# Capstone Project-4
## Topic Modelling On News Articles
### (Unsupervised ML Project)

By- Prasad Khedkar
*(Individual Project)*

# Points of Discussions :

1. Problem Statement
2. Summary of the dataset
3. Data Info
4. Data Cleaning
5. NLP Text Processing
   - Removing HTML tags/ URLs
   - Removing non-word characters
   - Removing Punctuations
   - Removing Numbers
6. Tokenization
7. Stop Words
8. Lemmatization.
9. Vectorizer Used
10. Different Model Implementations
11. Evaluations of Models
12. Final Conclusion.

# Problem Statement :

As the data is rapidly getting accumulated day by day , different machine learning methods dealing with different kinds of data types are being formulated.
One such method is the Topic Modelling which comes under Natural Language Processing (NLP) and mainly deals with the text data.

*Topic Modelling identifies the topics that best describes the given set of documents(text format).*

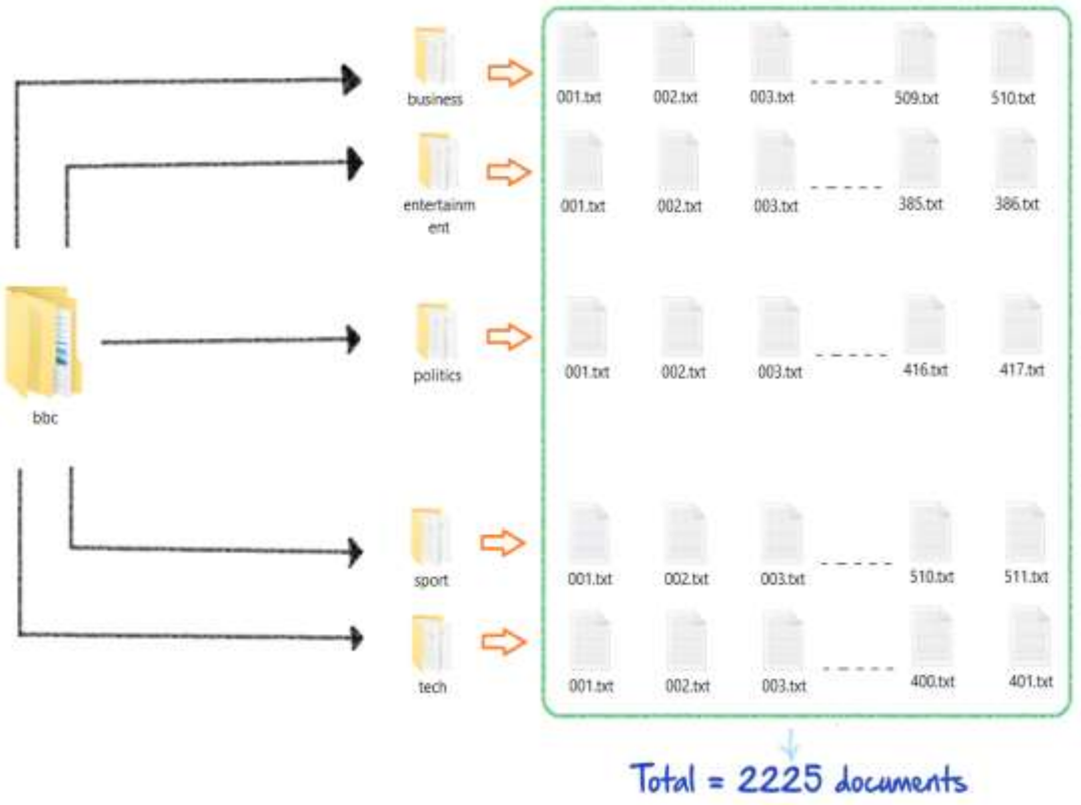
***In the given problem, our main task is to obtain the major topic/themes for given collection of BBC news which are in .txt format, using different algorithms.***

# Summary of the Dataset :

Given data consists of total 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005.

Topical Areas : 5 (Business, Entertainment, Politics, Sport, Tech)

We are fitting only txt documents in our models and not their labels(Topical Areas)



# After loading our data into the dataframe using pandas -

```
df_raw.head()
```

| | News_text | News_Theme |
|---|---|---|
| 0 | b'Ad sales boost Time Warner profit\n\nQuarter... | business |
| 1 | b'Dollar gains on Greenspan speech\n\nThe doll... | business |
| 2 | b'Yukos unit buyer faces loan claim\n\nThe own... | business |
| 3 | b'High fuel prices hit BA\'s profits\n\nBritis... | business |
| 4 | b"Pernod takeover talk lifts Domecq\n\nShares ... | business |

```
df_raw.shape
```

```
(2225, 2)
```

```
df_raw.tail()
```

| | News_text | News_Theme |
|---|---|---|
| 2220 | b'BT program to beat dialler scams\n\nBT is in... | tech |
| 2221 | b'Spam e-mails tempt net shoppers\n\nComputer ... | tech |
| 2222 | b'Be careful how you code\n\nA new European di... | tech |
| 2223 | b'US cyber security chief resigns\n\nThe man m... | tech |
| 2224 | b'Losing yourself in online gaming\n\nOnline r... | tech |

```
df['News_text'][0]
```

```
b'Ad sales boost Time Warner profit\n\nQuarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (\xc2\xa3600m) for the three months to December, from $639m year-earlier.\n\nThe firm, which is now one of the biggest investors in Google, benefited from sales of high-speed internet connections and higher advert sales. TimeWarner said fourth quarter sales rose 2% to $11.1bn from $10.9bn. Its profits were buoyed by one-off gains which offset a profit dip at Warner Bros, and less users for AOL.\n\nTime Warner said on Friday that it now owns 8% of search-engine Google. But its own internet business, AOL, had has mixed fortunes. It lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters. However, the company said AOL\'s underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues. It hopes to increase subscribers by offering the online service free to TimeWarner internet customers and will try to sign up AOL\'s existing customers for high-speed broadband. TimeWarner also has to restate 2000 and 2003 results following a probe by the US Securities Exchange Commission (SEC), which is close to concluding.\n\nTime Warner\'s fourth quarter profits were slightly better than analysts\' expectations. But its film division saw profits slump 27% to $284m, helped by box-office flops Alexander and Catwoman, a sharp contrast to year-earlier, when the third and final film in the Lord of the Rings trilogy boosted results. For the full-year, TimeWarner posted a profit of $3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to $42.09bn. "Our financial performance was strong, meeting or exceeding all of our full-year objectives and greatly enhancing our flexibility," chairman and chief executive Richard Parsons said. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins.\n\nTimeWarner is to restate its accounts as part of efforts to resolve an inquiry into AOL by US market regulators. It has already offered to pay $300m to settle charges, in a deal that is under review by the SEC. The company said it was unable to estimate the amount it needed to set aside for legal reserves, which it previously set at $500m. It intends to adjust the way it accounts for a deal with German music publisher Bertelsmann\'s purchase of a stake in AOL Europe, which it had reported as advertising revenue. It will now book the sale of its stake in AOL Europe as a loss on the value of that stake.\n'
```

# Data Info :

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2225 entries, 0 to 2224
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   News_text   2225 non-null   object
 1   News_Theme  2225 non-null   object
dtypes: object(2)
memory usage: 34.9+ KB
```

We have two columns having datatype object

# Data Cleaning :

First deep copy is created

```
df = df_raw.copy()
```

Checked null values, and found out none present in our dataset

```
df.isnull().sum()
```

```
News_text      0
News_Theme     0
dtype: int64
```

After using drop_duplicates() method shape of the dataframe (df) changed from 2225 to 2127 indicating 98 values were duplicates and removed from the dataframe

# NLP Text Processing :

Removed HTML tags and URLs from the data using -

```
df['News_text'] = [BeautifulSoup(k).get_text() for k in df['News_text'] ]
```

```
df['News_text'] = [re.sub(r'https?://\S+|www\.\S+', '', k1) for k1 in df['News_text']]
```

Removed non-word characters using regex -

```
df['News_text'] = [re.sub(r"\\n\\n", " ",k3) for k3 in df['News_text']]
df['News_text'] = [re.sub(r"\\\'s", " ",k4) for k4 in df['News_text']]
df['News_text'] = [re.sub(r"\\\'", " ", k5) for k5 in df['News_text']]
df['News_text'] = [re.sub(r"\\n\'", " ", k6) for k6 in df['News_text']]
df['News_text'] = [re.sub(r"\n\n", " ", k7) for k7 in df['News_text']]
df['News_text'] = [re.sub(r'\\xc2\\xa3','\xA3', k8) for k8 in df['News_text']]
df['News_text'] = [re.sub(r"\'s",'', k9) for k9 in df['News_text']]
df['News_text'] = [re.sub(r'\\n"','', k10) for k10 in df['News_text']]
df['News_text'] = [re.sub(r'b"','', k11) for k11 in df['News_text']]
df['News_text'] = [re.sub(r'\n','', k12) for k12 in df['News_text']]
```

## Removed punctuations by defining following function -

```python
def rem_punct(text):
  """This function will remove punctuations."""
  text_no_punct = [char for char in text if char not in '!"#$%&\'()*+,-./:;?@[\\]^_{|}~`£' ]
  text_no_punct = ''.join(text_no_punct)
  return text_no_punct
```

```python
df['News_text'] = df['News_text'].apply(lambda x: rem_punct(x))
```

## Removed numbers using -

```python
df['News_text'] = [re.sub(r'\d+','', d1) for d1 in df['News_text']]
```

# Tokenization :

Tokenization is converting the given text into the list of words aka tokens.

For that following function is defined and applied to get the tokens -

```
def tokenize(text):
  """ This function gives list of tokens."""
  tokens = re.split('\W+', text)
  return tokens
```

```
df['Tokens'] = df['News_text'].apply(tokenize)
```

```
df.head()
```

| | News_text | News_Theme | Tokens |
|---|---|---|---|
| 0 | ad sales boost time warner profit quarterly pr... | business | [ad, sales, boost, time, warner, profit, quart... |
| 1 | dollar gains on greenspan speech the dollar ha... | business | [dollar, gains, on, greenspan, speech, the, do... |
| 2 | yukos unit buyer faces loan claim the owners o... | business | [yukos, unit, buyer, faces, loan, claim, the, ... |
| 3 | high fuel prices hit ba profits british airway... | business | [high, fuel, prices, hit, ba, profits, british... |
| 4 | pernod takeover talk lifts domecq shares in uk... | business | [pernod, takeover, talk, lifts, domecq, shares... |

# Stop Words :

Stopwords are words like – on, in, me, and, or ... have to be removed from the dataset.
For that following function is defined and applied -

```
def remove_sw(tokenized_list):
  """This function will remove stopwords."""
  text = [word for word in tokenized_list if word not in swds]
  return text
```

```
df['Tokens(without stop words)'] = df['Tokens'].apply(remove_sw)
```

```
df.head()
```

| | News_text | News_Theme | Tokens | Tokens(without stop words) |
|---|---|---|---|---|
| 0 | ad sales boost time warner profit quarterly pr... | business | [ad, sales, boost, time, warner, profit, quart... | [ad, sales, boost, time, warner, profit, quart... |
| 1 | dollar gains on greenspan speech the dollar ha... | business | [dollar, gains, on, greenspan, speech, the, do... | [dollar, gains, greenspan, speech, dollar, hit... |
| 2 | yukos unit buyer faces loan claim the owners o... | business | [yukos, unit, buyer, faces, loan, claim, the, ... | [yukos, unit, buyer, faces, loan, claim, owner... |
| 3 | high fuel prices hit ba profits british airway... | business | [high, fuel, prices, hit, ba, profits, british... | [high, fuel, prices, hit, ba, profits, british... |
| 4 | pernod takeover talk lifts domecq shares in uk... | business | [pernod, takeover, talk, lifts, domecq, shares... | [pernod, takeover, talk, lifts, domecq, shares... |

# Lemmatization :

Lemmatization is converting the word to its root form(Lemma).

Playing → Play

Plays → Play

Played → Play

Common root form ‘play’

am, are, is → be

Car cars, car’s, cars’ → car

Lemmatization is achieved by defining and applying following function -

```
def lemmatizing(news):
  """This function will lemmatize each word in news."""
  text = [lmt.lemmatize(word) for word in news.split()]
  return text
```

```
df['lemmatized_tokens(no_sw)'] = df['news_without_stopwords'].apply(lemmatizing)
```

```
df.head()
```

| | News_text | News_Theme | Tokens | Tokens(without stop words) | news_without_stopwords | lemmatized_tokens(no_sw) |
|---|---|---|---|---|---|---|
| 0 | ad sales boost time warner profit quarterly pr... | business | [ad, sales, boost, time, warner, profit, quart... | [ad, sales, boost, time, warner, profit, quart... | ad sales boost time warner profit quarterly pr... | [ad, sale, boost, time, warner, profit, quarte... |
| 1 | dollar gains on greenspan speech the dollar ha... | business | [dollar, gains, on, greenspan, speech, the, do... | [dollar, gains, greenspan, speech, dollar, hit... | dollar gains greenspan speech dollar hit highe... | [dollar, gain, greenspan, speech, dollar, hit,... |
| 2 | yukos unit buyer faces loan claim the owners o... | business | [yukos, unit, buyer, faces, loan, claim, the, ... | [yukos, unit, buyer, faces, loan, claim, owner... | yukos unit buyer faces loan claim owners embat... | [yukos, unit, buyer, face, loan, claim, owner,... |
| 3 | high fuel prices hit ba profits british airway... | business | [high, fuel, prices, hit, ba, profits, british... | [high, fuel, prices, hit, ba, profits, british... | high fuel prices hit ba profits british airway... | [high, fuel, price, hit, ba, profit, british, ... |
| 4 | pernod takeover talk lifts domecq shares in uk... | business | [pernod, takeover, talk, lifts, domecq, shares... | [pernod, takeover, talk, lifts, domecq, shares... | pernod takeover talk lifts domecq shares uk dr... | [pernod, takeover, talk, lift, domecq, share, ... |

# Vectorizer :

Vectorizers are used to convert text data into machine readable matrix format(vectorized format).

Two most popular vectorizers which are widely used are –

- Count Vectorizer
- TF-IDF Vectorizer

We are using Count Vectorizer for this project.

After vectorizer converts the text data into vector form, converted vectorized data will look like -

```
pd.DataFrame(count_data.toarray())
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 29705 | 29706 | 29707 | 29708 | 29709 | 29710 | 29711 | 29712 | 29713 | 29714 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **2122** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2123** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2124** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2125** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **2126** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

2127 rows × 29715 columns

# Different Model Implementations :

Different Models tried are based upon following algorithms –

- Truncated SVD (LSA/ LSI)
- LDA
- NMF

# Truncated SVD (LSA/LSI)

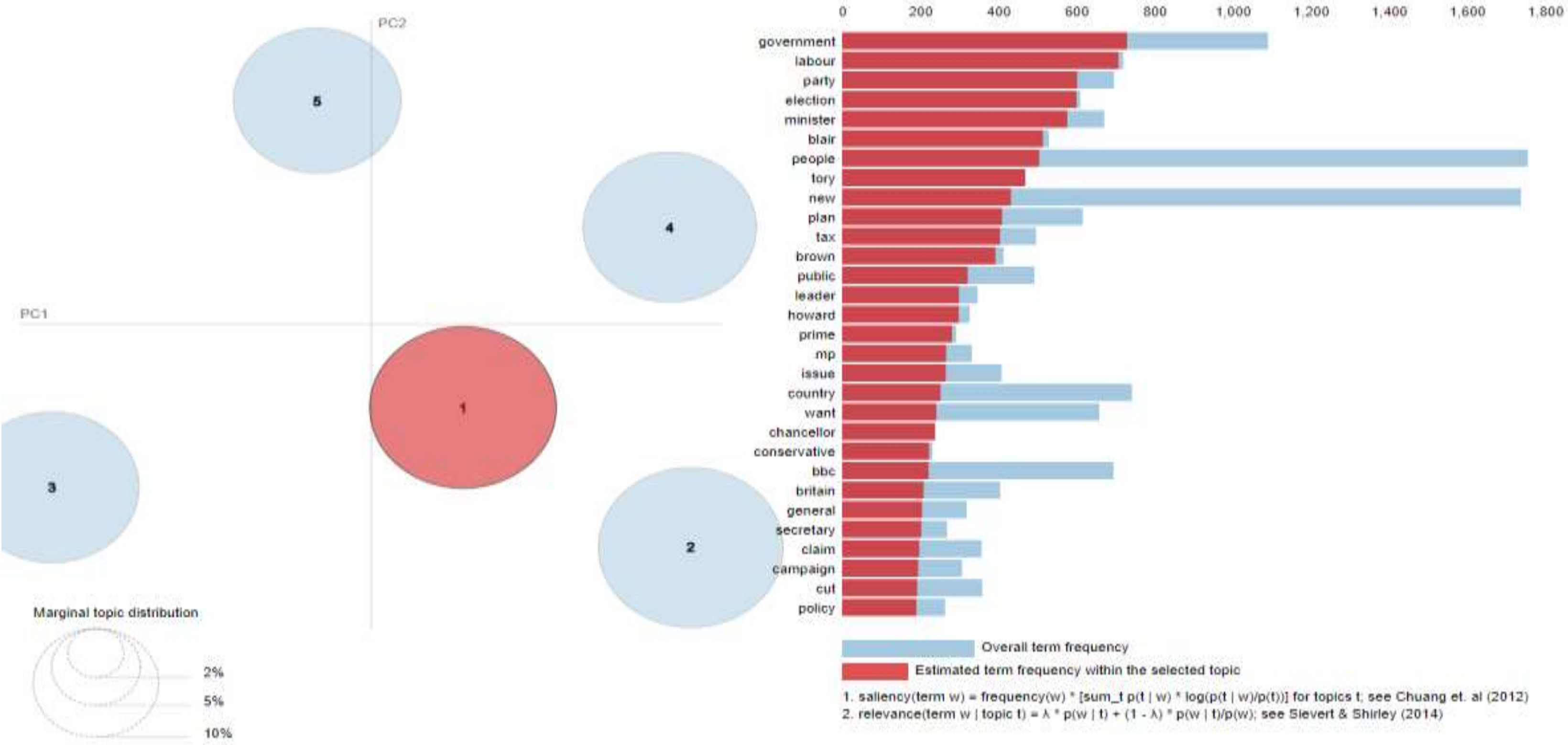
Vectorized data is fitted to SVD model -

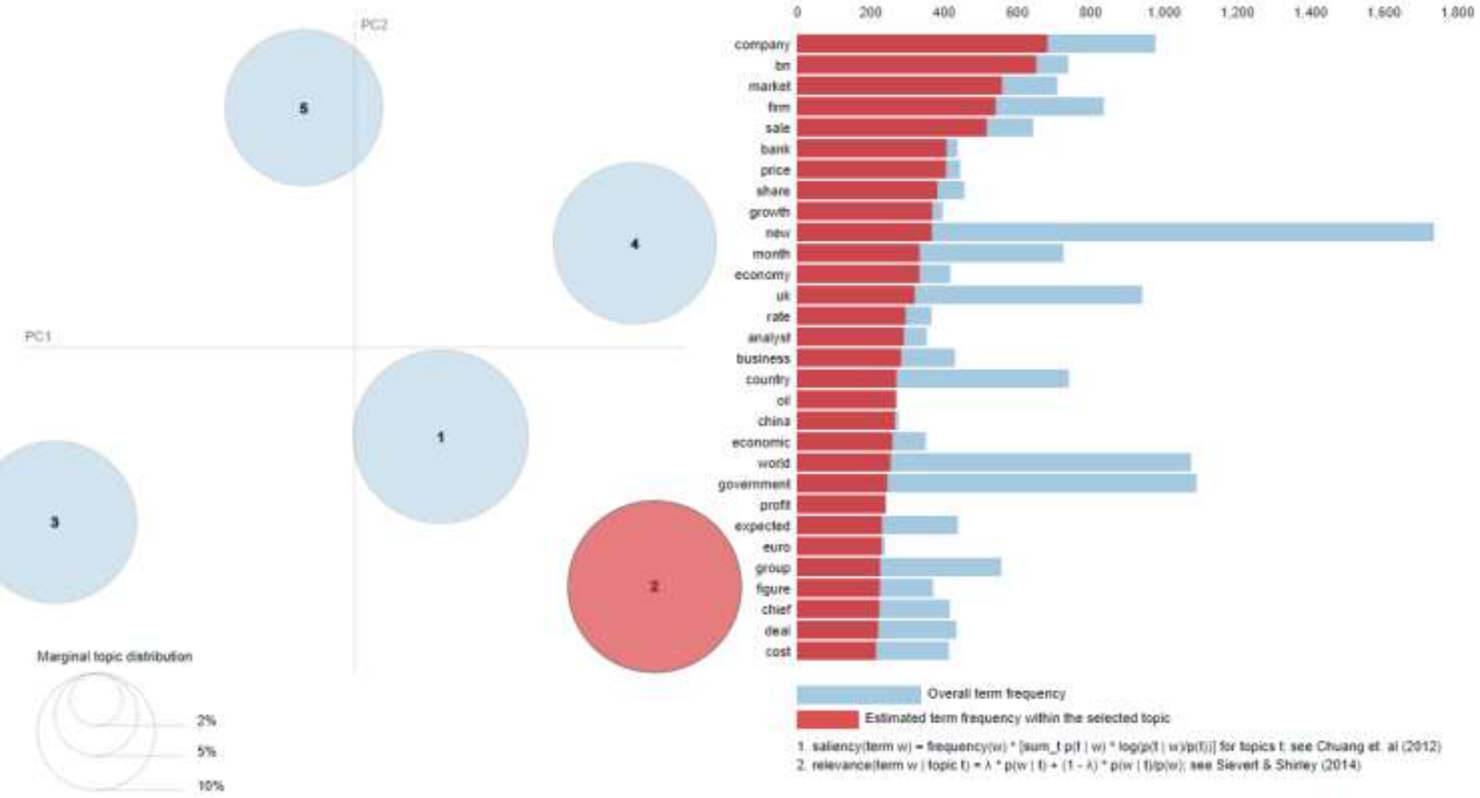
```
svd_model = svd.fit_transform(count_data)
```

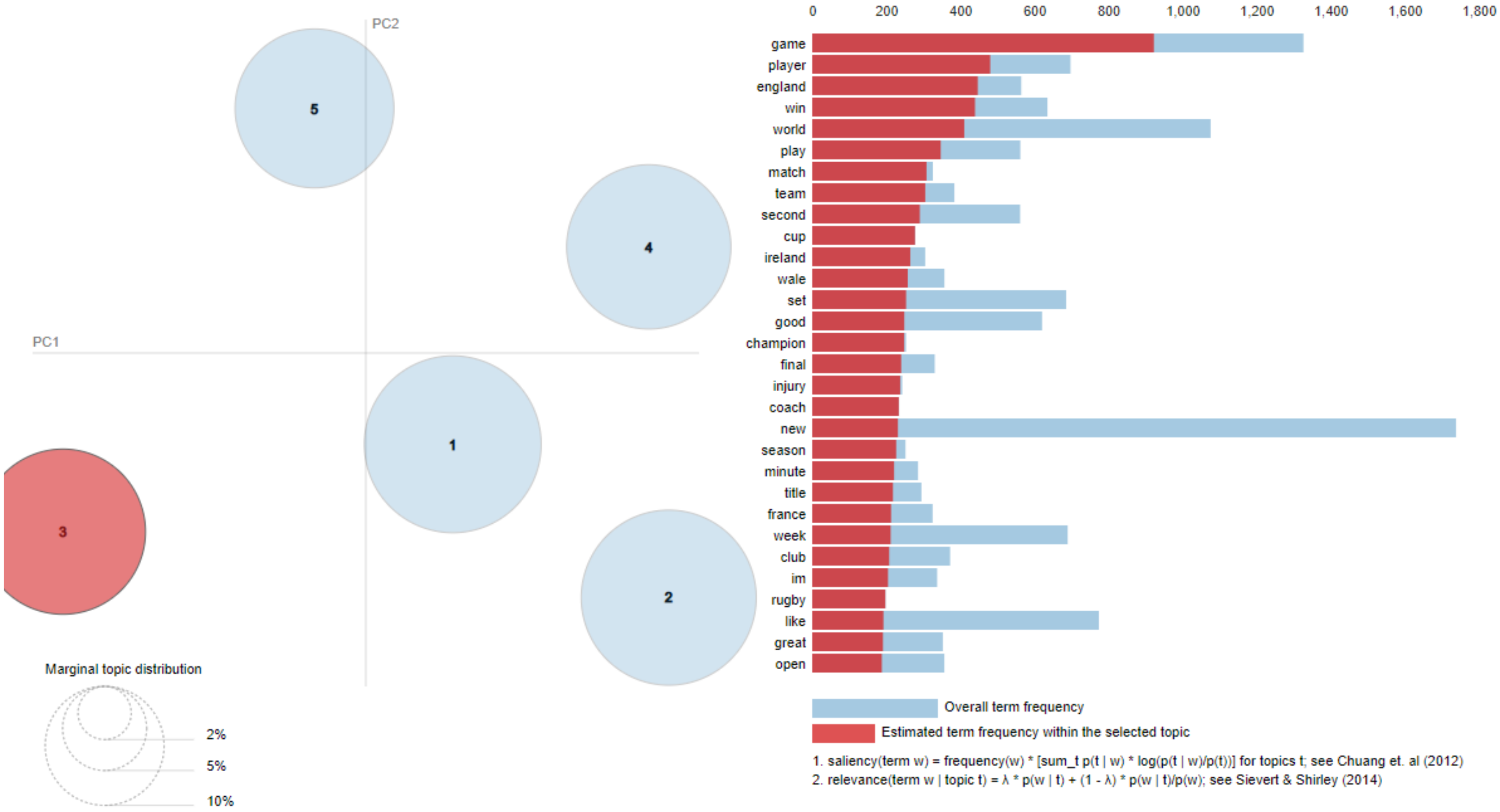
Top 20 for each topic according to SVD model -

| | Top 20 words |
|---|---|
| **Topic 1** | [people, new, game, government, music, best, like, world, uk, way, think, party, good, service, company, labour, country, mobile, song, right] |
| **Topic 2** | [best, song, music, award, angel, robbie, film, game, urban, think, prize, artist, british, dont, stone, im, williams, album, brit, joss] |
| **Topic 3** | [best, song, labour, government, party, election, blair, award, tax, tory, music, minister, british, brown, angel, public, robbie, howard, britain, plan] |
| **Topic 4** | [game, england, win, party, wale, labour, play, roddick, best, election, team, world, ireland, blair, match, point, nadal, playing, cup, zealand] |
| **Topic 5** | [music, party, people, labour, game, election, mobile, urban, phone, tory, blair, ukip, kilroysilk, like, black, howard, joss, mp, campaign, thing] |

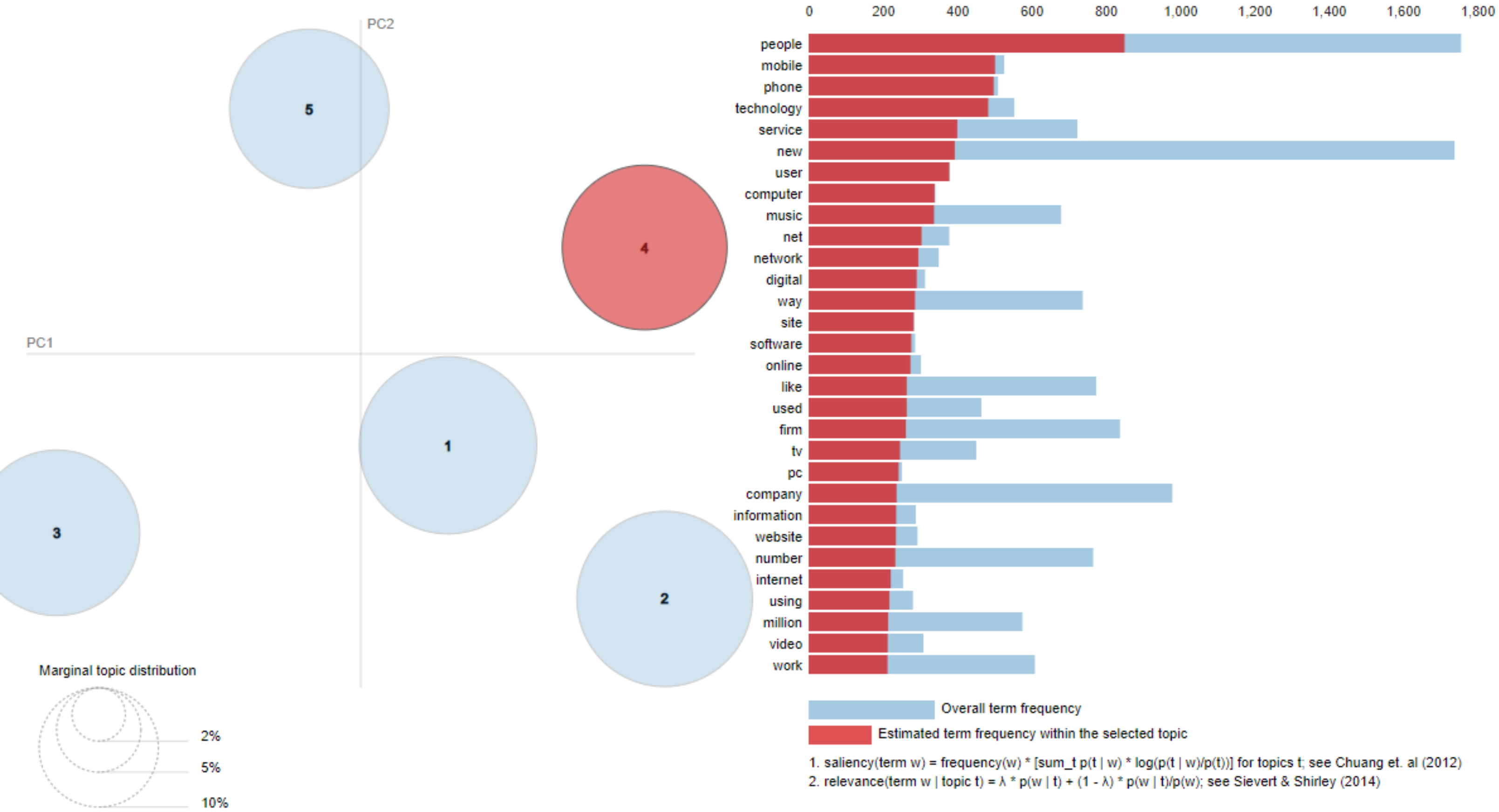
# Latent Dirichlet Allocation (LDA) :



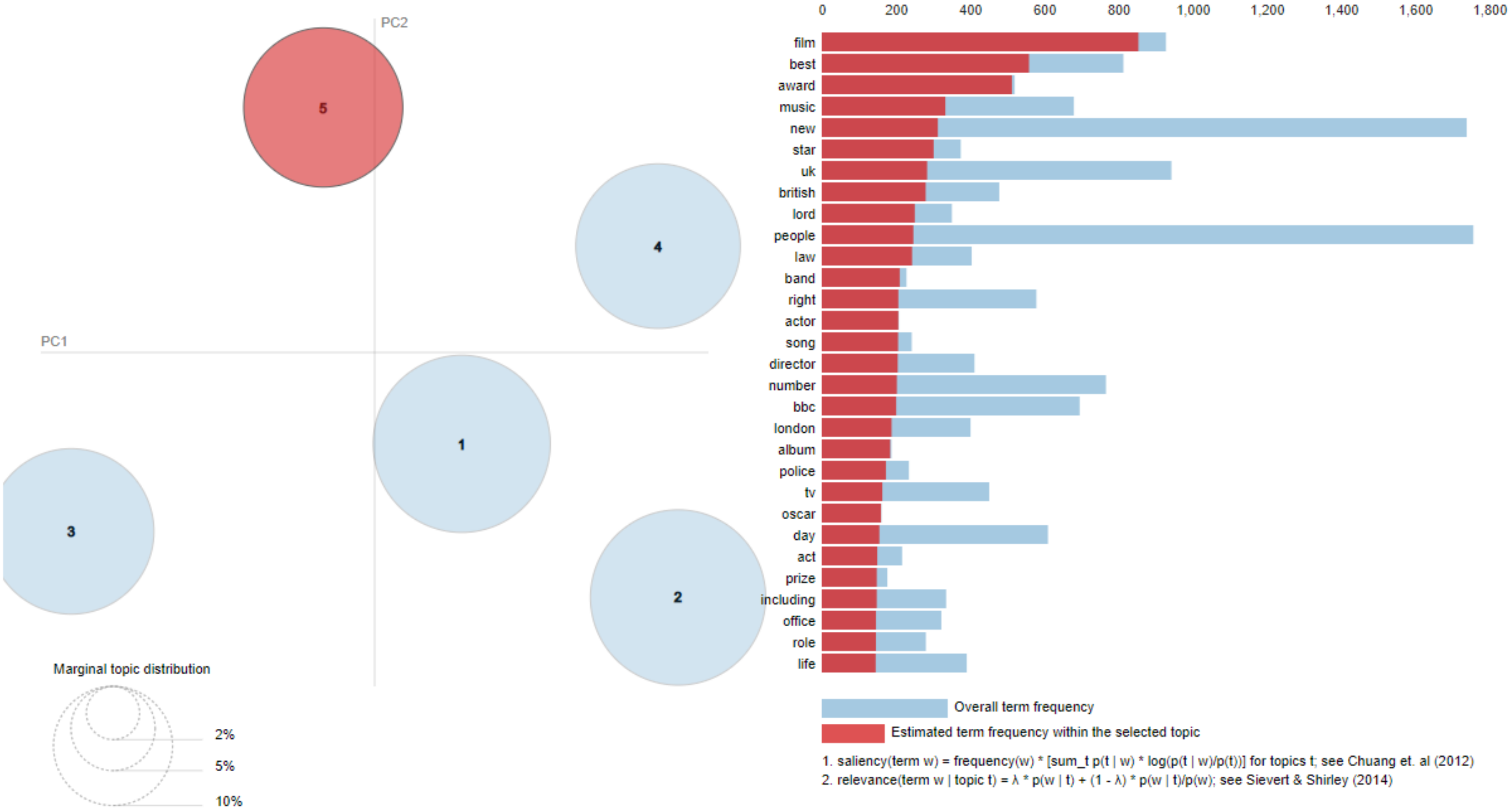

PC2
PC1
5
4
1
3
2
Marginal topic distribution
2%
5%
10%
0
200
400
600
800
1,000
1,200
1,400
1,600
1,800
company
bn
market
firm
sale
bank
price
share
growth
new
month
economy
uk
rate
analyst
business
country
oil
china
economic
world
government
profit
expected
euro
group
figure
chief
deal
cost
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)




1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)



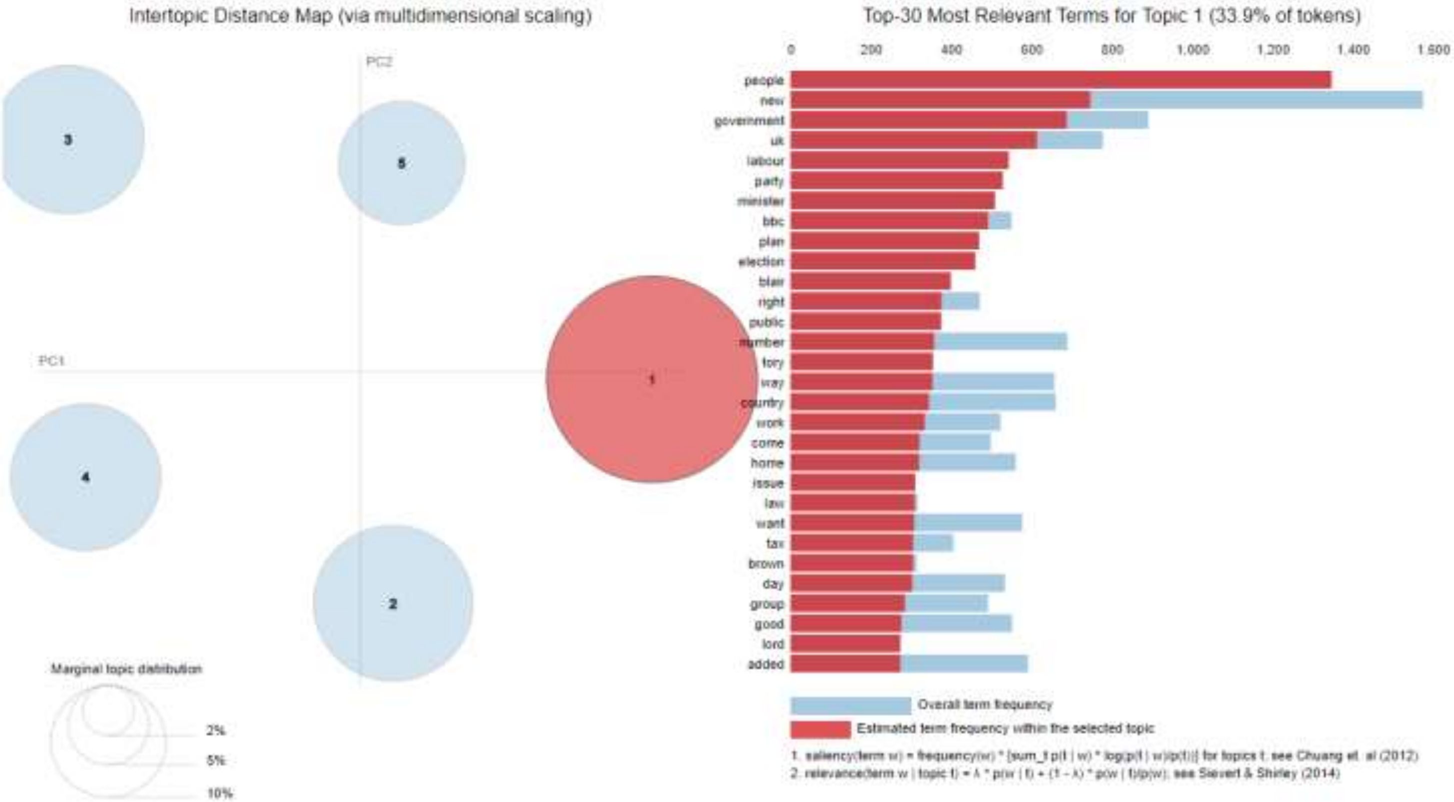

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

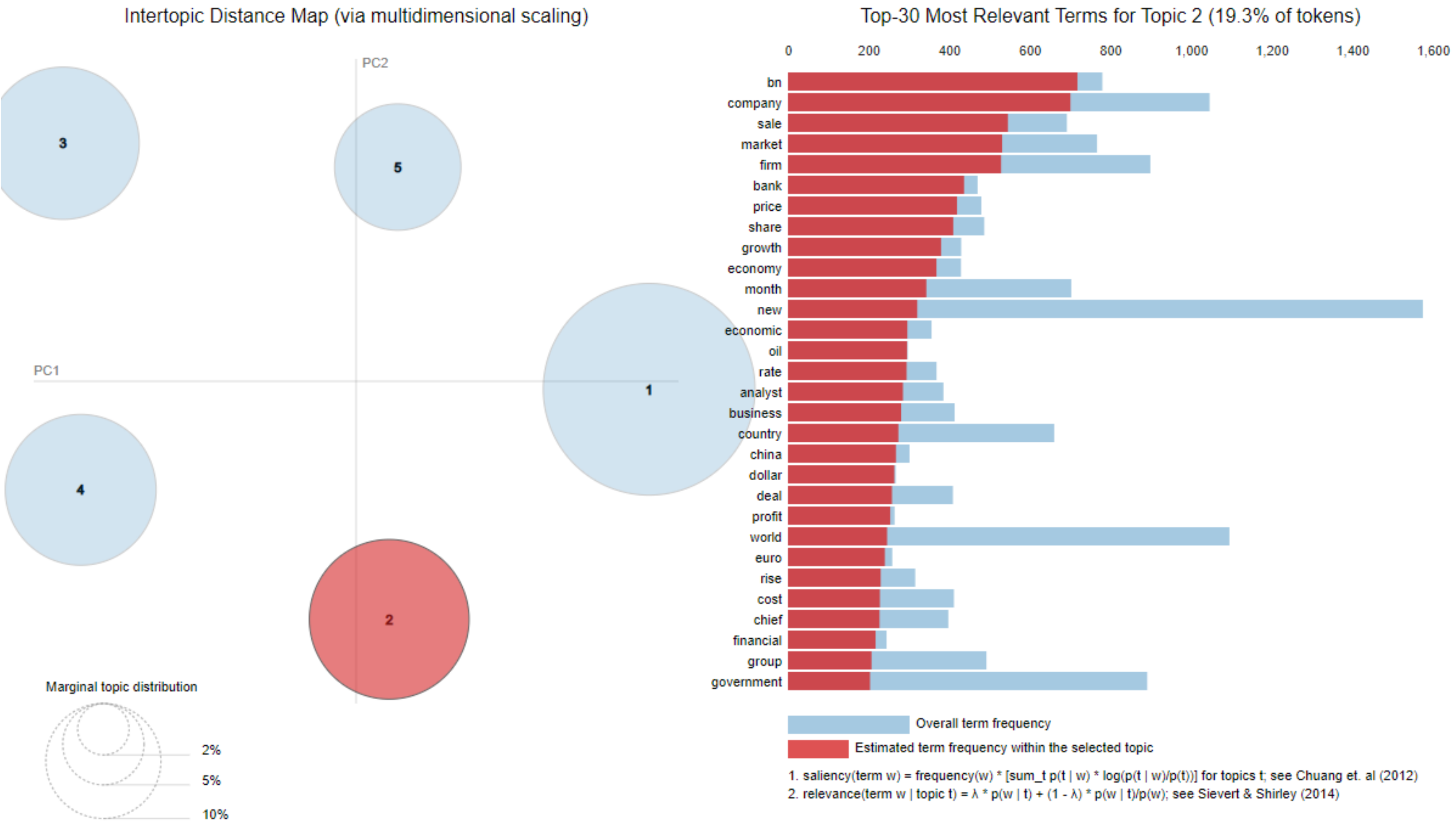


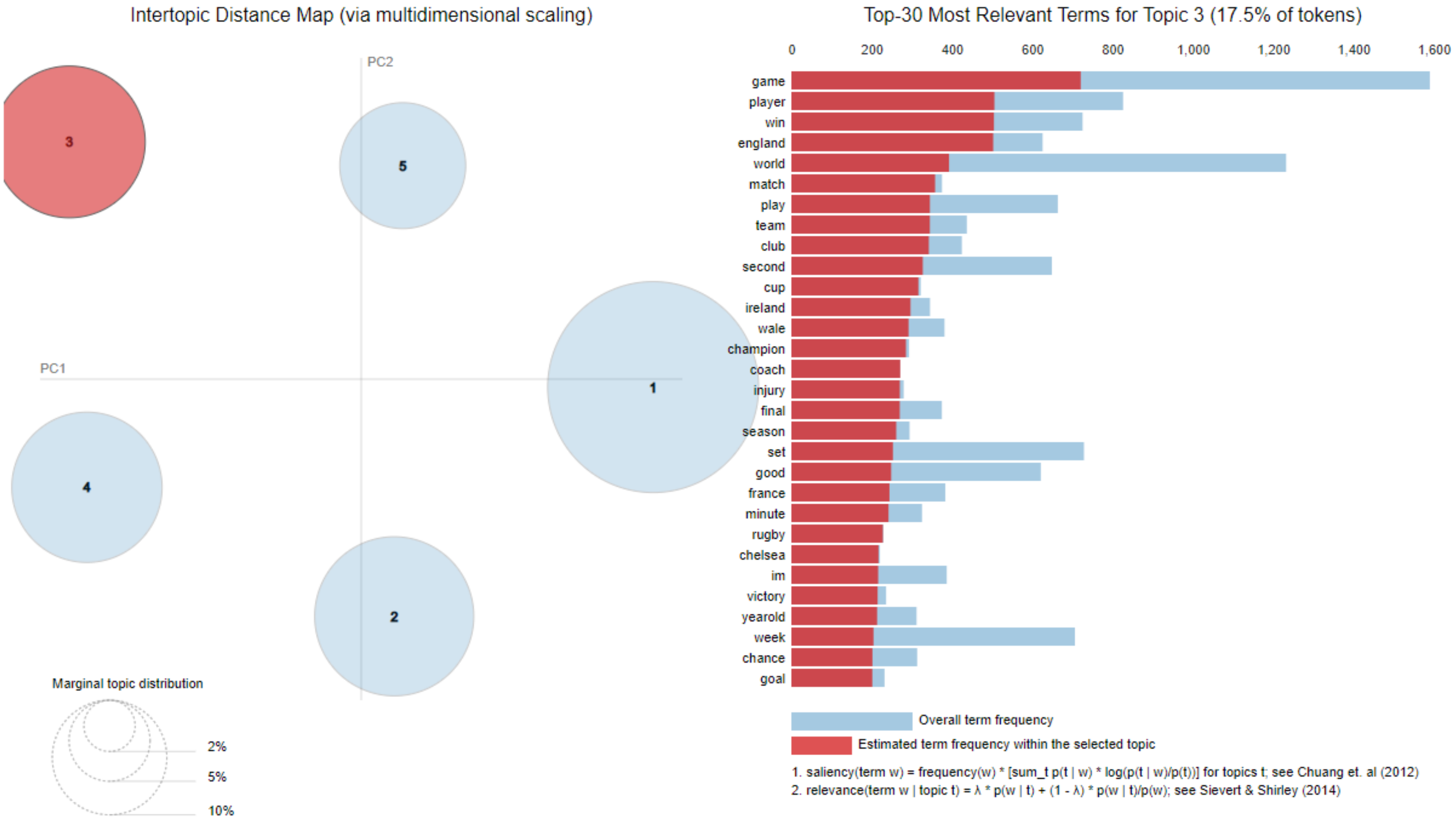

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

# Non- Negative Matrix Factorization (NMF)

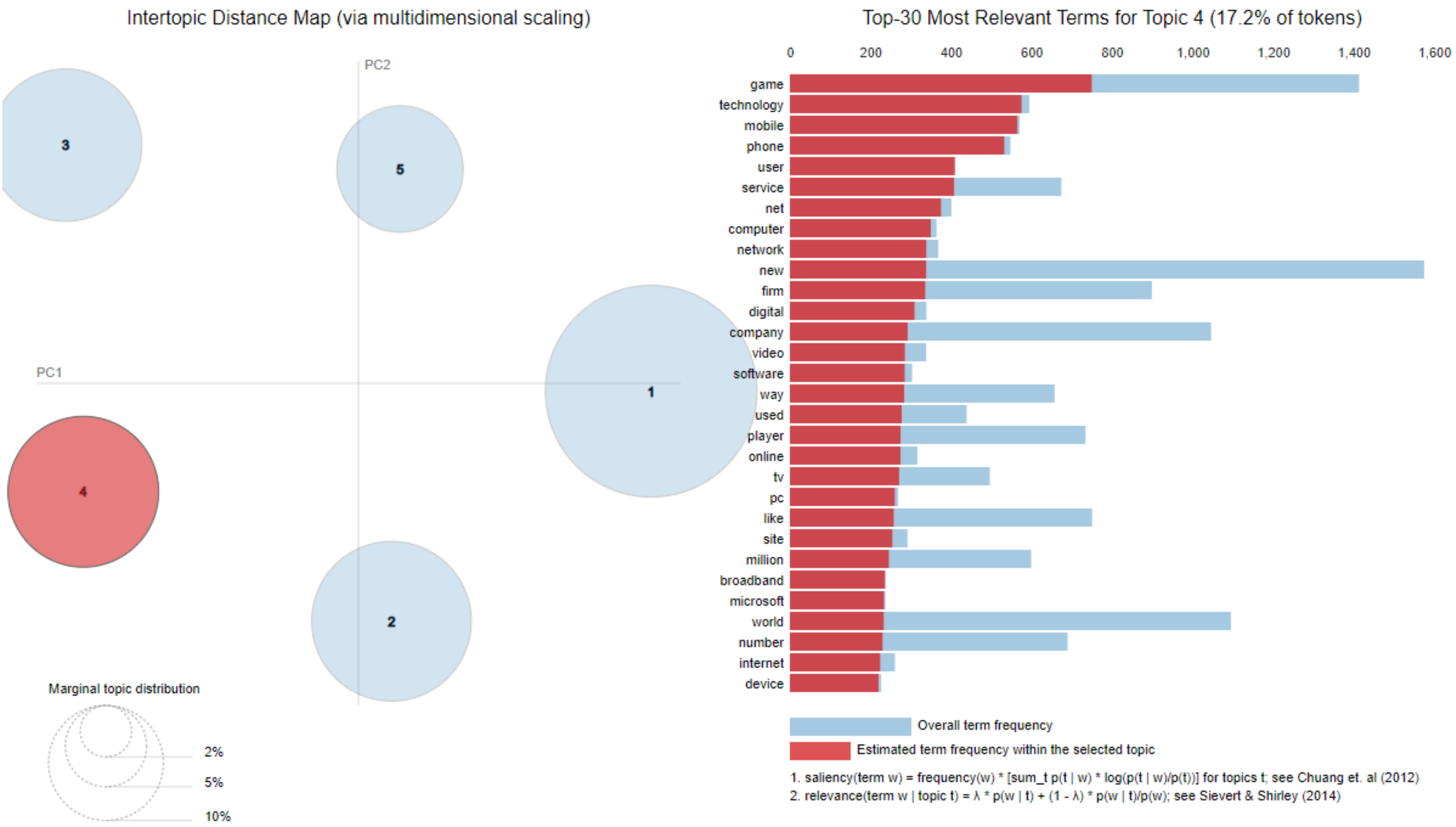
NMF - with Kullback-Leibler Divergence along with ‘tsne’ multi-dimensional spacing-

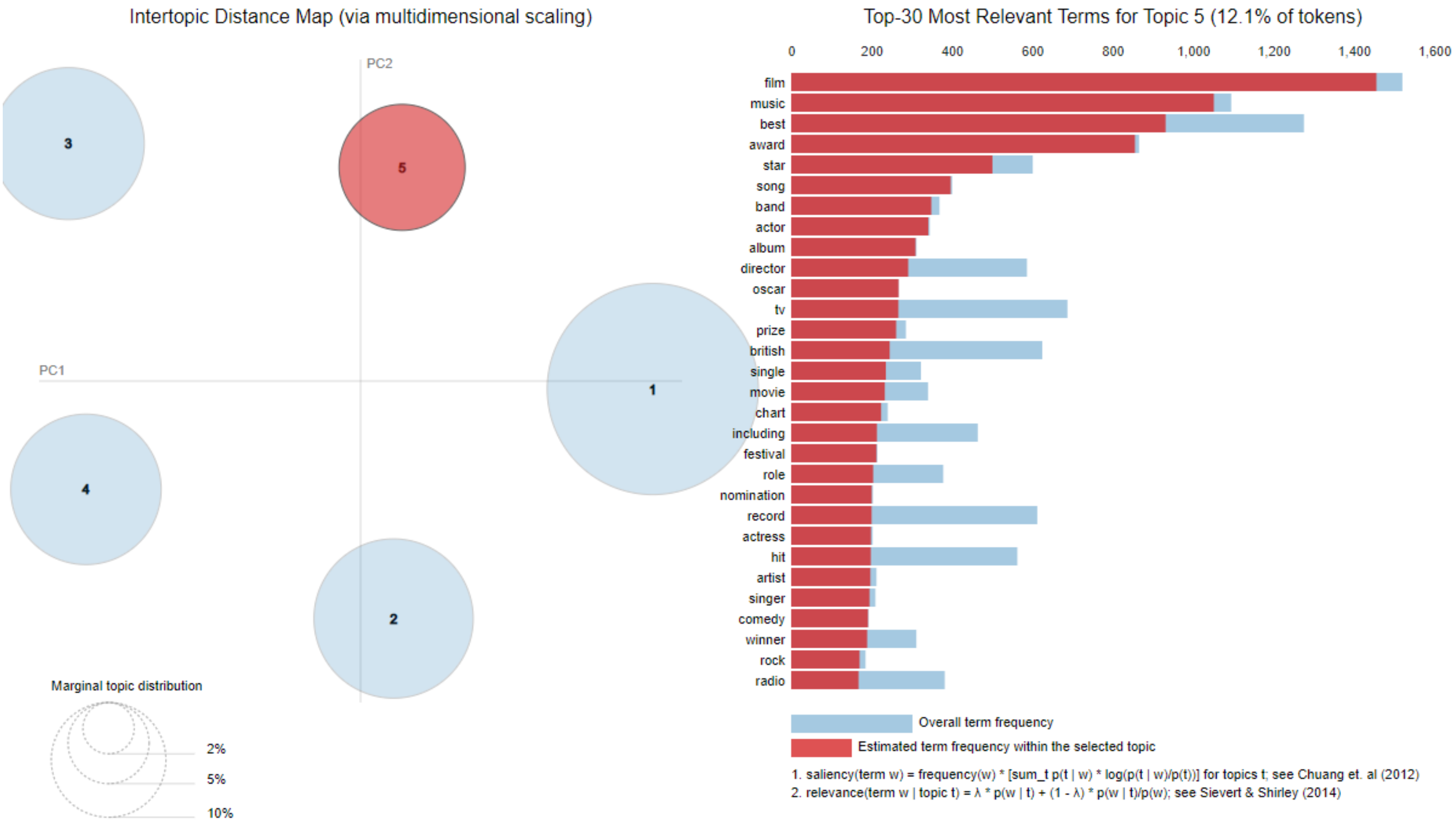



Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 2 (19.3% of tokens)
0
200
400
600
800
1,000
1,200
1,400
1,600
bn
company
sale
market
firm
bank
price
share
growth
economy
month
new
economic
oil
rate
analyst
business
country
china
dollar
deal
profit
world
euro
rise
cost
chief
financial
group
government
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)



Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
3
5
1
4
2
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 3 (17.5% of tokens)
0
200
400
600
800
1,000
1,200
1,400
1,600
game
player
win
england
world
match
play
team
club
second
cup
ireland
wale
champion
coach
injury
final
season
set
good
france
minute
rugby
chelsea
im
victory
yearold
week
chance
goal
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 4 (17.2% of tokens)
0
200
400
600
800
1,000
1,200
1,400
1,600
game
technology
mobile
phone
user
service
net
computer
network
new
firm
digital
company
video
software
way
used
player
online
tv
pc
like
site
million
broadband
microsoft
world
number
internet
device
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)



Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 5 (12.1% of tokens)
0
200
400
600
800
1,000
1,200
1,400
1,600
film
music
best
award
star
song
band
actor
album
director
oscar
tv
prize
british
single
movie
chart
including
festival
role
nomination
record
actress
hit
artist
singer
comedy
winner
rock
radio
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

# Final Conclusions :

From above , it can be concluded that-

1. The words given by SVD were mixed in nature & didn’t represent any topic and hence the worst model.

2. The words given by LDA correctly gave us the idea of topic to which the words belong and hence good model.

3. But, the bag of words given by NMF were best and precisely represented the topics which we already knew.

Hence, NMF algorithm is best suited for this problem.