ADVANCED MACHINE LEARNING AND DATA MINING FINAL PROJECT REPORT

Aspect-Based Sentimental Analysis On Customer Reviews

Nguyen Hai Dang, Le Tran Ngoc Minh, Le Thi Trang

June 26th, 2019

Contents

List of Figures

Chapter 1

Introduction and data description

1.1 Define problem

With the development of e-commerce platforms, customers are now being able to give feedback about goods and service they purchased. These data are generated continuously and with increasing volume. By examining the review from users, companies can get knowledge about the topics that is being discussed as well as whether they are positive or negative. These information provide useful insights to understand their customers better and improve service quality.

Our project aims at doing aspect and sentiment analysis for customer reviews at the same time. It includes two tasks:

1. Identify the topic that is mentioned in each customer feedback.
2. Identify the polarity of each comments (whether it is positive or negative)

1.2 Data

Our target language for this project is Vietnamese and the data is crawled from the website **foody.vn**. Foody is a platform that connects restaurants, food and coffee shop owners and consumers. It provides a convenient way for consumers to look for the information about eating and drinking places as well as see the reviews from other customers about a place before deciding to order food or drinks.

We have collected two data sets. The first one contains 30,000 reviews with label positive or negative, each are written in a separated text file. The second one contains 174,437 reviews from different shops without label and are saved in a pickle file. The source of our data sets is website: *https://forum.machinelearningcoban.com*

Chapter 2

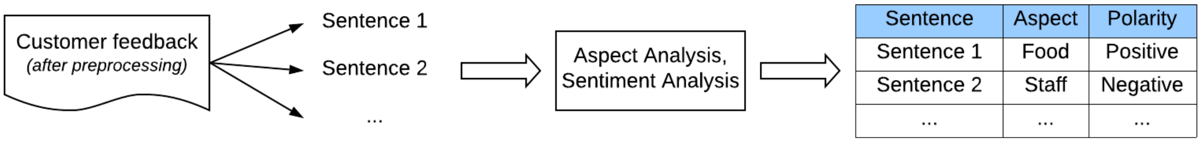
Solution

Firstly, the data is preprocessed. Since a customer feedback usually contains many sentences. We assume that each sentence represents an aspect. We then split the customer feedback into sentences.

Secondly, we separate our problem into two tasks:

- Aspect analysis: Identify the aspect which are mentioned in each sentence.
- Sentiment analysis: Identify the polarity of each sentence (whether it is positive or negative).

Following is our system work flow:



Sentence	Aspect	Polarity
Sentence 1	Food	Positive
Sentence 2	Staff	Negative
...	...	...

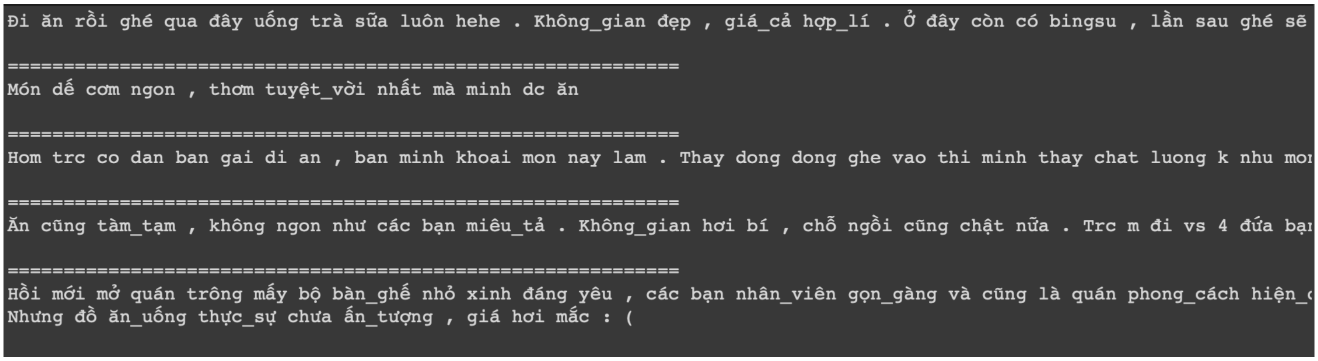
Figure 2.1: System work flow

Chapter 3

Implementation

3.1 Preprocessing and Analysis

Prepossessing on Vietnamese review is a challenging part because of the informal, icons, spelling mistakes, bi-language, false decoding sentences and no spelling mark on sentences.



```
Đi ăn rồi ghé qua đây uống trà sữa luôn hehe . Không_gian đẹp , giá_cả hợp_lí . Ở đây còn có bingsu , lần sau ghé sẽ

============================================================
Món dế cơm ngon , thơm tuyệt_vời nhất mà minh dc ăn

============================================================
Hom trc co dan ban gai di an , ban minh khoai mon nay lam . Thay dong dong ghe vao thi minh thay chat luong k nhu mo

============================================================
Ăn cũng tàm_tạm , không ngon như các bạn miêu_tả . Không_gian hơi bí , chỗ ngồi cũng chật nữa . Trc m đi vs 4 đứa bạ

============================================================
Hồi mới mở quán trông mấy bộ bàn_ghế nhỏ xinh đáng yêu , các bạn nhân_viên gọn_gàng và cũng là quán phong_cách hiện_
Nhưng đồ ăn_uống thực_sự chưa ấn_tượng , giá hơi mắc : (
```

Figure 3.1: Typical Reviews on Foody

Because checking spelling mark is the difficult task, and also, their occurrences are bellow 5%, so we remove them all.

One property of Vietnamese is they using a lot of phrase of words. For examples, ”không gian”, ,”không ngon”, ”trà đào”. The third example could also read as two separate words without losing the meaning. The seconds could be sentimentally classified by machine learning technique. However, the first could not be understood as a noun phrase and may lead to misleading in the training phase.

We use the Phrase object of gensim library to combine those words into one single word by checking how frequency they occur together: ”không_gian”, ”trà_đào”.

The final step is removing stop words. Because the Vietnamese stop words recommended on the internet in general and do not capture all the unimportant words, we add and remove some words in stop words corpus by counting the occurrence of them in data set.

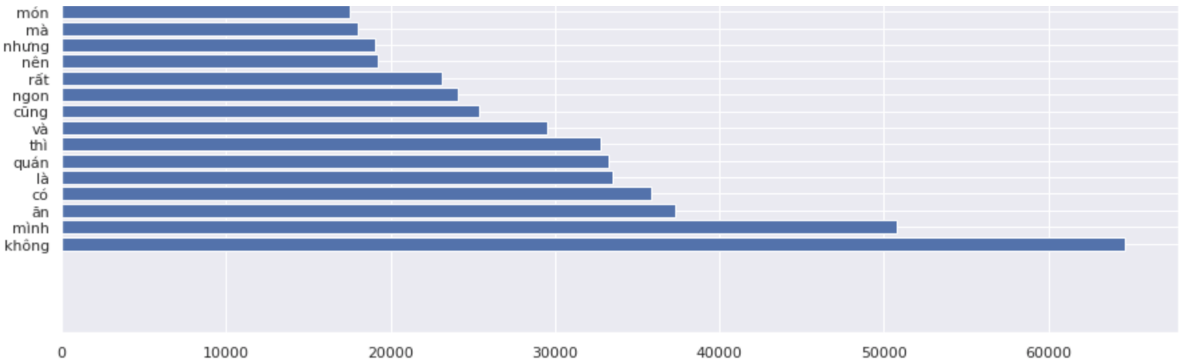


Figure 3.2: Top Occurrence Words

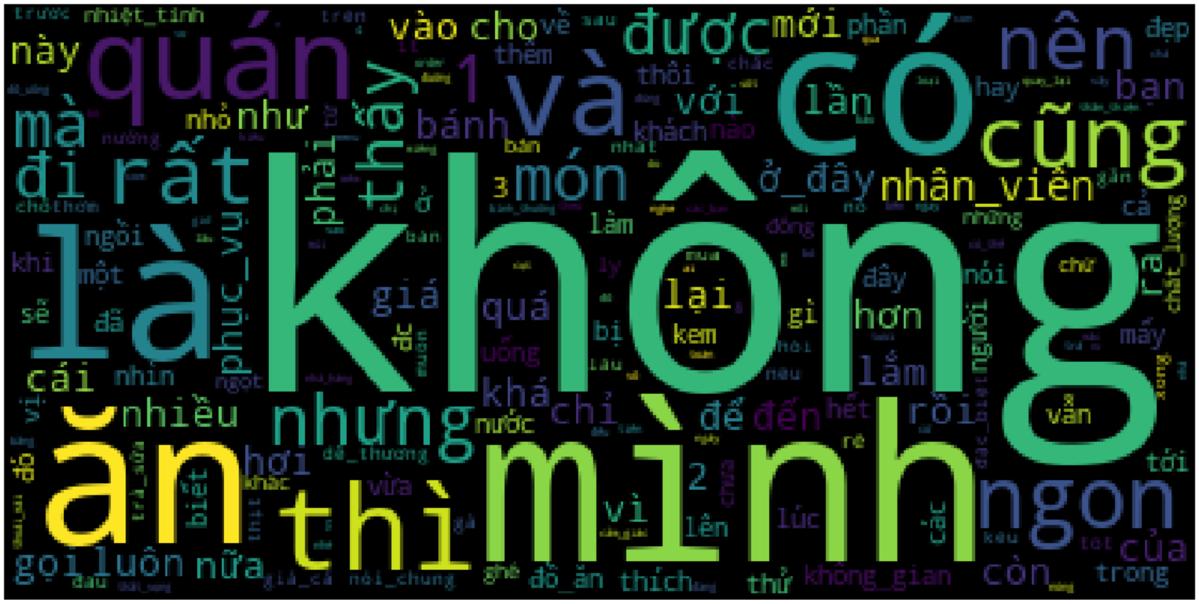


Figure 3.3: Word Cloud

3.2 Aspect Analysis

In this part, we identify the topics that are mentioned in each review. Since the topics are not given, we use unsupervised methods to do this task. The methods we choose are:
1. Latent Dirichlet Allocation (LDA)
2. Word2Vec
Each method will be described in details below.

3.2.1 LDA

LDA is a generative model introduced by (Blei et al., 2003) that quickly gained popularity because it is an unsupervised, flexible and extensible technique to model documents. LDA models documents as multinomial distributions of so-called topics. Topics are multinomial distributions of words over a fixed vocabulary. Topics can be interpreted as the categories from which each document is built up, and they can be used for several kinds of tasks, like dimensionality reduction or unsupervised clustering.

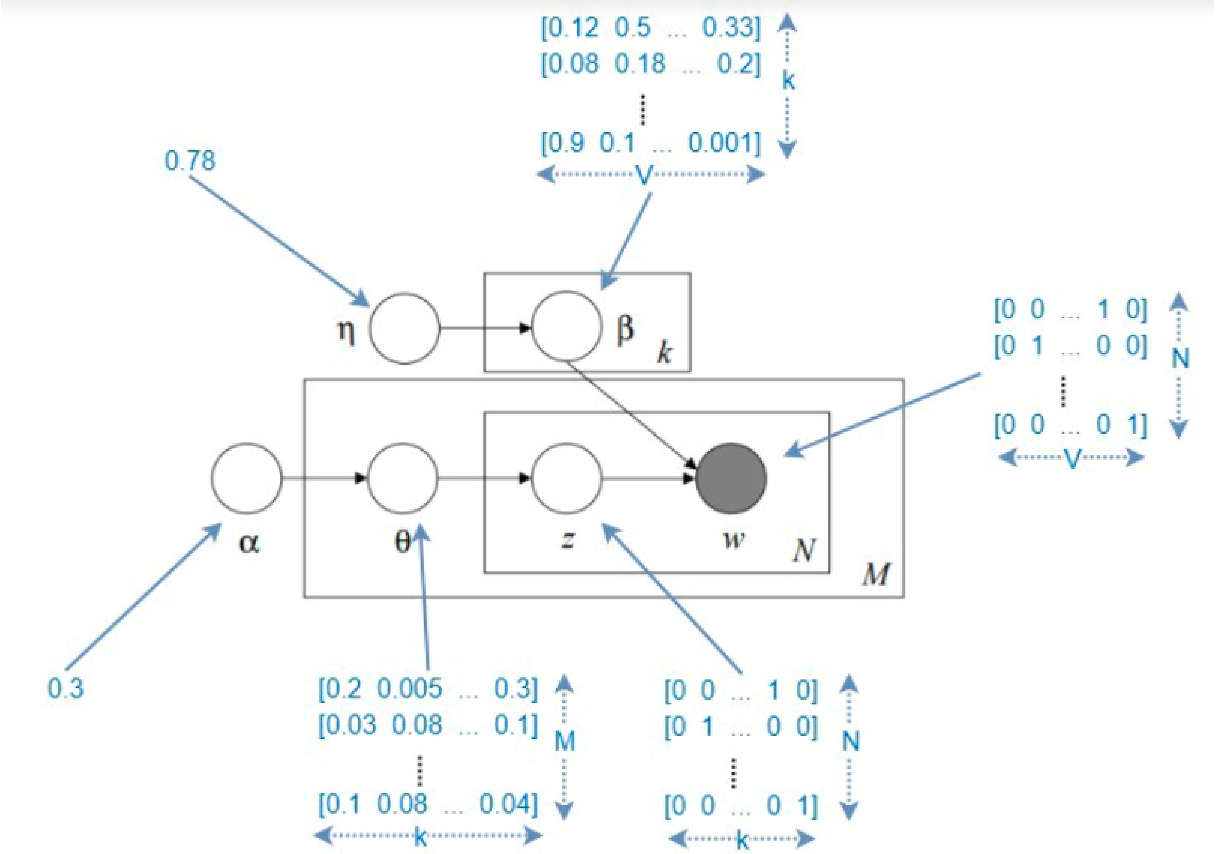


Figure 3.4: Graphical model of the LDA

With the notations as follow

- k: Number of topics a document belongs to (a fixed number)
- V: Size of the vocabulary
- M: Number of documents
- N: Number of words in each document
- w: A word in a document. This is represented as a one hot encoded vector of size V (i.e. V: vocabulary size)
- **w** (bold w): represents a document (i.e. vector of “w”s) of N words
- D: Corpus, a collection of M documents

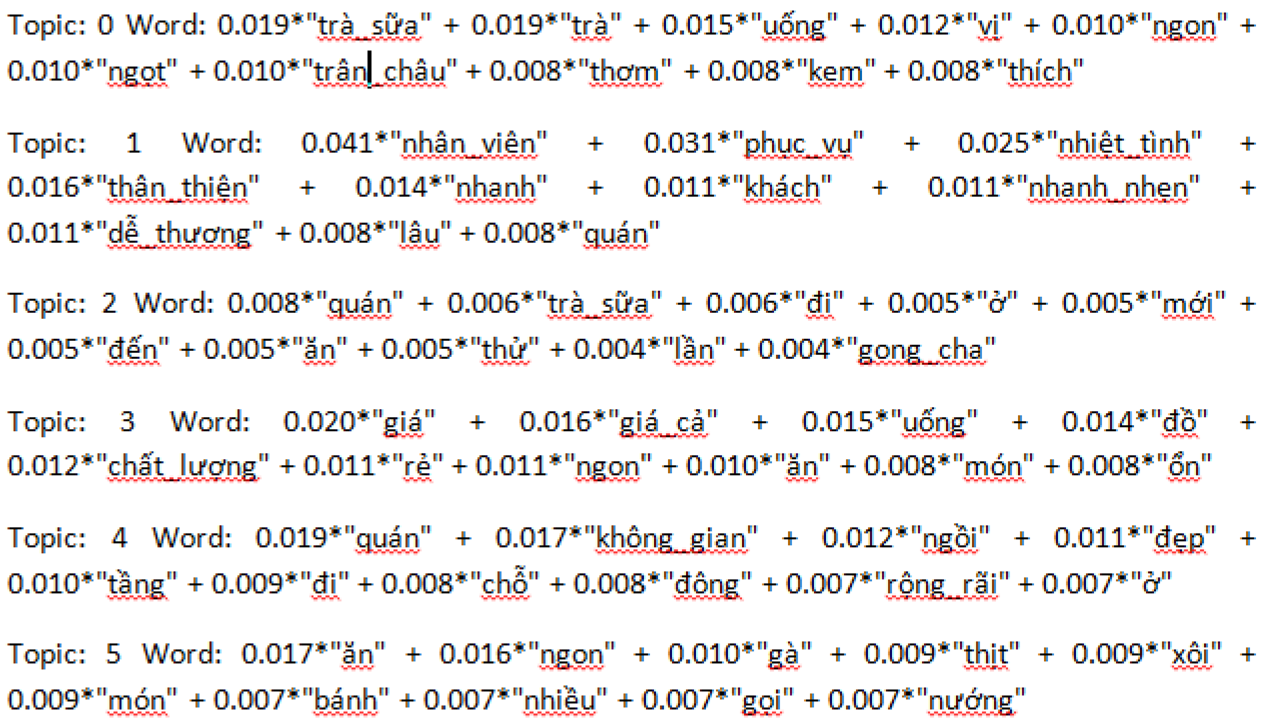
- z: A topic from a set of k topics. A topic is a distribution words. For example it might be

To apply the LDA method to our data set, we assume that each sentence in the review represents one topic only, although sometimes this is not true. The steps are as follows:

1. Split cleaned reviews into separated sentences based on punctuation
2. Tokenize each review into separated words and remove stop words (words that have high frequency but do not really contribute to the sentiment of each sentence)
3. Build dictionary form the tokenized words in each review
4. Create Term Frequency-Inverse Document Frequency(TF-IDF) matrix from the corpus. The TF-IDF is also a bag-of-words model but unlike the regular corpus, TF-IDF down weights tokens (words) that appears frequently across documents.
5. Feed the dictionary and TF-IDF matrix into the LDA model and try with different number of topics
6. Choose the best number of topics and apply the result to each sentence to get back the topic with the highest probability.

We use Python package **gensim** - a package for processing texts, working with word vector models (such as Word2Vec, FastText etc) and for building topic models. After trying with a different number of topics, we found that basically the reviews can be divided into 6 topics. Base on this result, we manually set the topic for each combination of words as follow:
- Topic 0: Drink
- Topic 1: Staff
- Topic 2: Experience
- Topic 3: Price
- Topic 4: Ambiance
- Topic 5: Food

We then apply the result above for each splitted sentence in the corpus and get back to topic with highest probability. Some example below shows that the model give somewhat reasonable outcome for simple sentence, mentioning only one topic.
After getting topic for each sentence, we summarize them number on each topic and plot the result.
From the plot, we can see that **food** aspect get the most number of reviews on, next are ambiance and price.



Topic: 0 Word: 0.019*"trà_sữa" + 0.019*"trà" + 0.015*"uống" + 0.012*"vị" + 0.010*"ngon" + 0.010*"ngọt" + 0.010*"trân_châu" + 0.008*"thơm" + 0.008*"kem" + 0.008*"thích"

Topic: 1 Word: 0.041*"nhân_viên" + 0.031*"phục_vụ" + 0.025*"nhiệt_tình" + 0.016*"thân_thiện" + 0.014*"nhanh" + 0.011*"khách" + 0.011*"nhanh_nhẹn" + 0.011*"dễ_thương" + 0.008*"lâu" + 0.008*"quán"

Topic: 2 Word: 0.008*"quán" + 0.006*"trà_sữa" + 0.006*"đi" + 0.005*"ở" + 0.005*"mới" + 0.005*"đến" + 0.005*"ăn" + 0.005*"thử" + 0.004*"lần" + 0.004*"gong_cha"

Topic: 3 Word: 0.020*"giá" + 0.016*"giá_cả" + 0.015*"uống" + 0.014*"đồ" + 0.012*"chất_lượng" + 0.011*"rẻ" + 0.011*"ngon" + 0.010*"ăn" + 0.008*"món" + 0.008*"ổn"

Topic: 4 Word: 0.019*"quán" + 0.017*"không_gian" + 0.012*"ngồi" + 0.011*"đẹp" + 0.010*"tầng" + 0.009*"đi" + 0.008*"chỗ" + 0.008*"đông" + 0.007*"rộng_rãi" + 0.007*"ở"

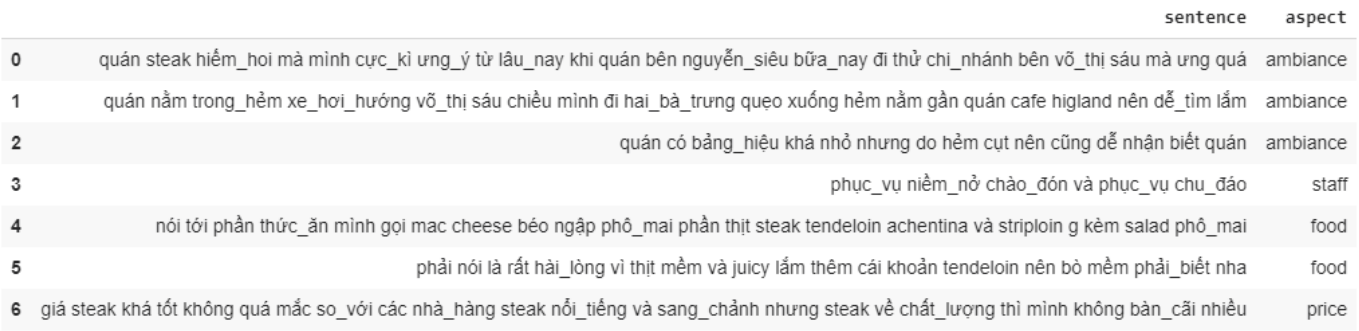
Topic: 5 Word: 0.017*"ăn" + 0.016*"ngon" + 0.010*"gà" + 0.009*"thịt" + 0.009*"xôi" + 0.009*"món" + 0.007*"bánh" + 0.007*"nhiều" + 0.007*"gọi" + 0.007*"nướng"

Figure 3.5: Topics by LDA models

3.2.2 Word2Vec

Word2Vec is a Word Embedding method, which is present each word in corpus into vector space. The Word2Vec could capture the similarity and differences of words in a corpus. The Word2Vec have the structure as follow:

- **Input Layer**: the one-hot encoder vector presentation of the corpus.
- **Hidden Layer**: the vector presentations of the words.
- **Output Layer**: the probability of the word nearby is chosen. This layer has the same node as the input layer, which is corresponding with each word on the corpus.



	sentence	aspect
0	quán steak hiếm_hoi mà mình cực_kì ưng_ý từ lâu_nay khi quán bên nguyễn_siêu bữa_nay đi thử chi_nhánh bên võ_thị sáu mà ưng quá	ambiance
1	quán nằm trong_hẻm xe_hơi_hướng võ_thị sáu chiều mình đi hai_bà_trưng quẹo xuống hẻm nằm gần quán cafe higland nên dễ_tìm lắm	ambiance
2	quán có bảng_hiệu khá nhỏ nhưng do hẻm cụt nên cũng dễ nhận biết quán	ambiance
3	phục_vụ niềm_nở chào_đón và phục_vụ chu_đáo	staff
4	nói tới phần thức_ăn mình gọi mac cheese béo ngập phô_mai phần thịt steak tendeloin achentina và striploin g kèm salad phô_mai	food
5	phải nói là rất hài_lòng vì thịt mềm và juicy lắm thêm cái khoản tendeloin nên bò mềm phải_biết nha	food
6	giá steak khá tốt không quá mắc so_với các nhà_hàng steak nổi_tiếng và sang_chảnh nhưng steak về chất_lượng thì mình không bàn_cãi nhiều	price

Figure 3.6: Apply result of LDA topic on each sentence

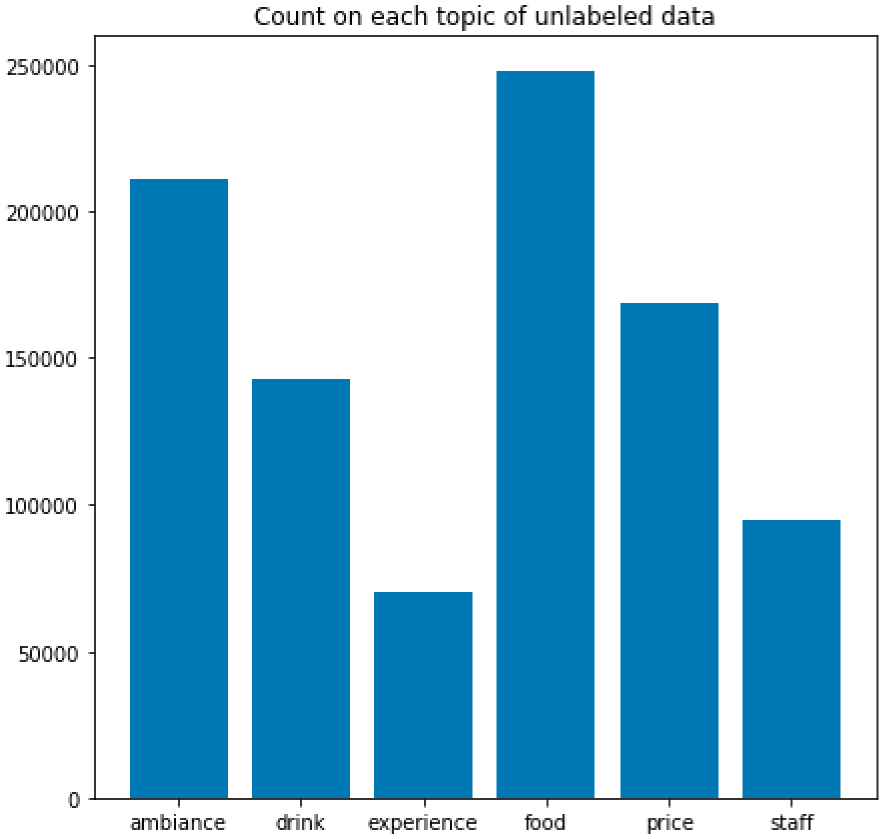


Figure 3.7: Count on each topic of unlabeled data

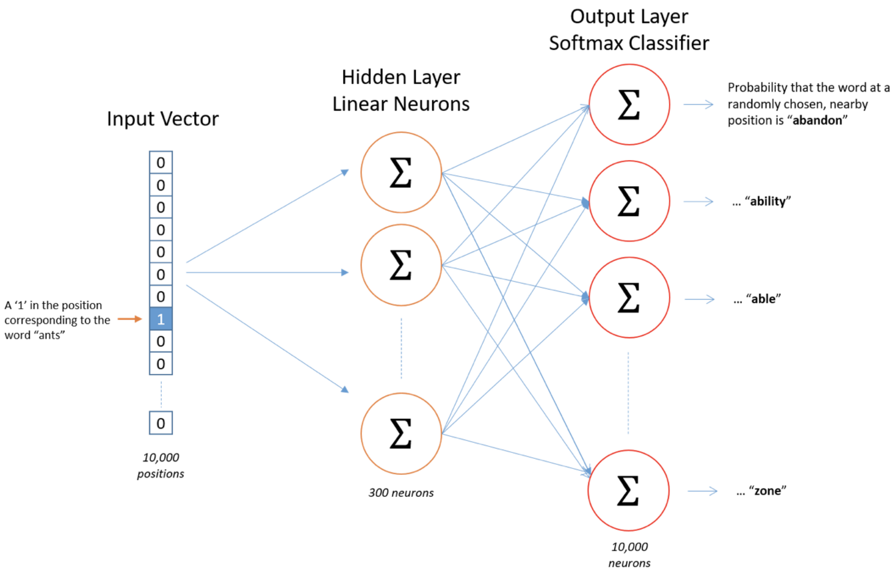


Figure 3.8: Word2Vec

The hyper-parameter we need to define for our network is:

- **Hidden Layer Nodes**: Number of dimensions that our vector will present the word. In practical, we finding that the neuron works best in 300 hidden nodes.
- **Min-Count**: The threshold of word's occurrences. We want to eliminate the unimportant word, which could lead us into false clustering topics. We set it to 500
- **Windows**: The distances around words to get the cor-occurrences probability. We set it to 2.

The final result is visualized (using the T-SNE method to reducing the dimension to 2) as following images.

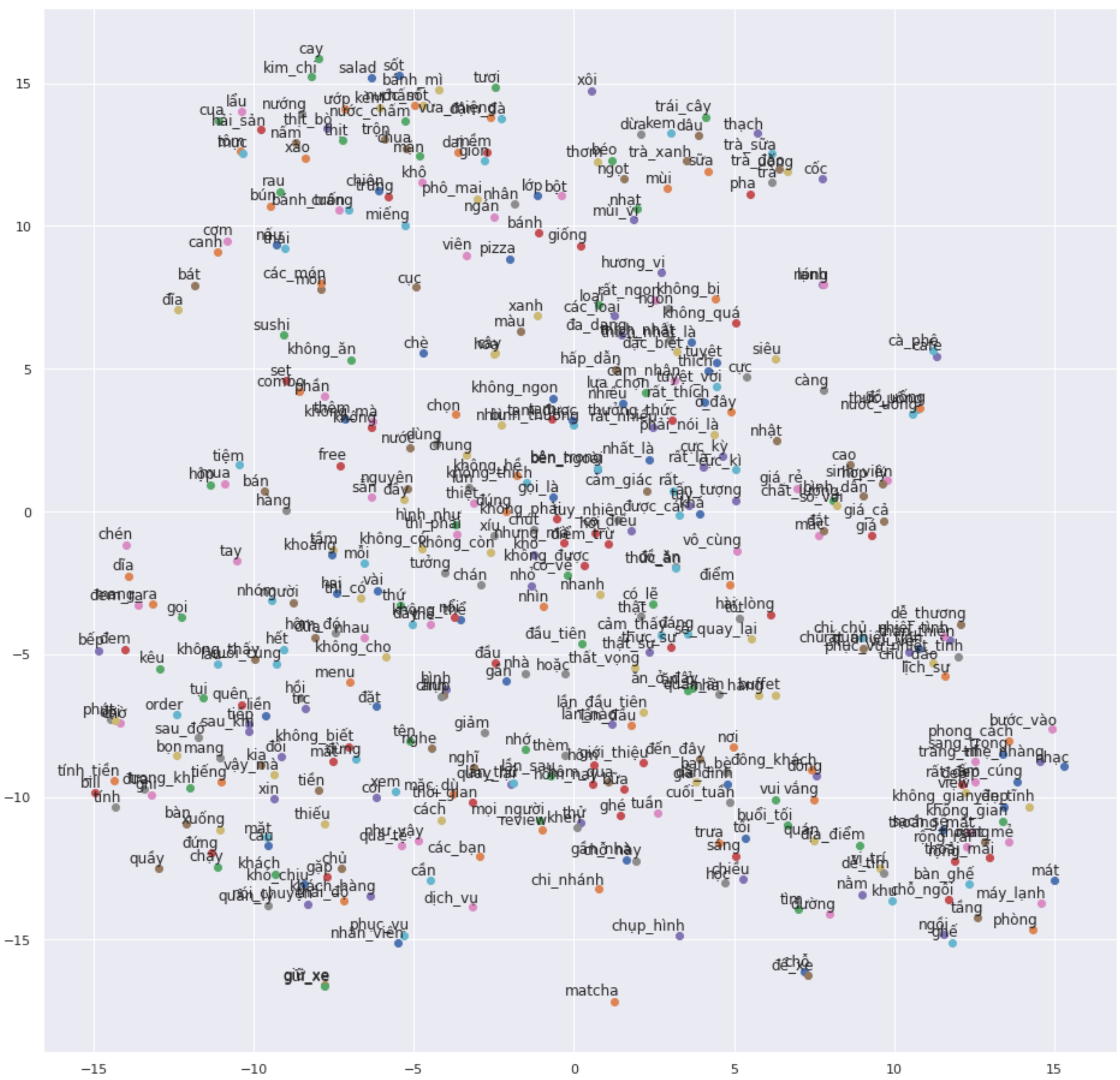
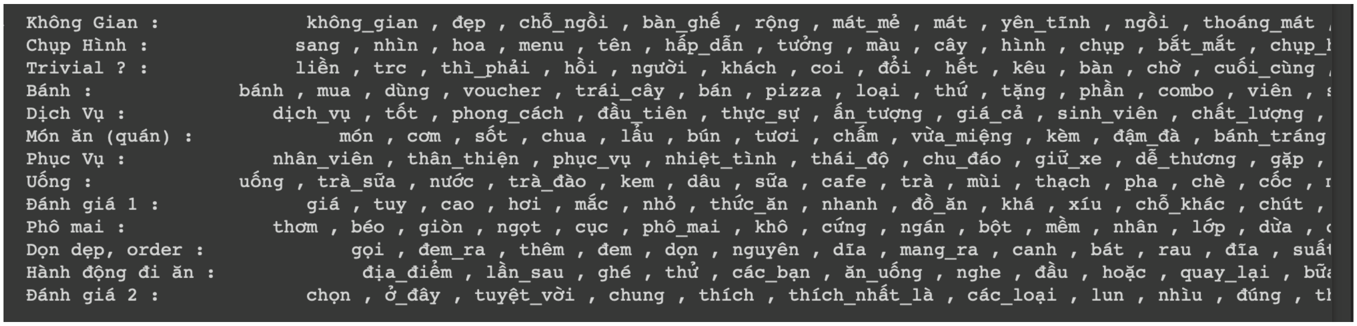


Figure 3.9: Visualize Word2Vec

Then we using the clustering method for selecting topics. There are two methods concerned: KMean and Gaussian Mixture Models. The KMean using the cosine distances to clustering. However we finding that it works not well as Gaussian Mixture Models. The optimal cluster is 12. The final result is shown below.



```
Không Gian :            không_gian , đẹp , chỗ_ngồi , bàn_ghế , rộng , mát_mẻ , mát , yên_tĩnh , ngồi , thoáng_mát ,
Chụp Hình :            sang , nhìn , hoa , menu , tên , hấp_dẫn , tưởng , màu , cây , hình , chụp , bắt_mắt , chụp_l
Trivial ? :            liền , trc , thì_phải , hồi , người , khách , coi , đổi , hết , kêu , bàn , chờ , cuối_cùng ,
Bánh :             bánh , mua , dùng , voucher , trái_cây , bán , pizza , loại , thứ , tặng , phần , combo , viên , s
Dịch Vụ :             dịch_vụ , tốt , phong_cách , đầu_tiên , thực_sự , ấn_tượng , giá_cả , sinh_viên , chất_lượng ,
Món ăn (quán) :            món , cơm , sốt , chua , lẩu , bún , tươi , chấm , vừa_miệng , kèm , đậm_đà , bánh_tráng
Phục Vụ :             nhân_viên , thân_thiện , phục_vụ , nhiệt_tình , thái_độ , chu_đáo , giữ_xe , dễ_thương , gặp ,
Uống :             uống , trà_sữa , nước , trà_đào , kem , dâu , sữa , cafe , trà , mùi , thạch , pha , chè , cốc , r
Đánh giá 1 :            giá , tuy , cao , hơi , mắc , nhỏ , thức_ăn , nhanh , đồ_ăn , khá , xíu , chỗ_khác , chút ,
Phô mai :             thơm , béo , giòn , ngọt , cục , phô_mai , khô , cứng , ngán , bột , mềm , nhân , lớp , dừa , c
Dọn dẹp, order :             gọi , đem_ra , thêm , đem , dọn , nguyên , dĩa , mang_ra , canh , bát , rau , đĩa , suất
Hành động đi ăn :             địa_điểm , lần_sau , ghé , thử , các_bạn , ăn_uống , nghe , đầu , hoặc , quay_lại , bữa
Đánh giá 2 :            chọn , ở_đây , tuyệt_vời , chung , thích , thích_nhất_là , các_loại , lun , nhìu , đúng , th
```

Figure 3.10: Topics Modeling by GMM and Word2Vec

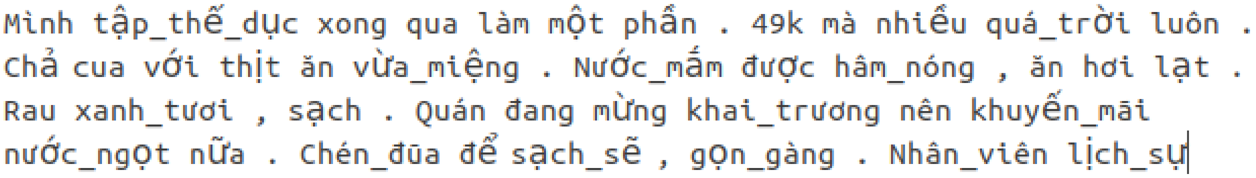
Comparing with LDA, the Word2Vec method is more stable and accurate with small corpus,however, LDA out perform Word2Vec on large scale. Moreover it fixes each topic with the specific words and the bias is more than LDA models. Therefore we using LDA models to select the aspect.

3.3 Sentiment Analysis

The labeled data set which contains 50,000 reviews was splitted into 3 subsets:

- 30,000 reviews for training.
- 10,000 reviews for validation.
- 10,000 reviews for testing.

Figure 3.11 shows a sample review. We use 3 models: Logistic Regression (baseline model), Neural Network, Bidirectional LSTM.



Mình tập_thể_dục xong qua làm một phần . 49k mà nhiều quá_trời luôn . Chả cua với thịt ăn vừa_miệng . Nước_mắm được hâm_nóng , ăn hơi lạt . Rau xanh_tươi , sạch . Quán đang mừng khai_trương nên khuyến_mãi nước_ngọt nữa . Chén_đũa để sạch_sẽ , gọn_gàng . Nhân_viên lịch_sự

Figure 3.11: A sample review

Logistic regression[1] is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled ”0” and ”1”.

[1] https://en.wikipedia.org/wiki/Logistic$_{r}egression$

Neural networks[2] or Artificial neural networks (ANN) are computing systems that are inspired by, but not necessarily identical to, the biological neural networks that constitute animal brains. Such systems ”learn” to perform tasks by considering examples, generally without being programmed with any task-specific rules.

Long short-term memory[3] (LSTM) is an artificial recurrent neural network (RNN) architecture which avoids the vanishing gradient problem. LSTM is normally augmented by recurrent gates called ”forget” gates. LSTM prevents backpropagated errors from vanishing or exploding.

Bidirectional LSTM[4] use a finite sequence to predict or label each element of the sequence based on the element’s past and future contexts. This is done by concatenating the outputs of two LSTM, one processing the sequence from left to right, the other one from right to left. We use the Bidirectional LSTM model since it is naturally think that the meaning of a word in the review can refer to it’s preceding or succeeding words.

We train 7 binary classifiers to compare the performance and select the most appropriate classifier. We then use the selected classifier to predict the sentences retrieved from Aspect Analysis step. Following are our classifiers:

- C1: Logistic Regression using bag-of-word.
- C2: Neural Network using bag-of-word.
- C3: Neural Network with embedding layer.
- C4: Neural Network with pre-trained word embedding in labeled dataset.
- C5: Neural Network with pre-trained word embedding in unlabeled dataset.
- C6: Bidirectional LSTM with pre-trained word embedding in labeled dataset.
- C7: Bidirectional LSTM with pre-trained word embedding in unlabeled dataset.

Table 3.1 shows the performance of each classifier.

The classifier C6 and C7 have the best performance over 7 classifiers based on accuracy. Related work achieved the accuracy of 87.64% with LSTM model [5].

Since the data we want to predict is in the unlabeled dataset, we choose the classifier C7 to utilize the pre-trained word embedding in the unlabeled dataset.

The Figure 3.12 shows the predicted results in 10 random sentences retrieved from Aspect Analysis step.

[2] https://en.wikipedia.org/wiki/Artificial$_{n}eural_{n}etwork$

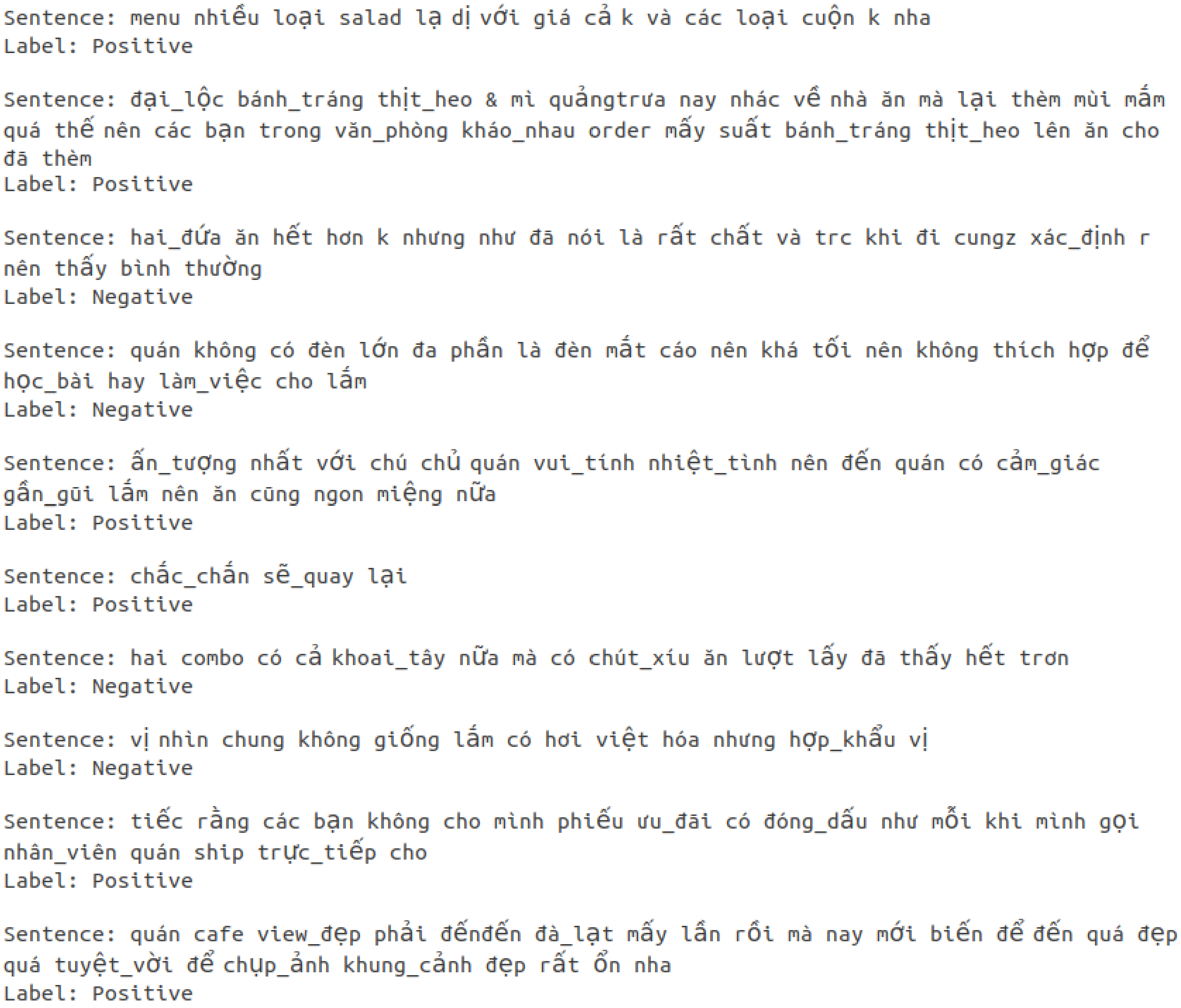
[3] https://en.wikipedia.org/wiki/Recurrent$_{n}eural_{n}etwork$

[4] https://en.wikipedia.org/wiki/Recurrent$_{n}eural_{n}etworkBi - directional$

[5] https://streetcodevn.com/blog/sav

Model	Training accuracy	Validation accuracy	Testing accuracy	Epochs
C1	0.909	0.875	0.879	
C2	0.972	0.853	0.853	10
C3	0.999	0.864	0.867	10
C4	0.800	0.787	0.791	30
C5	0.956	0.843	0.844	50
C6	**0.930**	**0.879**	**0.886**	10
C7	**0.929**	**0.873**	**0.885**	10

Table 3.1: Performance of 7 classifiers in labeled dataset



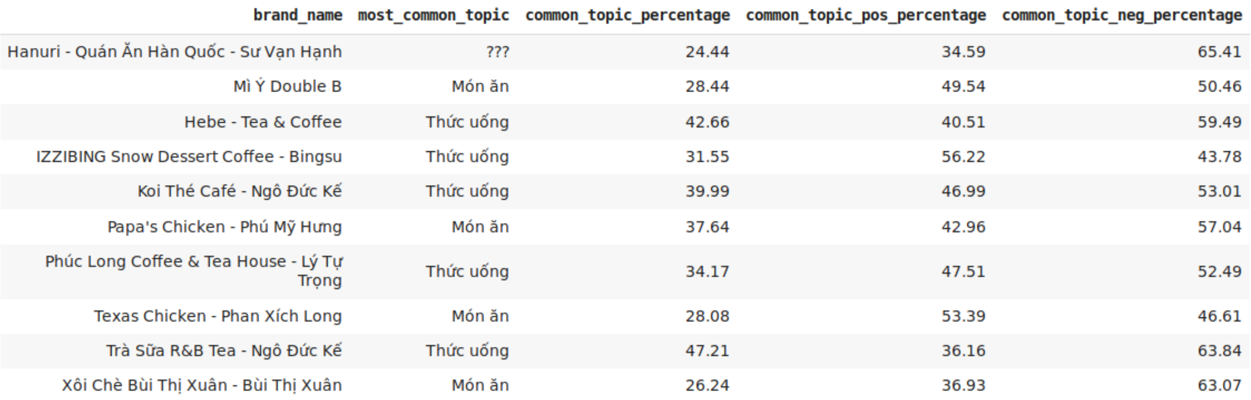
```
Sentence: menu nhiều loại salad lạ dị với giá cả k và các loại cuộn k nha
Label: Positive

Sentence: đại_lộc bánh_tráng thịt_heo & mì quảngtrưa nay nhác về nhà ăn mà lại thèm mùi mắm
quá thế nên các bạn trong văn_phòng kháo_nhau order mấy suất bánh_tráng thịt_heo lên ăn cho
đã thèm
Label: Positive

Sentence: hai_đứa ăn hết hơn k nhưng như đã nói là rất chất và trc khi đi cungz xác_định r
nên thấy bình thường
Label: Negative

Sentence: quán không có đèn lớn đa phần là đèn mắt cáo nên khá tối nên không thích hợp để
học_bài hay làm_việc cho lắm
Label: Negative

Sentence: ấn_tượng nhất với chú chủ quán vui_tính nhiệt_tình nên đến quán có cảm_giác
gần_gũi lắm nên ăn cũng ngon miệng nữa
Label: Positive

Sentence: chắc_chắn sẽ_quay lại
Label: Positive

Sentence: hai combo có cả khoai_tây nữa mà có chút_xíu ăn lượt lấy đã thấy hết trơn
Label: Negative

Sentence: vị nhìn chung không giống lắm có hơi việt hóa nhưng hợp_khẩu vị
Label: Negative

Sentence: tiếc rằng các bạn không cho mình phiếu ưu_đãi có đóng_dấu như mỗi khi mình gọi
nhân_viên quán ship trực_tiếp cho
Label: Positive

Sentence: quán cafe view_đẹp phải đếnđến đà_lạt mấy lần rồi mà nay mới biến để đến quá đẹp
quá tuyệt_vời để chụp_ảnh khung_cảnh đẹp rất ổn nha
Label: Positive
```

Figure 3.12: Predicted results in 10 random sentences retrieved from Aspect Analysis step

Chapter 4

Experimental Result

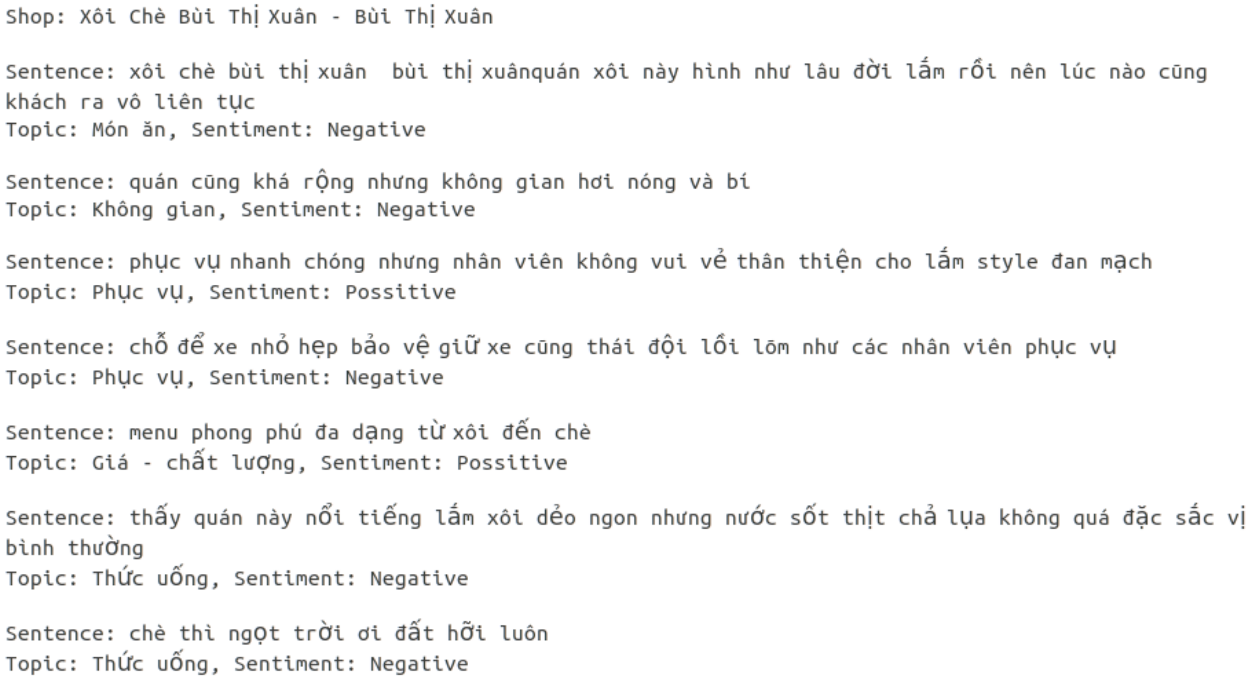
For each shop, we observe all reviews and analyze the reviews to understand what customers mostly talk about. Figure 4.1 shows the result in top 10 most reviewed shops.



brand_name	most_common_topic	common_topic_percentage	common_topic_pos_percentage	common_topic_neg_percentage
Hanuri - Quán Ăn Hàn Quốc - Sư Vạn Hạnh	???	24.44	34.59	65.41
Mì Ý Double B	Món ăn	28.44	49.54	50.46
Hebe - Tea & Coffee	Thức uống	42.66	40.51	59.49
IZZIBING Snow Dessert Coffee - Bingsu	Thức uống	31.55	56.22	43.78
Koi Thé Café - Ngô Đức Kế	Thức uống	39.99	46.99	53.01
Papa's Chicken - Phú Mỹ Hưng	Món ăn	37.64	42.96	57.04
Phúc Long Coffee & Tea House - Lý Tự Trọng	Thức uống	34.17	47.51	52.49
Texas Chicken - Phan Xích Long	Món ăn	28.08	53.39	46.61
Trà Sữa R&B Tea - Ngô Đức Kế	Thức uống	47.21	36.16	63.84
Xôi Chè Bùi Thị Xuân - Bùi Thị Xuân	Món ăn	26.24	36.93	63.07

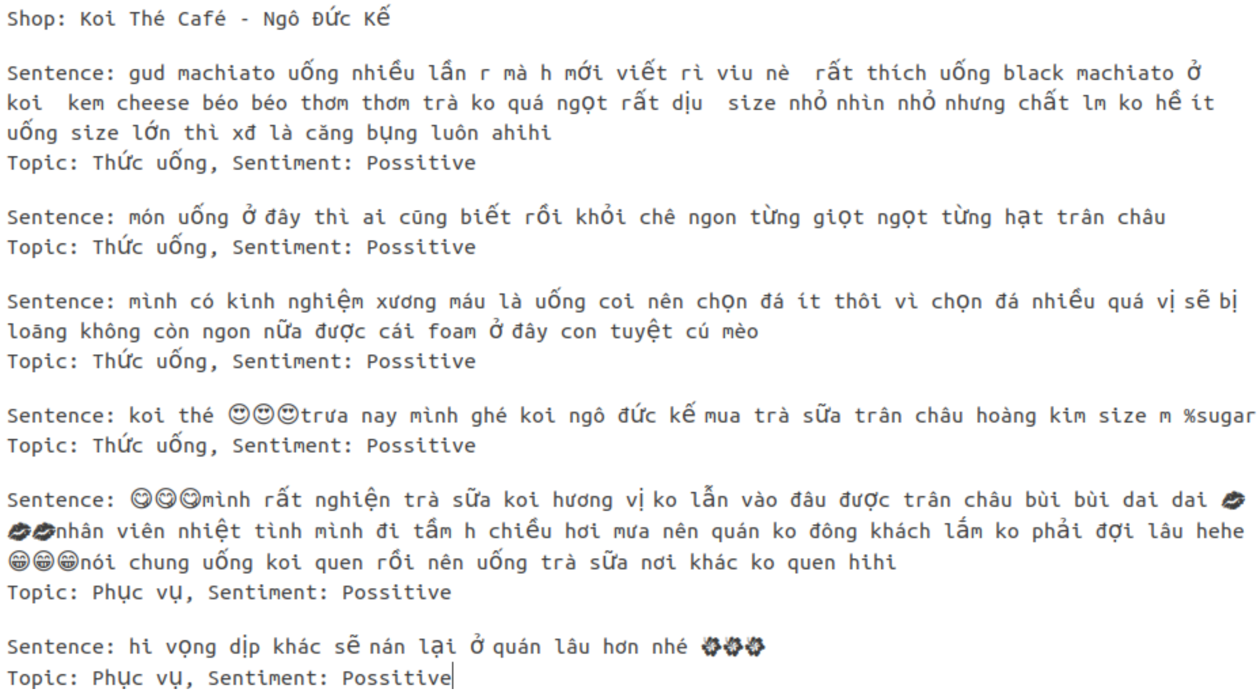
Figure 4.1: An analysis in top 10 most reviewed shops

We also analyze in detail each customer review in each shop. Following are the results in 3 selected shops:



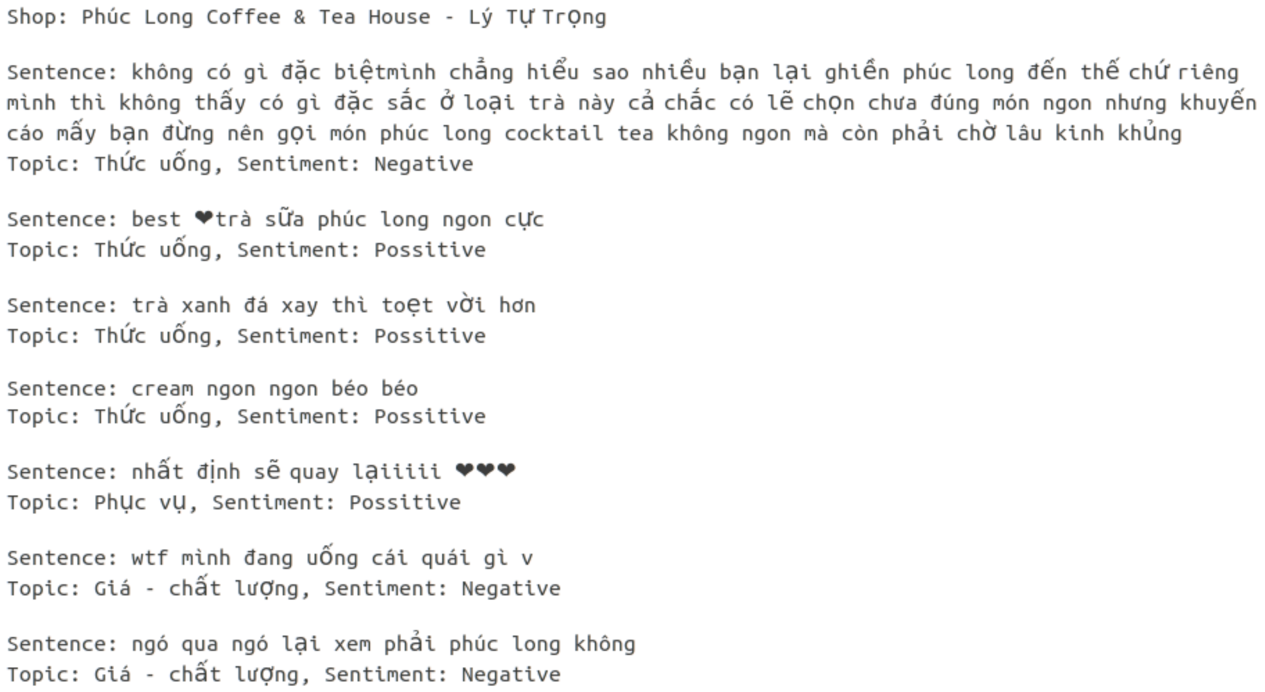
```
Shop: Xôi Chè Bùi Thị Xuân - Bùi Thị Xuân

Sentence: xôi chè bùi thị xuân  bùi thị xuânquán xôi này hình như lâu đời lắm rồi nên lúc nào cũng
khách ra vô liên tục
Topic: Món ăn, Sentiment: Negative

Sentence: quán cũng khá rộng nhưng không gian hơi nóng và bí
Topic: Không gian, Sentiment: Negative

Sentence: phục vụ nhanh chóng nhưng nhân viên không vui vẻ thân thiện cho lắm style đan mạch
Topic: Phục vụ, Sentiment: Possitive

Sentence: chỗ để xe nhỏ hẹp bảo vệ giữ xe cũng thái đội lồi lõm như các nhân viên phục vụ
Topic: Phục vụ, Sentiment: Negative

Sentence: menu phong phú đa dạng từ xôi đến chè
Topic: Giá - chất lượng, Sentiment: Possitive

Sentence: thấy quán này nổi tiếng lắm xôi dẻo ngon nhưng nước sốt thịt chả lụa không quá đặc sắc vị
bình thường
Topic: Thức uống, Sentiment: Negative

Sentence: chè thì ngọt trời ơi đất hỡi luôn
Topic: Thức uống, Sentiment: Negative
```

Figure 4.2: An detail analysis in Bui Thi Xuan restaurant



```
Shop: Koi Thé Café - Ngô Đức Kế

Sentence: gud machiato uống nhiều lần r mà h mới viết ri viu nè  rất thích uống black machiato ở
koi  kem cheese béo béo thơm thơm trà ko quá ngọt rất dịu  size nhỏ nhìn nhỏ nhưng chất lm ko hề ít
uống size lớn thì xđ là căng bụng luôn ahihi
Topic: Thức uống, Sentiment: Possitive

Sentence: món uống ở đây thì ai cũng biết rồi khỏi chê ngon từng giọt ngọt từng hạt trân châu
Topic: Thức uống, Sentiment: Possitive

Sentence: mình có kinh nghiệm xương máu là uống coi nên chọn đá ít thôi vì chọn đá nhiều quá vị sẽ bị
loãng không còn ngon nữa được cái foam ở đây con tuyệt cú mèo
Topic: Thức uống, Sentiment: Possitive

Sentence: koi thé 😍😍😍trưa nay mình ghé koi ngô đức kế mua trà sữa trân châu hoàng kim size m %sugar
Topic: Thức uống, Sentiment: Possitive

Sentence: 😋😋😋mình rất nghiện trà sữa koi hương vị ko lẫn vào đâu được trân châu bùi bùi dai dai 💋
💋💋nhân viên nhiệt tình mình đi tầm h chiều hơi mưa nên quán ko đông khách lắm ko phải đợi lâu hehe
😁😁😁nói chung uống koi quen rồi nên uống trà sữa nơi khác ko quen hihi
Topic: Phục vụ, Sentiment: Possitive

Sentence: hi vọng dịp khác sẽ nán lại ở quán lâu hơn nhé 🌺🌺🌺
Topic: Phục vụ, Sentiment: Possitive
```

Figure 4.3: An detail analysis in KOI Cafe



```
Shop: Phúc Long Coffee & Tea House - Lý TỰ TrỌng

Sentence: không có gì đặc biệtmình chẳng hiểu sao nhiều bạn lại ghiền phúc long đến thế chứ riêng
mình thì không thấy có gì đặc sắc ở loại trà này cả chắc có lẽ chọn chưa đúng món ngon nhưng khuyến
cáo mấy bạn đừng nên gọi món phúc long cocktail tea không ngon mà còn phải chờ lâu kinh khủng
Topic: Thức uống, Sentiment: Negative

Sentence: best ❤trà sữa phúc long ngon cực
Topic: Thức uống, Sentiment: Possitive

Sentence: trà xanh đá xay thì toẹt vời hơn
Topic: Thức uống, Sentiment: Possitive

Sentence: cream ngon ngon béo béo
Topic: Thức uống, Sentiment: Possitive

Sentence: nhất định sẽ quay lạiiiii ❤❤❤
Topic: Phục vụ, Sentiment: Possitive

Sentence: wtf mình đang uống cái quái gì v
Topic: Giá - chất lượng, Sentiment: Negative

Sentence: ngó qua ngó lại xem phải phúc long không
Topic: Giá - chất lượng, Sentiment: Negative
```

Figure 4.4: An detail analysis in Phuc Long Coffee and Tea House

Chapter 5

Conclusion

We achieve the objective of this project in terms of doing aspect-based sentiment analysis.
For aspect analysis, LDA generate more accurate cluster than Word2Vec with large number of corpus.
For sentiment analysis, we have not tuned hyper-parameters for the neural networks. We think model performance could be improved if we tune the hyper-parameters.
Customer reviews on e-commerce platforms have unusual characteristics: false marks, no marks, icon,... that need to be handled carefully.
Finally, in this project we assume that each sentence in one review represent an aspect. Sometimes this is not true since customer may mention two aspects within one sentence. For future work, it could be worthwhile trying new ways to separate the aspect more efficiently.