Final Project Report

Categorization of news article by analysis on hypernym tree

(2015 KAIST Spring, CS372)
20112034 Joohee Park(blackmochi@kaist.ac.kr)
20130690 Sumin Han(hsm6911@kaist.ac.kr)

1. Introduction

A hypernym tree is the network where the nodes are the synsets in the hypernym paths obtained from any word in the article and the edges are the relationship obtained from the hierarchy in each hypernym path. Basically, hypernym is a semantic field which includes a word or a phrase as a higher concept. Since words can share a common hypernym at the higher concept, the branch structure can be imagined and the tree structure from the parent node to the leaf node can be drawn. In this research, we considered popular hypernym as the nodes with more children nodes rather than any others. The popular hypernyms will be used to calculate distance to standard hypernym list for each category.

Our team set two goals for this project;

1. **Categorization of article**: Analyze the main keywords of the given article by comparing "popular hypernyms" with golden standard, so that the program can categorize a given article automatically.
2. **Author Identification**: Analyze the similarity of the hypernym tree using the network analysis methods and tools to identify the author. (We expect there is relationship between the writing style and the structure of the hypernym tree)

2. Resources

For the training corpus, our team used **brown corpus**. The Brown Corpus was the first million-word electronic corpus of English, created in 1961 at Brown University. This corpus contains text from 500 sources, and the sources have been categorized by genre, such as news, editorial as follows:

ID	File	Genre	Description
A16	ca16	news	Chicago Tribune: *Society Reportage*

B02	cb02	editorial	Christian Science Monitor: *Editorials*
C17	cc17	reviews	Time Magazine: *Reviews*
D12	cd12	religion	Underwood: *Probing the Ethics of Realtors*
E36	ce36	hobbies	Norling: *RENTING A CAR in Europe*
F25	cf25	lore	Boroff: *Jewish Teenage Culture*
G22	cg22	belles_lettres	Reiner: *Coping with Runaway Technology*
H15	ch15	government	US Office of Civil and Defence Mobilization: *The Family Fallout Shelter*
J17	cj19	learned	Mosteller: *Probability with Statistical Applications*
K04	ck04	fiction	W.E.B. Du Bois: *Worlds of Color*
L13	cl13	mystery	Hitchens: *Footsteps in the Night*
M01	cm01	science_fiction	Heinlein: *Stranger in a Strange Land*
N14	cn15	adventure	Field: *Rattlesnake Ridge*
P12	cp12	romance	Callaghan: *A Passion in Rome*
R06	cr06	humor	Thurber: *The Future, If Any, of Comedy*

Also used **nltk.wordnet** to find the synsets and hypernym. To remove out redundant words, the word will be filtered by **nltk.stopwords**. In addition, **BeautifulSoup** was used to extract word list from news article, and **Pajek**(http://vlado.fmf.uni-lj.si/pub/networks/pajek/) was used to draw network.

3. Process

For the overview of approach First, we (1) extract hypernym tree (V,E) from each file in the Brown Corpus. Then (2) calculate popular hypernym nodes from the network. By sum of the results for each category, (3) we can build up standard FreqDict for each category. Later, (4) extract hypernym network from our targeting article and calculate FreqDist for popular hypernyms. Then (5) we calculate the distance to each category and find the closest category: which best category matches up for the target article.

3.1. Extract hypernym tree (V, E) from Brwon Corpus

```
def addToList(L, e):
    if e not in L:
        L.append(e)
… (some code) ...
for w in brown.words(f):
  if w.lower() not in stpwd and w not in W and w.isalpha():
    addToList(W, w)
    syns = wn.synsets(w)
    if syns != []:
      ws = syns[0] # use the first synset
      leafN += 1
      for path in ws.hypernym_paths():
        if(len(path) > maxPath): maxPath = len(path)
        addToList(V, path[0].name()) # adding first node
        for i in range(1, len(path)):
          addToList(V, path[i].name())
          addToList(E, (V.index(path[i].name()), V.index(path[i-1].name())))
```

- **addToList** is the function which adds element to the list without overlap.

- For word list in the file from a certain category, add the word to list W to avoid overlap, and find the best fit synset for that word using wn.synsets(w). Later we look up the hypernym paths and add the elements to the node list V and edge list E properly.

By adding several lines of code,.net and .clu file can be made as input files for Pajek program to draw the network, and cluster to recognize popular hypernym as red dots.

3.2. Calculate popular hypernym nodes from the network

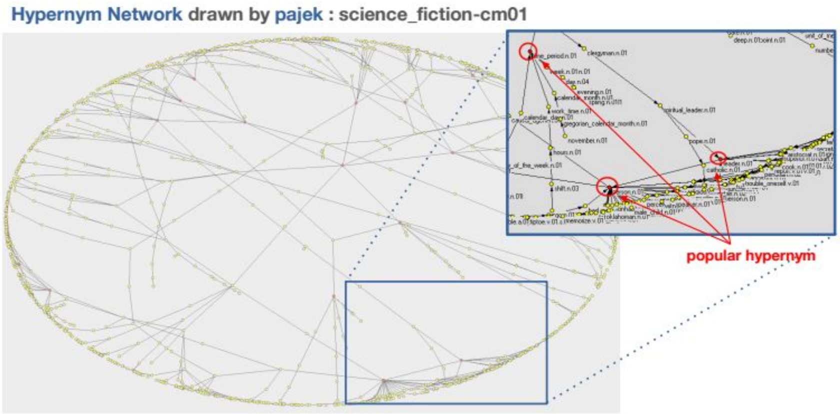
```
for v, w in E:
  if w not in EforVn:
    EforVn[w] = 0
  EforVn[w]+=1
fd = nltk.FreqDist(EforVn)
fdL = [w for w, c in fd.most_common(300)]
```

- Then count the number of incoming edges to w which is the number of children for that node. EforVn is dictionary, and FreqDist can be calculated.

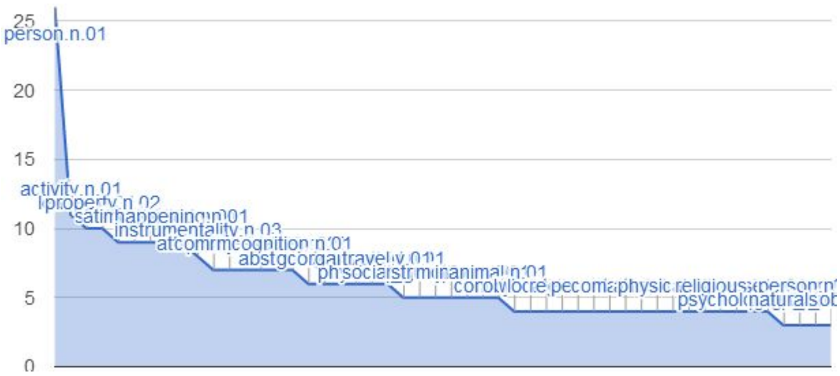
The sample data with file name, number of vertices, edges, maximum path(tree height), and 20 most common synsets(popular hypernyms) are listed as below:

#FILE:science_fiction-cm01
#Vertices 1344
#Edges 1221
#MaxPath 16

Rank	Count	Synset
0	26	person.n.01
1	11	activity.n.01
2	10	leader.n.01
3	10	property.n.02
4	9	state.n.02
5	9	artifact.n.01
6	9	act.n.02
7	9	time_period.n.01
8	9	happening.n.01
9	8	instrumentality.n.03
10	7	attribute.n.02
11	7	quality.n.01
12	7	measure.n.02
13	7	communication.n.02
14	7	message.n.02
15	7	cognition.n.01
16	6	abstraction.n.06
17	6	group.n.01
18	6	change.v.02
19	6	device.n.01



- The network can be drawn using **Pajek**.
- Red dots are the popular hypernyms as in the list in left cell.
- The distribution graph is also drawn as below.



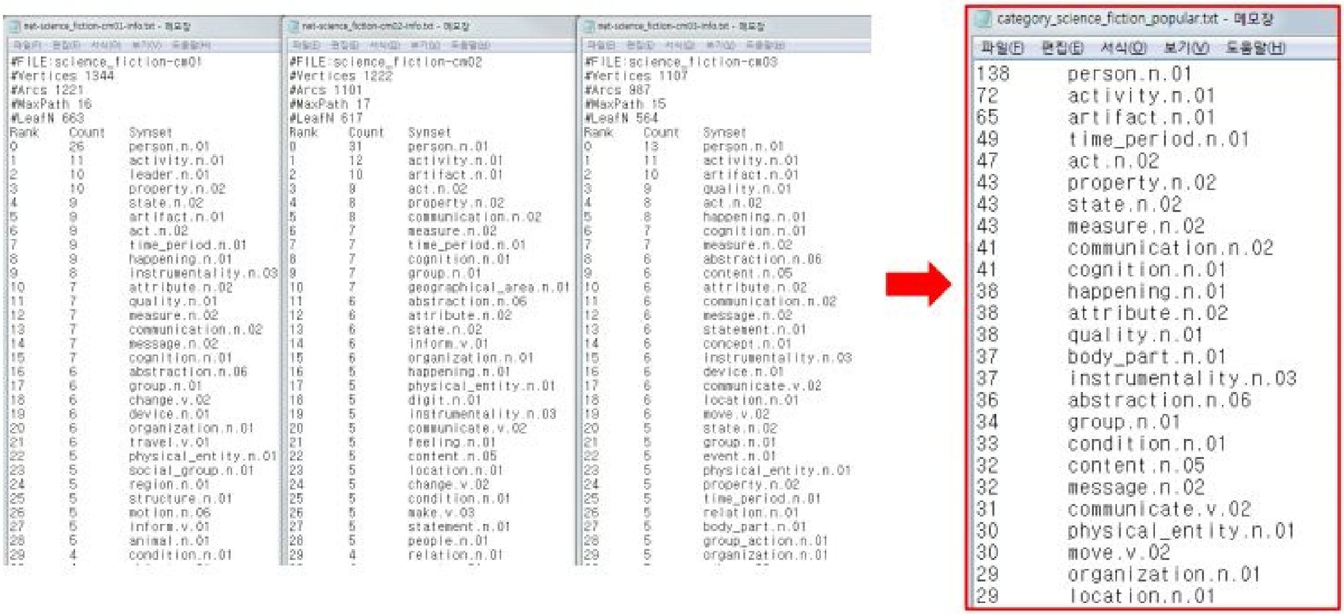
By processing each file, the statistics were calculated as table below.

category	#files	leaves	maxpath	V	E
adventure	29	609.482759	16.37931	1243.413793	1133.241379
belles_lettres	75	649.96	15.973333	1254.706667	1137.626667
editorial	27	708.555556	16.555556	1347.185185	1233.925926
fiction	29	629.896552	15.62069	1294.172414	1193.517241
government	30	563.3	15.8	998.066667	900.633333
hobbies	36	617.694444	16.305556	1164.527778	1067.388889
humor	9	698.111111	15.444444	1411.444444	1291.444444
learned	80	554.0875	15.7375	1004.175	890.525
lore	48	645.270833	16.375	1228.104167	1116.166667
mystery	24	575.083333	15.416667	1181.708333	1081.375

news	44	669.090909	16.227273	1299.545455	1217.068182
religion	17	612.882353	16.352941	1146.235294	1032.588235
reviews	17	768.882353	16	1460.470588	1331.470588
romance	29	586.448276	15.551724	1215.62069	1115.137931
science_fiction	6	631.333333	15	1235	1110.666667

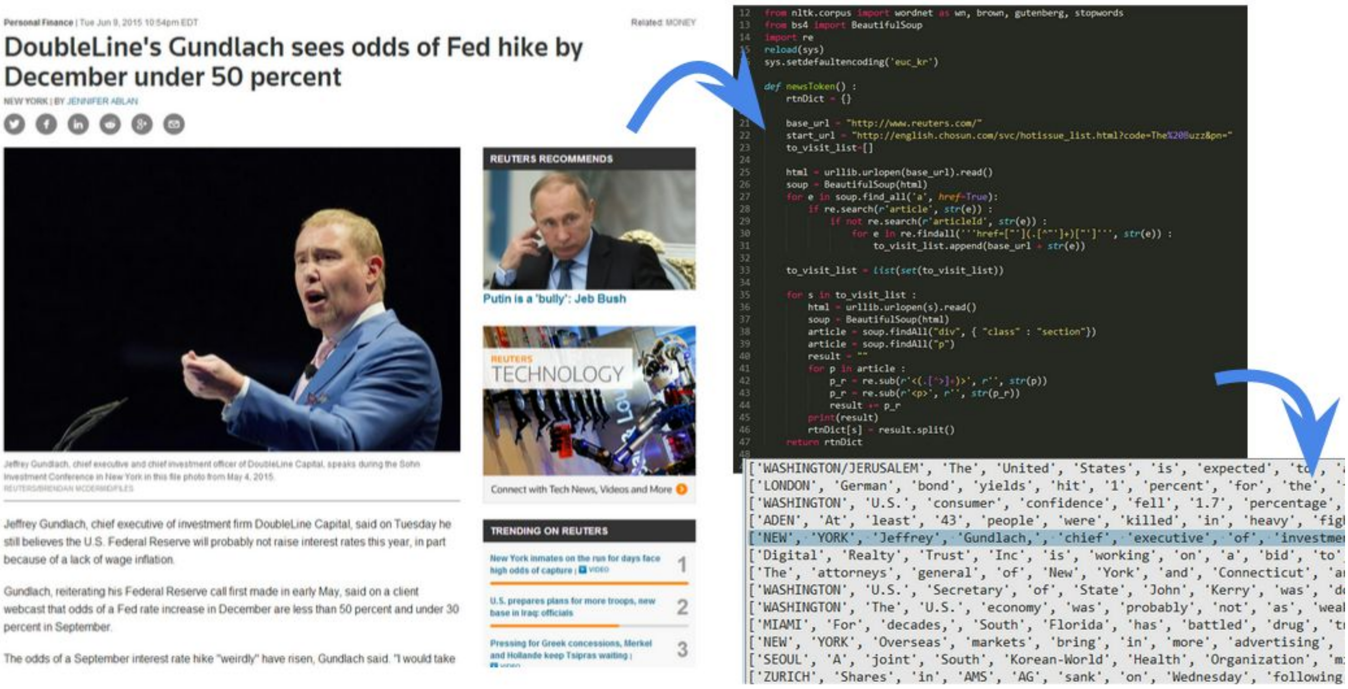
*maxpath represents the maximum height of each hypernym tree. The average values are denoted.

3.3. Make into standard FreqDict for each category

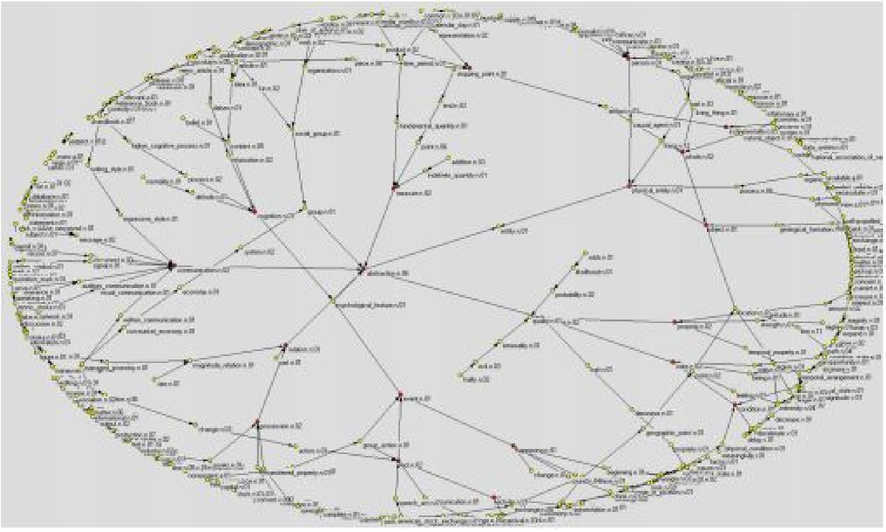


After the process with popular hypernyms, the standard for each category can be created by combining data. Each category(adventure, editorial, news …) has its own standard category and it will be used to calculate distance from hypernym tree of our target article.

3.4. Calculate the popular hypernym of target article



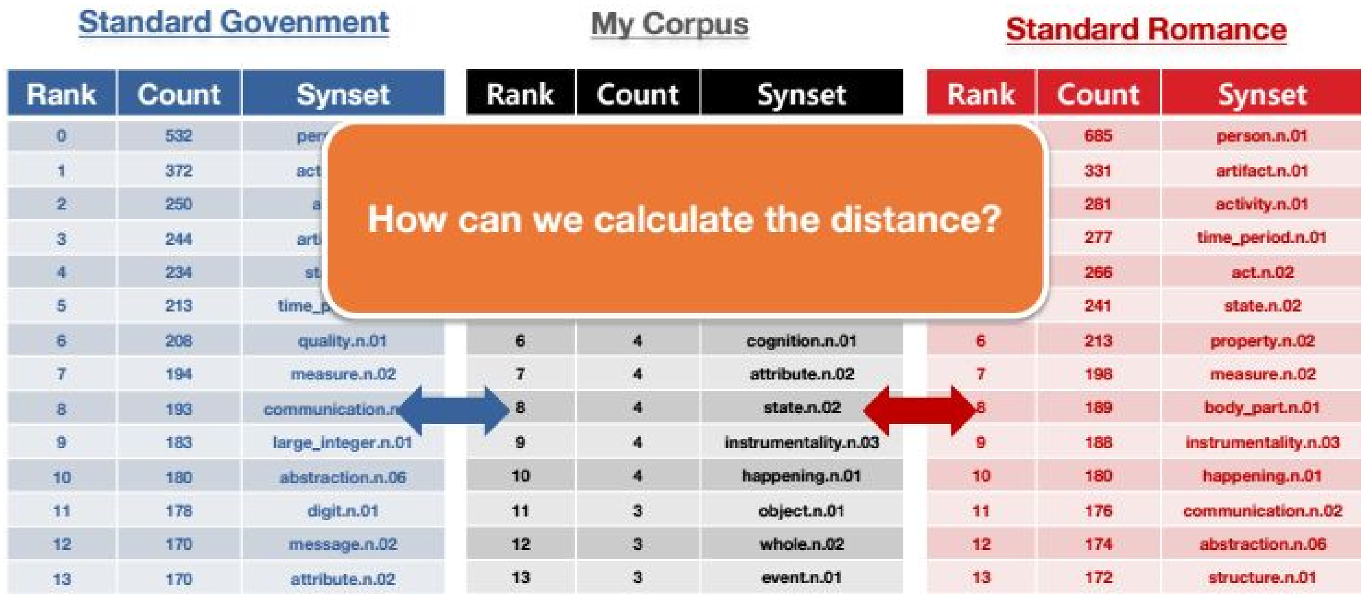
By using code, we can extract word list from target article. It downloads the html news article, and find the article part, and then reduce the html tag and split to make into word list.



Rank	Count	Synset
0	7	person.n.01
1	7	communication.n.02
2	6	abstraction.n.06
3	5	act.n.02
4	5	activity.n.01
5	4	physical_entity.n.01
6	4	cognition.n.01
7	4	attribute.n.02
8	4	state.n.02
9	4	instrumentality.n.03
10	4	happening.n.01
11	3	object.n.01
12	3	whole.n.02
13	3	event.n.01
14	3	measure.n.02
...		

Similar to process (1) and (2), the hypernym network and the list of popular hypernyms can be calculated.

3.5. Calculate distance and find closest category



Rank	Count	Synset
0	532	per
1	372	act
2	250	a
3	244	art
4	234	st
5	213	time_p
6	208	quality.n.01
7	194	measure.n.02
8	193	communication.n
9	183	large_integer.n.01
10	180	abstraction.n.06
11	178	digit.n.01
12	170	message.n.02
13	170	attribute.n.02

Rank	Count	Synset
6	4	cognition.n.01
7	4	attribute.n.02
8	4	state.n.02
9	4	instrumentality.n.03
10	4	happening.n.01
11	3	object.n.01
12	3	whole.n.02
13	3	event.n.01

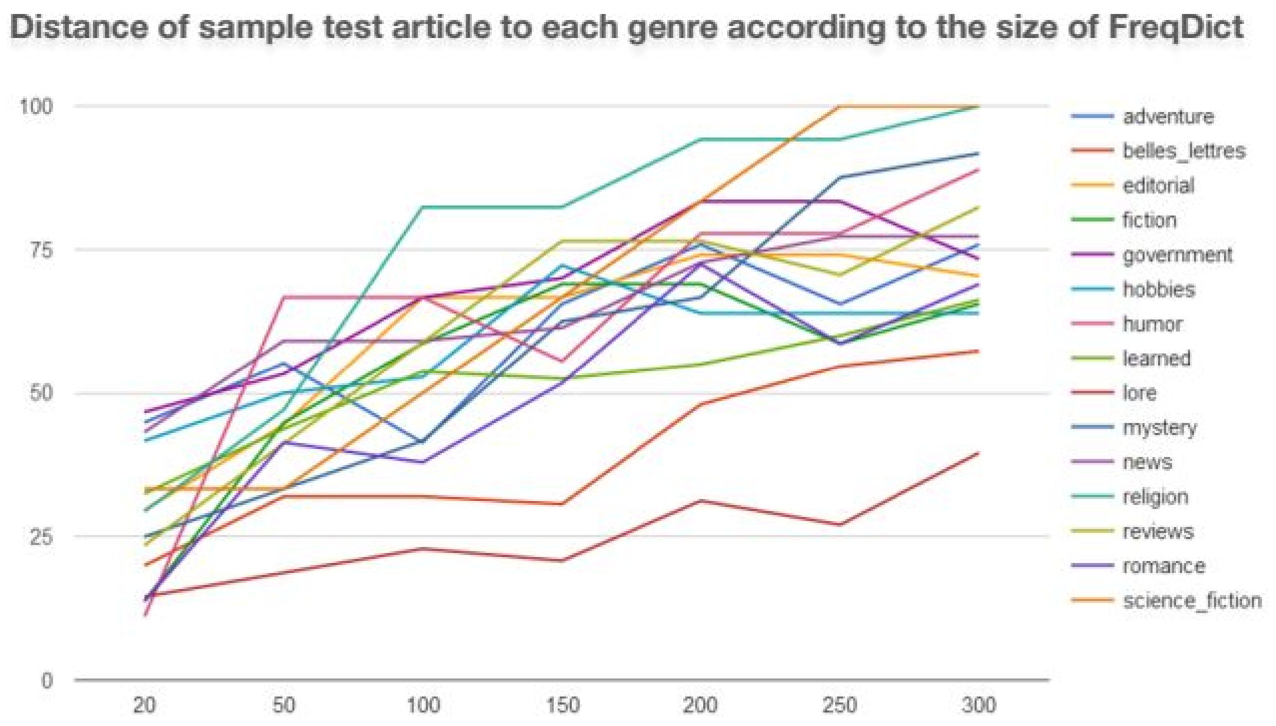
Rank	Count	Synset
	685	person.n.01
	331	artifact.n.01
	281	activity.n.01
	277	time_period.n.01
	266	act.n.02
	241	state.n.02
6	213	property.n.02
7	198	measure.n.02
8	189	body_part.n.01
9	188	instrumentality.n.03
10	180	happening.n.01
11	176	communication.n.02
12	174	abstraction.n.06
13	172	structure.n.01

Now we have a popular hypernym list of our target article and the standard popular hypernym list for each category. We have to calculate distance but how can we measure it? Our approach was to calculate distance as vector.

```
def dist(L1, L2):
    d = 0
    for e in L1:
        d += (L1.index(e) - L2.index(e))**2 if e in L2 else len(L2)**2
    return math.sqrt(d)
```

For example, when there is two list L1 = [1, ‘a’, 4, 3.0], and L2 = [‘a’, 3.0, 7, 1], we first look up each element in L1. Starting from 1, since L1.index(1) == 0 and L2.index(1) == 3, we add (3-0)^2 to variable d. Similarly, L1.index(‘a’) == 1 and L2.index(‘a’) == 0, so we add (0-1)^2 to variable d. Now it’s time to proceed with 4, but L2 do not contain 4. Then we just add len(L2)^2 to represent it is far away. Finally L1.index(3.0) == 3 and L2.index(3.0) == 1, so add (1-3)^2 to variable d. Finally we square root d (but it’s not necessary because we only need to measure the relative distance to each category). In this case, d = sqrt((3-0)^2 + (0-1)^2 + 4^2 + (1-3)^2) = 5.47.

We applied this to brown corpus, which is the source for our standardized popular hypernym expecting the result would be that it will distinct each category precisly even in the small number of popular hypernyms we use for distance calculation. However, it could recognize its own category when we use more than 300 words. The graph of accuracy is drawn below.



4. Result and Discussion

We tried this algorithm with two different news company:

- chosun english news (http://english.chosun.com/)
- reuters (http://www.reuters.com/)

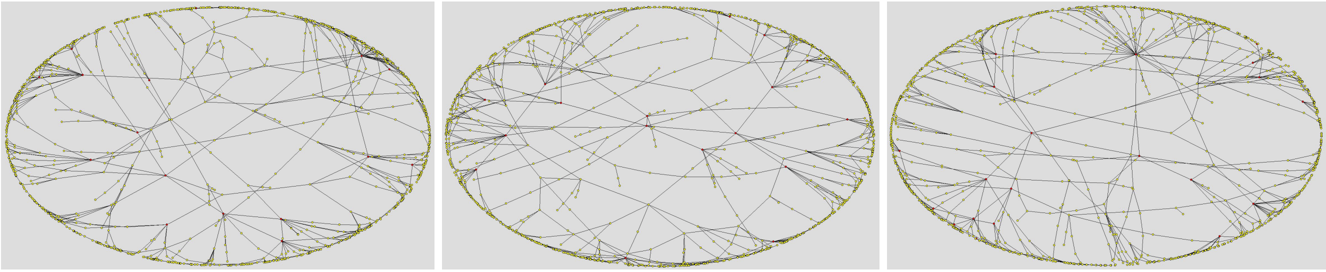
The results are below, however, articles from chosun news has not enough words, so we could only compare for 100 popular hypernyms when we calculating the distance. That is the reason why their category are somewhat strange (red box). However, reuters corpus was rich in hypernyms, so we could use 300 popular hypernyms for our distance calculation (yellow box). That is why they are categorized reasonably.

News Title	1st closest	2nd closest	3rd closest
Who Are the Unhappiest Koreans?	learned	government	science_fiction
Social Skills Under Threat as Texting Becomes the Norm	learned	lore	hobbies
Over-60s Become Most Powerful Consumers Group	religion	adventure	editorial
What Korean Bathhouses Can Teach Visitors	mystery	fiction	romance
Hit Products That Confound Marketers	learned	government	religion
N.Korean Propaganda Against the South Is Failing	lore	learned	news
CNN Lists 10 Areas Where Korea Leads the World	hobbies	learned	government
How Modern Life Reduces Sperm Count	learned	hobbies	government
Dollar, Yuan Rule in N.Korea	learned	government	lore
Most S.Korean Men Would Marry N.Korean Women	adventure	hobbies	lore
Why More Young People Decide to Take It Easy	hobbies	science_fiction	lore
Young People Got to Extremes to Bolster Resume	mystery	learned	lore
90% of Foreigners Would Date a Korean	editorial	lore	government
Most Couples Still Depend on Parents for Marriage	government	news	religion

More Than 50,000 Chinese Study in Korea	hobbies	news	government
Chinese Couples Fly to Korea for Wedding Photos	government	news	learned
Has Korea Gotten Any Safer in the Last 20 Years?	belles_lettres	reviews	hobbies
Divorce Consultants Blossom as Sanctity of Marriage Wilts	government	learned	science_fiction
New Pyongyang Mall Breaks Every Capitalist Taboo	hobbies	lore	news
Chinese Embrace Korean Word	editorial	reviews	fiction
How to Get a Refreshing Night's Sleep	hobbies	adventure	mystery
Fewer Young People Go On to University	mystery	science_fiction	hobbies
S.Koreans Have Mixed Feelings About Reunification	learned	lore	belles_lettres
Chinese Tourists Go Mad for Korean Rice Cookers	learned	hobbies	government
U.S. prepares plans for more troops, new base in Iraq: officials	news	editorial	belles_lettres
Bund yield hits 1 percent as stock markets halt sell-off	editorial	learned	news
At least 43 killed in Yemen clashes as parties prepare for talks	editorial	news	learned
DoubleLine's Gundlach sees odds of Fed hike by December under 50 percent	government	editorial	learned
Pentagon bars discrimination against gays, lesbians in uniform	editorial	news	government
Apple Music faces antitrust scrutiny in NY, Connecticut	editorial	learned	government
U.S. Secretary of State Kerry tweets photo of himself in hospital	editorial	news	government
Cheap, synthetic 'flakka' dethroning cocaine on Florida drug scene	lore	government	editorial
Exclusive: Facebook earns 51 percent of ad revenue overseas	government	editorial	learned
WHO team urges South Korea to reopen schools as more close in MERS crisis	editorial	news	government
Convicted killer in New York prison break on third escape attempt	news	religion	fiction
Dow opens higher for first time in five days	government	news	editorial
Tokio Marine to buy HCC Insurance for $7.5 billion	editorial	news	government
Bayer sells Diabetes Care business to Panasonic Healthcare	government	learned	editorial
Spotify raises $115 million in share sale	news	government	editorial
U.S. Marine goes on trial again for killing of Iraqi civilian	editorial	belles_lettres	news
Pimco's Mather says firm expects Fed to begin raising rates in September	government	editorial	religion
As Greece lurches toward default, businesses hit the wall	editorial	government	lore
Putin is a 'bully,' U.S. needs to respond resolutely: Jeb Bush	editorial	news	government
Pressing for Greek concessions, Merkel and Hollande keep Tsipras waiting	news	editorial	government
Dozens arrested in European cyber crime sweep: Europol	government	editorial	learned
Suicide bomber attacks tourist site in Luxor, four Egyptians wounded	government	editorial	hobbies
House lawmakers overcome hurdle on key trade bill	news	editorial	government
Pet food maker Blue Buffalo files for IPO of up to $500 million	government	news	editorial
E-cigarette usage surges in past year: Reuters/Ipsos poll	government	learned	lore
Apple drives vehicles to collect data to improve Maps	editorial	government	hobbies
Burwell says U.S. Congress should fix Obamacare if court rules against it	editorial	news	belles_lettres
Pakistan military says 19 militants, 7 soldiers killed in clash	editorial	news	government

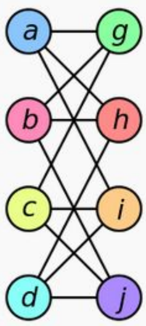
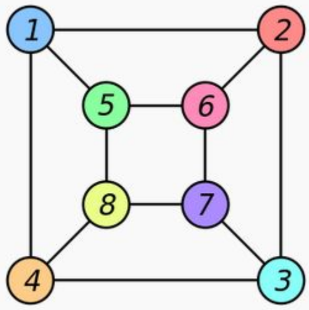
5. Further Research

So far in this report, we only compared the famous hypernym nodes by their degree for analyzing hypernym tree structure. To analyze the tree structure, we first build up network image to find the approximate similarity. However, it is not easy to find without any mathematical approach.



(Network from FILE:romance-cp01, romance-cp02, romance-cp03)
They are similar and different in some way, but how can we measure mathematically?

To improve our ideas, we may apply mathematical theory of graph to our project : we can compare the tree structure based on graph isomorphism.

Graph G	Graph H	An isomorphism between G and H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

(Source : Wikipedia)

A graphs G is said to be isormorphic to graph H if thereis a bijection between the vertex sets of G and H $f\colon V(G) \to V(H)$ such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. Determining the isomorphism between two graph is known as NP-hard, but if we are just considering about the trees, then there is an efficient algorithm for detecting congruences (Kelly, 1974). So by using this tree congruence theorem, we are expecting to improve the performance of our solution in the future.

6. Research about other existing approaches : Machine-learning based document classification

There are number of trials to classify document by their subject for a long time. We tried to classify the document by analyzing the hypernym tree structure, but most of other researches take machine-learning based approach. Though there are a lot of machine learning algorithms known, some approaches are thought to be effective in document categorization : K-nearest neighbor clustering algorithm (KNN), Support Vector Machine (SVM) and Naive Bayes Classifier. KNN is an unsupervised learning algorithm while the others are supervised learning algorithm.

In our approach, we need to represent a given text into some kind of structure that expresses the property of it and it was hypernym tree. Machine-learning-based approach also needs to represent the given article into some features that needs to train the model. According to previous researches (Joachims, 1998) [1] , the bag of words that excludes the stop-words can generate reasonable feature vectors for the given text. Since all words are classified as a single feature, the feature space becomes enormous, so usually SVM shows better performance than KNN by avoiding overfitting problem and its superior capability for handling sparse feature matrices.

	Bayes	Rocchio	C4.5	k-NN	SVM (poly) degree $d =$ 1	2	3	4	5	SVM (rbf) width $\gamma =$ 0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

Fig. 2. Precision/recall-breakeven point on the ten most frequent Reuters categories and microaveraged performance over all Reuters categories. k-NN, Rocchio, and C4.5 achieve highest performance at 1000 features (with $k = 30$ for k-NN and $\beta = 1.0$ for Rocchio). Naive Bayes performs best using all features.

The above table shows the result of classifying Reuters Corpus into its categories by using some famous machine learning algorithms [1]. Though we can not compare this results directly to our approch because we used Brown corpus instead of Reuters Corpus, the result from the above table looks quiet powerful. So in conclusion, machine-learning based approach for document classification by subject is also promising and posseses large potential.

7. Appendix

Source Code:

1. **hypernetwork.py**: Shows option to make network files for specific category. Just pressing enter will create for all the categories.
2. **hyperall-statistics.py**: After creating all the networks for hypernetwork (1), ***category_statistics.txt*** file for overall result and ***category_'ctg'_popular.txt*** file for list of popular hypernyms will be created. ***category_'ctg'_popular.txt*** file is necessary to calculate hyper-closest because we use this file for the standard.
3. **hyper-closest.py**: Calculating correction for each category. The correction graph in this report was based on this data. ***dist_result.txt*** represents the best category for each file in the Brown Corpus, and ***dist_correct.txt*** containing accuracy data.
4. **news_chosun.py**: extracts word list from english chosun news article.
5. **news_reuters.py**: extracts word list from reuters news article.
6. **hyper-closest-chosun.py:** calculates the closest category for chosun news article.
7. **hyper-closest-reuters.py**: calculates the closest category for reuters article.