Topic Modelling: CBA Glassdoor reviews

Obaid Pervaiz Gill

## Business Question

* Why do people choose to work at CBA?

## Approach

* The data used to answer this question was obtained from glassdoor.com.au. In particular the ‘pros’ of working at CBA were chosen. Only the first page of reviews from glassdoor was scrapped.
* Although there are many methods which could be used for Topic Modelling. Latent Dirichlet Allocation was used here through Gensim, as it was recommended by Product Analytics leads at Atlassian in one of his data science talks and secondly there are plenty of use cases for this on the internet.
* The language used for data and modelling was Python and all the results were tabulated and visualized using Excel.

## Table and Outputs

* Although, a number of topics could be obtained but since we only had ten data points, only 3 topics was extracted. The topics are tabulated and visualized below with the likelihood of terms within a topic as well. The x-axis shows the terms appearing in each topic (note that these terms are stemmed) and y-axis shows the likelihood of each term within a topic.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Topic 1** | **Likelihood** |  | **Topic2** | **Likelihood** |  | **Topic 3** | **Likelihood** |
| work | 0.062 |  | work | 0.033 |  | team | 0.057 |
| support | 0.062 |  | great | 0.033 |  | vast | 0.033 |
| cultur | 0.044 |  | good | 0.032 |  | dynam | 0.033 |
| environ | 0.044 |  | cultur | 0.032 |  | inform | 0.033 |
| flexible | 0.044 |  | well | 0.019 |  | manag | 0.032 |
| great | 0.027 |  | corpor | 0.019 |  | custom | 0.032 |
| good | 0.026 |  | balanc | 0.019 |  | bank | 0.032 |
| love | 0.025 |  | opportun | 0.019 |  | leader | 0.031 |
| valu | 0.025 |  | within | 0.019 |  | compani | 0.031 |
| peopl | 0.025 |  | life | 0.019 |  | technolog | 0.031 |

## Code

# -\*- coding: utf-8 -\*-

"""

Created on Tue Oct 25 20:49:02 2016

@author: ogill

"""

**## Step 1 - Downloading the data ##**

***#scrapping data from "https://www.glassdoor.com.au/Reviews/Commonwealth-Bank-of-Australia-Reviews-E7922.htm"***

d1 = 'Great work environment and fits in well with my family'

d2 = 'Good corporate culture that cares for the well being of its people'

d3 = 'Dynamic team, vast information management'

d4 = 'Good exposure to Banking domain, enterprise IT solutions. Mature SOA architecture capability in place.'

d5 = 'Has a number of ecosystems and culture climates. I was lucky in investment team to play hard but was rewarded well.'

d6 = 'Great culture and values. Love the work environment and flexibility'

d7 = 'Great company, technology, benefits and team. Customer leader in many areas including online banking :-).'

d8 = 'Friendly and supportive environment. Where you can find diferent and challenging projects to work on. Working in a flexible is possible and supported. At least in support functions.'

d9 = 'Great reputation and plenty of opportunities for training and diversification. Promotes a positive ethical culture within. Good work life balance for a corporate.'

d10 = 'Good progression and training options. Good social life. Good location of offices darling Park'

***#appending all the scrapped data points into an array***

data = []

for i in range(1,10):

print 'd' + str(i)

data.append(eval('d' + str(i)))

**## Step 2 - Importing the packages and initiating the classes for data preparation ##**

***#importing relevant libraries***

from nltk.tokenize import RegexpTokenizer

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

from gensim import corpora, models

import gensim

#Initiating the word grabbing class

tokenizer = RegexpTokenizer('\w+')

***#Initiating the instance to remove english stop words***

stop = set(stopwords.words('english'))

***#Initiating the word stemming class (to make long words shorter)***

stem = PorterStemmer()

**## Step 3 - Preparing the data for Topic Modelling ##**

***#lowering and splitting words***

clean\_data = [i.lower() for i in data]

clean\_data = [tokenizer.tokenize(i) for i in clean\_data]

***#removing stop words and making the remaing words shorter***

for i in range(1,len(clean\_data)):

clean\_data[i] = [j for j in clean\_data[i] if not j in stop]

clean\_data[i] = [stem.stem(j) for j in clean\_data[i]]

**## Step 4 - Piping the data into an LDA for Topic Modelling ##**

***#extracting the terms dictionary from the clean\_data***

clean\_data\_dict = corpora.Dictionary(clean\_data)

***#transforming the clean data into a term matrix***

clean\_data\_corpus = [clean\_data\_dict.doc2bow(i) for i in clean\_data]

***#fitting a Latent Dirichlet Allocation model for topic modelling***

topic\_model = gensim.models.ldamodel.LdaModel(clean\_data\_corpus, num\_topics=3, id2word = clean\_data\_dict, passes=3)

print(topic\_model.show\_topics())