabilities and Challenges by Ralph Grishman.
2. Foundations of Statistical Natural Language Processing by Christopher D. Manning and Hinrich Schutze, MIT Press, 1999.
3. Introduction to Information Retrieval by Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Cambridge University Press.
4. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics by Daniel Jurafsky and James H. Martin. 2008. . 2nd edition. Prentice-Hall.
5. Automatic acquisition of domain knowledge for Information Extraction by Roman Yangarber New York University, Ralph Grishman New York University ,Pasi Tapanainen Conexor oy, Helsinki, Finland, Silja Huttunen University of Helsinki, Finland