Business Report Analysis

Text Mining Lab

Chu-I Chao, Xiaoqi Ma

Table of Contents

1. Introduction
2. Problem Definition
3. Our Approaches
    a. TF-IDF
    b. LSI
    c. LDA
4. Results
5. Summary

Introduction

Business reports tell us….

- Current situations
- Concerns and risks of an ongoing policy or program
- Competitors...

=> Might be long and take lots of time to read

=> How to extract useful information from them?

Problem Definition

Retrieve the following information from the given business reports

- Keywords, for document retrieval using words search
- Topics, for categorizing and retrieving documents

Approach for Keywords Mining : tf-idf

Goal: score the importance of a word in a report

=> regarded as ***keywords*** of a report

Idea: consider

1. how often does a word appear in a report => "term frequency ***tf*** "
2. how rare is a word across all reports => "inverse document frequency ***idf***"

=> $\boxed{\textbf{importance score = tf * idf}}$

Keywords from tf-idf

	PUMA-2015	PUMA-2016
0	direktor	verwaltungsrat
1	verwaltungsrat	direktor
2	geschäftsführen	verwaltungsrats
3	schuh	schuh
4	verwaltungsrats	rohertragsmarge

Administration

	Adidas-2015	Adidas-2016
0	reebokccm	schuh
1	rockport	konsument
2	schuh	rockport
3	marketinginvest	athlet
4	konsument	performancebonu

Product, Customer

Keywords from tf-idf

	Daimler-2015	Daimler-2016	BMW-2015	BMW-2016
0	fahrzeug	fahrzeug	automobile	automobile
1	daimler	eklasse	vorzugsaktien	vorzugsaktien
2	sprinter	athlon	fahrzeug	fahrzeug
3	daimlerkonzerns	vans	finanzdienstlei	motorrad
4	toll	mercedesbenz	werken	finanzdienstlei

Product

Finance

We can also use tf-idf for searching relevant documents

HOWEVER, if some related words, except the search word itself, appear in a report….

query=”vehicle”

	tf-idf value
BMW-2016-Q3	0
BMW-2015-Q3	0.00245885
BMW-2016-Q1	0
BMW-2016-Q2	0
Draeger-2013-Q3	0
Draeger-2014-Q3	0
Deutsche_Post-2013-Q1	0
Deutsche_Post-2011-Q3	0

Latent Semantic Indexing (LSI)

Goal: find truly relevant reports regarding to the query

Idea: Use SVD to reduce the dimension of tf-idf matrix

=> can be imagined as a compression of words with similar meaning

LSI

{<vehicle>, <automobile>, <dog>} -----> {<0.6 * vehicle + 0.4 * automobile>, <dog>}

The Power of LSI

```
q = 'vehicle'
getRelatedDocuments(q, u, s, v, word2index, index2document)
```

	relevance score	tf-idf value
BMW-2016-Q3	0.752678	0
BMW-2015-Q3	0.715746	0.00245885
BMW-2016-Q1	0.681478	0
BMW-2016-Q2	0.592523	0
Draeger-2013-Q3	0.225724	0
Draeger-2014-Q3	0.210355	0
Deutsche_Post-2013-Q1	0.153236	0
Deutsche_Post-2011-Q3	0.145989	0

```
q = 'wo kann ich ein schuh order eine socke kaufen '
getRelatedDocuments(q, u, s, v, word2index, index2document)
```

```
keywords "wo" not found
keywords "kann" not found
keywords "ich" not found
keywords "ein" not found
keywords "order" not found
keywords "eine" not found
```

	relevance score
Adidas-2016	0.936709
Adidas-2015	0.932854
PUMA-2015	0.371554
PUMA-2016	0.300578
Bosch-2016	-0.00329964
Allianz-2016	-0.0115815
Bosch-2015	-0.0124566
Allianz-2015	-0.0131769
BVB-2015	-0.0231023
Deutsche_Post-2016	-0.0248835
BMW-2015	-0.0297912
BMW-2016	-0.0331917

Approach for Topic Modeling: LDA

Idea: view a report as a mixture of various topics

Goal: Discover the hidden topics from all reports in an ***unsupervised*** way

=> no labelling needed

Somehow, it means output topics might be ***not interpretable*** to human

Also, how to decide the number of topics is a problem...

LDA Experiment

Training Set: 123 bank quarterly reports

Steps of our approach:

1. Run LDA from 5 to 19 topics and get 15 models
2. Evaluate the LDA models by computing coherence score and store the best one
3. Repeat step 1 and 2 for several times

Coherence Score

Goal : evaluate the LDA model
"How well are output topics understandable to human ?"

Idea : if words always show up together

=> they likely belong to the same topic

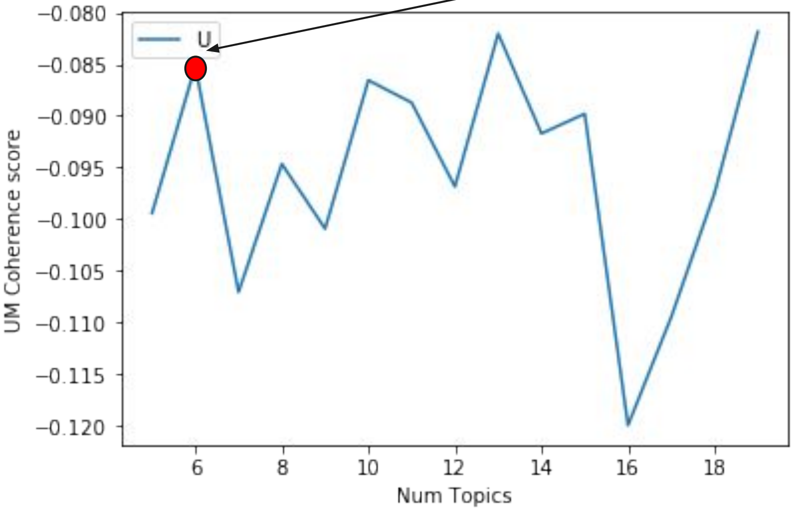
2 measures in topic coherence: UMass and C_V (consider words ordering or not)

https://nanonaren.wordpress.com/2015/06/17/topic-coherence

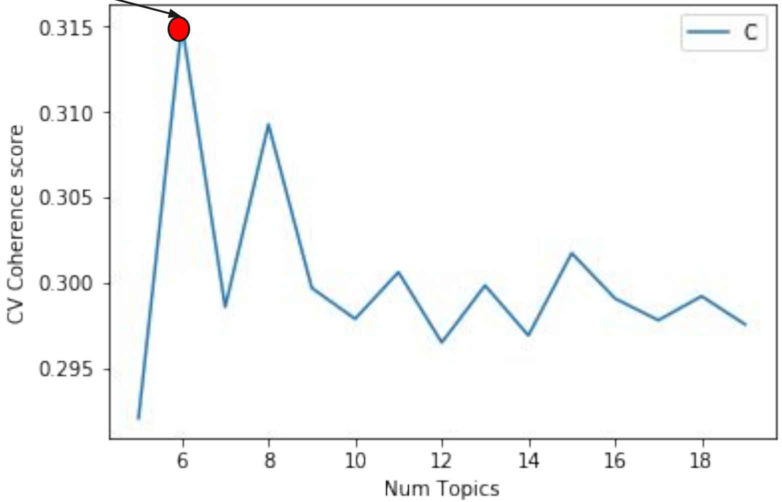
Experiment Result

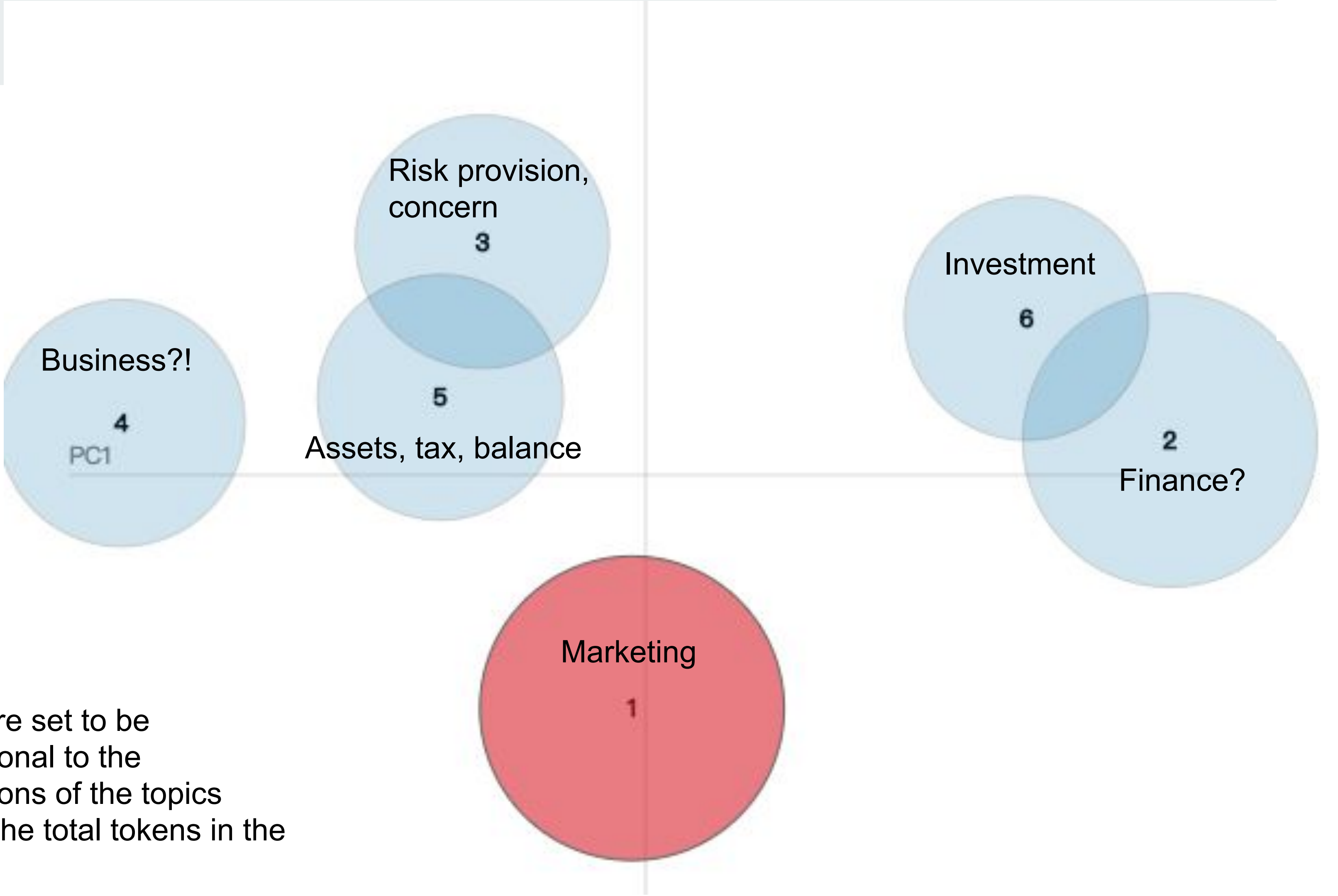
6 topics might be a good choice

UMass



C_V





areas are set to be proportional to the proportions of the topics across the total tokens in the corpus

Summary

1. Tf-idf helps us to find keywords of a report
2. LSI groups words with similar meaning => relevant document retrieval
3. LDA is easy to train, but hard to evaluate


vorjahr
direktor
mitarbeiter
höhe
produkt
ergebnis
umsatz
rahmen
markt
vergütung
zulieferer
programm
dezember
investition
bereich
textilien
anteil
gewinnen
zielen
schuh
welt
region
kategorie
leistung
forderung
strategie
aufwand
wesentliche
teilen
fokus
anzahl
fair
ertrag
anstieg
ziffer
option
basis
enden
team
steuer
mark
ifrs
geschäft
aktie
partner
land
sport
bezug
bericht
konzern
kodex
kapital
segment
prüfung
gesellschaft
risiko
zeitwert
angeben
design
vermögen
bewertung
standard
frau
posten
januar
erfolgen
wachstum
maßnahme
umsetzung
satzung
buchwert
faster
verlust
ansatz
material
beteiligung
sicherheit
aktionär
laufzeit
fabrik
emission
monat
firmenwert
betragen
prognose
differenz
änderung
darstellung
auswirkung
management
annehmen
thema
ebene
marke
einführung
gesetz
schulden
stück
wert
bild
system
standort
kunde
formen
vertrag
puma
gmbh
finanz
modell
vorschrift
steigerung

```
while (best_UM_round != best_CV_round):
  for i = 1 to 3:
    model_list = get_ldamodel_list(start=5, limit=20, step=1)

    UM_coherence_values = compute_UM_coherence_values(model_list)
    CV_coherence_values = compute_CV_coherence_values(model_list)

    max_UM_value = np.max(UM_coherence_values)
    if max_UM_value > best_UM_result:
      best_UM_result = max_UM_value
      best_UM_model_list = model_list.copy()
      best_UM_round = i

    max_CV_value = max(CV_coherence_values)
    if max_CV_value > best_CV_result:
      best_CV_result = max_CV_value
      best_CV_model_list = model_list.copy()
      best_CV_round = i
```