Z534: Final Project

Aishwarya Iyer
Maiyaporn Phanich
Rohit Zawar

+ Task 1: Categories Prediction

Predict restaurant's categories from review texts

Solution

- Topic Modelling (Latent Dirichlet Allocation)
  - Group businesses together by their category
  - Concatenate all reviews within the same group
  - Train LDA to find distributions over K topics for each category
- Predict categories by measuring topics similarity between review text and category documents
  - Cosine Similarity
  - Hellinger Distance
- Evaluate precision, recall, and F-measure

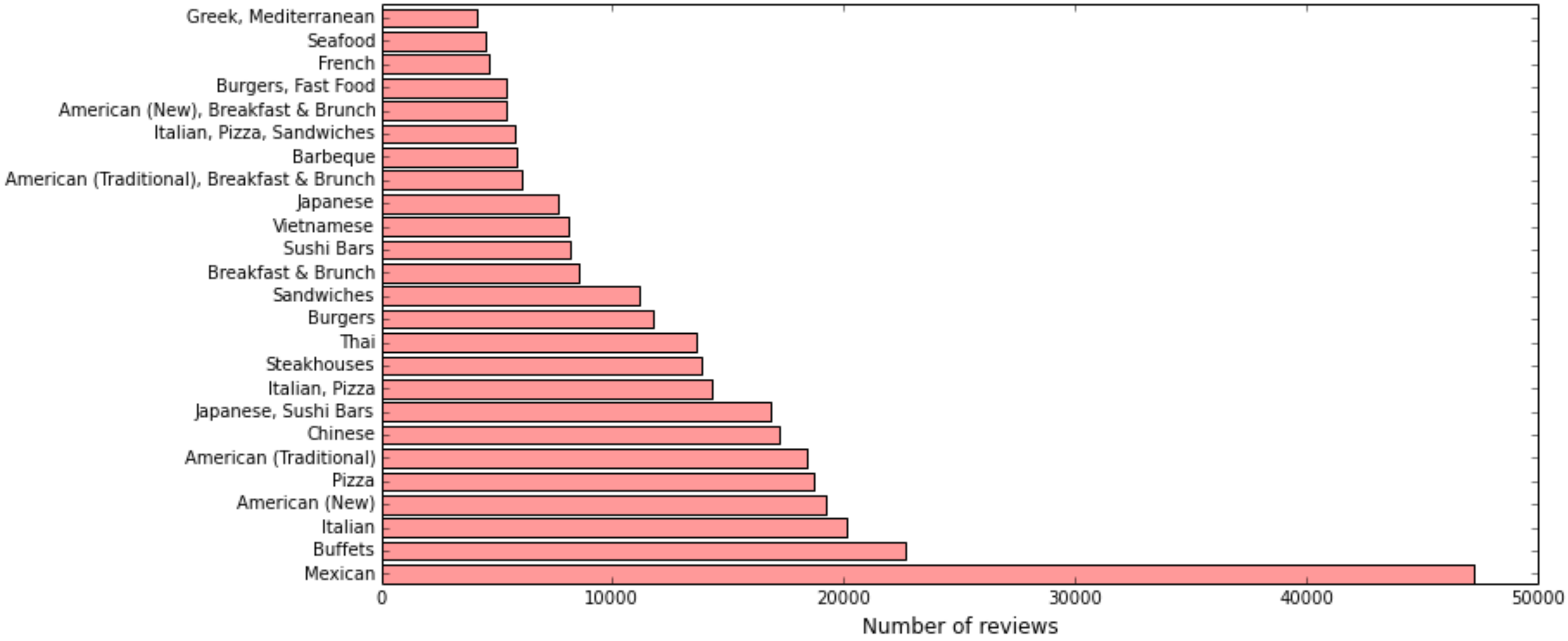
Tools and Library

- Pandas - Python Data Analysis Library
- NLTK - Natural Language Toolkit
- Gensim - Topic modelling library
- Numpy, Scipy - Scientific computing library
- Matplotlib - Plotting library
- Word cloud - Word cloud generator

Data and Pre-processing

- Select only restaurant businesses from the dataset
- Pick the top 25 populated categories by number of reviews
    - 6,620 restaurants
    - 319,431 reviews
    - 119,623 users
- Pre-process review texts
    - Remove stop words and punctuation
    - Remove word with less than 3 characters
    - Lemmatization
    - Remove extreme words (less than 20% and more than 70%)

+ Top 25 categories



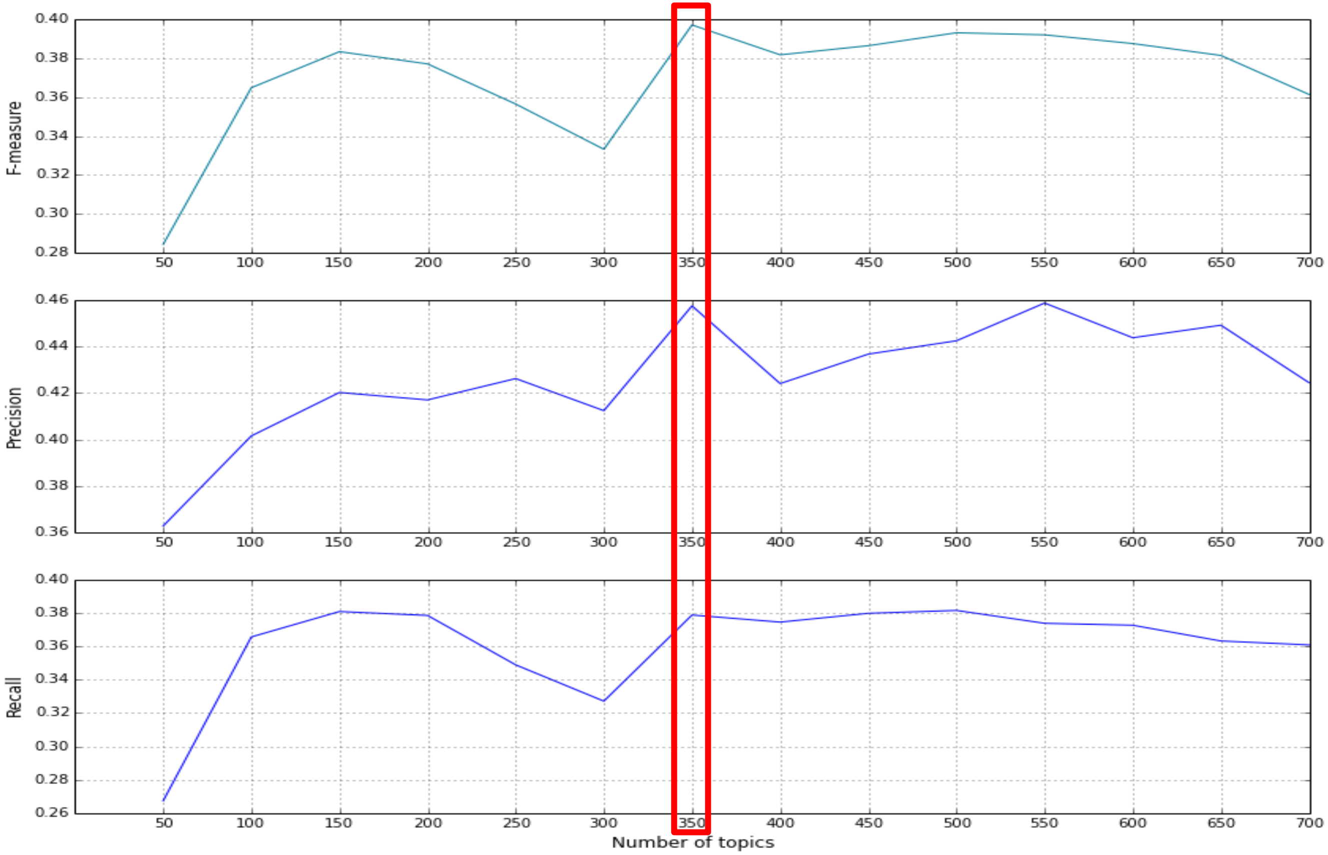
Experiment

- Split the data into 60:20:20
  - Training (191661 reviews)
  - Validation (63887 reviews)
  - Test (63884 reviews)
- Group training data by categories and combine texts within group
  - 25 categories/documents
  - 18498 unique tokens after pre-processing
- Train LDA model with training set
  - Batch training with 20 iterations
  - Different values of K from 50 to 700 (+50)

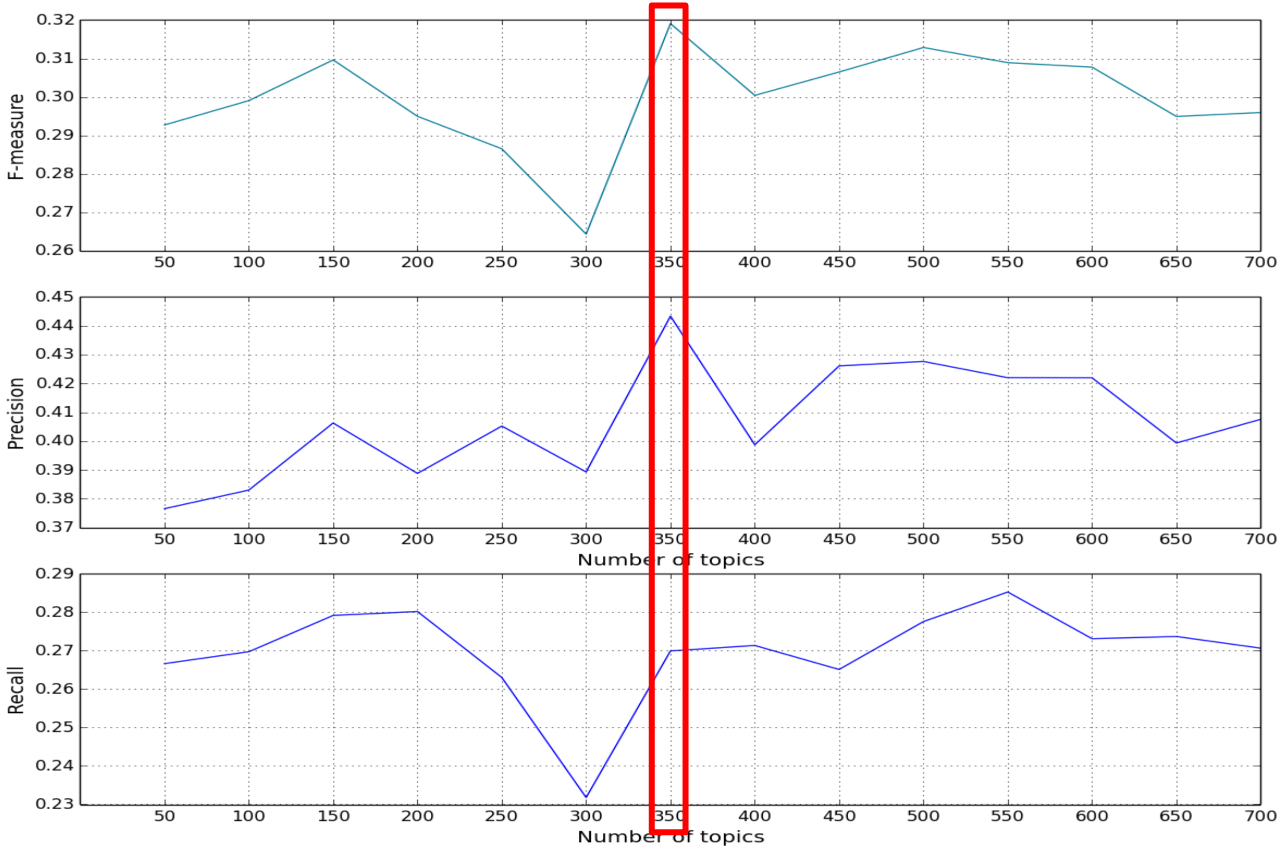
Experiment (Cont'd)

- Estimate the number of topics (K) from validation set
  - Calculate similarity score for each review and all training documents
  - Assign the category from the most similar document as a prediction
  - Calculate precision, recall, and F-measure of the predicted categories and the actual categories
  - Pick k that gives the best results
- Repeat the same process for the selected k on test set
- Compare the results with baseline system (TF-IDF)

+ Cosine Similarity



+ Hellinger Distance



+ Category - Mexican

[(249, 0.70121786618165927), (256, 0.085152231945685691), (158, 0.082377171771929214)]

+ Category - Japanese

[(199, 0.71582295325677625), (97, 0.092941510768914437), (256, 0.074904510006921415)]

+ Category - Barbeque

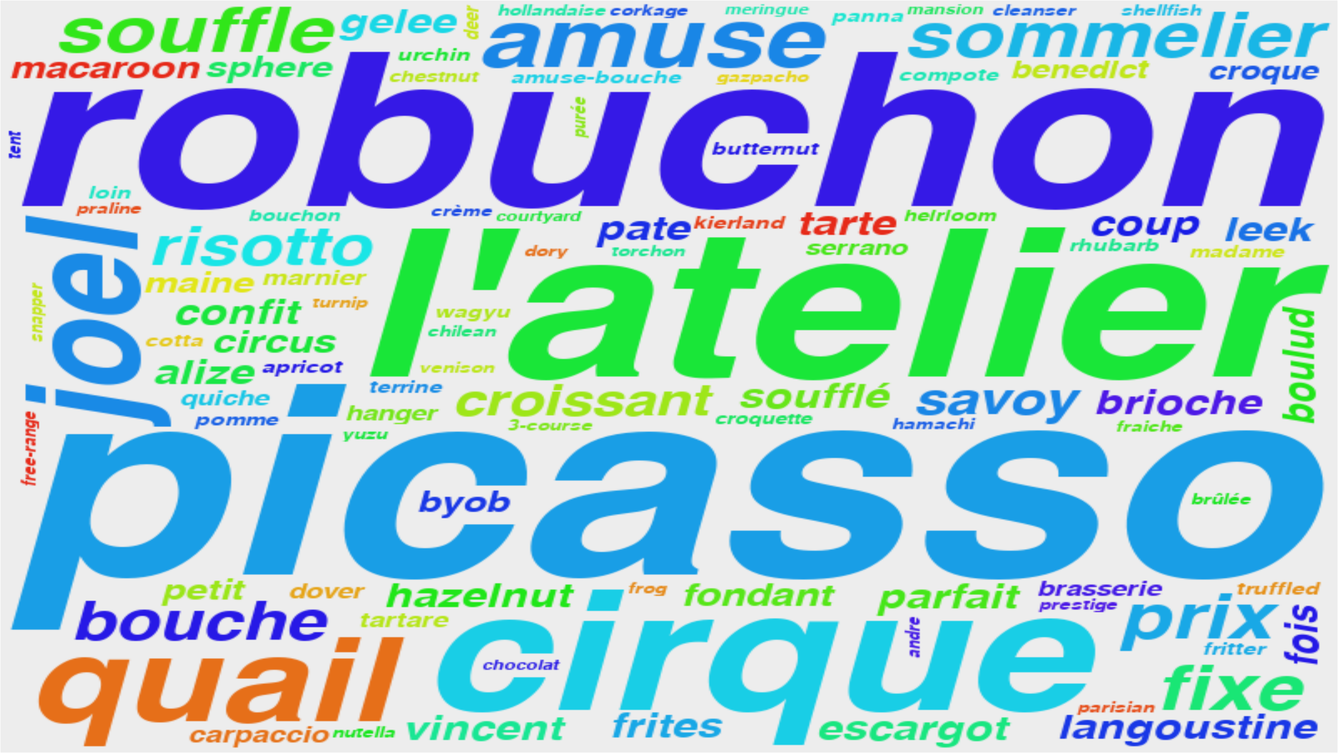
[(158, 0.99954876376874924)]

+ Category - Italian, Pizza, Sandwiches

[(17, 0.86010425652878764), (158, 0.068995143099881459), (256, 0.036552512147857014)]

+ Category - French

[(58, 0.67275509702210834), (256, 0.088202317335136893), (158, 0.060950497077218002)]



Results

Baseline: TF-IDF

	Precision	Recall	F-Measure
Cosine Similarity	0.4373097	0.379436	0.395036
Hellinger Distance	-	-	-

LDA with K=350

	Precision	Recall	F-Measure
Cosine Similarity	0.421972	0.356173	0.372198
Hellinger Distance	0.412693	0.257169	0.301927

Summary

- Stopwords list is very important!!
- LDA is a time consuming model to train
- Hard to determine training parameters
  - Number of topics (k)
  - Number of iterations (i)
- High number of k gives repetitive words in many topics

+ Task 2: Rating Prediction

Predict review rating from review texts

Solution

- Multi-class classification/regression problem
- Classes = [1, 2, 3, 4, 5] Stars
- Preprocess review data to club all reviews of same user
- Process review text and perform Sentiment Analysis to extract features
- Train 80% dataset and test 20% dataset for each user model
- Evaluation Metrics : Accuracy, RMSE, Precision, Recall, F-Measure

Data Preprocessing

- Total number of Reviews : 1, 125, 458
- Total number of Distinct Users : 252, 898
- Users with > 100 reviews : 392
- Clubbed reviews of 392 users together
- Tool : MongoDB

Data Preprocessing (Contd..)

- Processed Data JSON format :

{

“ user_id” : “….”,

“ reviews” : [

{ review_id : “…”, text : “……………………”, business_id : “…”,

stars: “…” } ,

{ review_id : “…”, text : “……………………”, business_id : “…”,

stars: “…” } ,

…….

…….

]

}

Sentiment Analysis

- For each user model,
  - extract sentiment of each review
  - Sentiment classes : 5
    - very negative, neg, neutral, positive, very positive
- Before sentiment analysis , process text :
  - tokenize
  - sentence split
  - parse
- Then perform sentiment analysis on each sentence of review text
- Tool : Stanford NLP parser

Machine Learning

- Tool : Weka
- Train model for every user

Features					Classes
Normalized Count of 'negative' sentences	Normalized Count of 'negative' sentences	Normalized Count of 'negative' sentences	Normalized Count of 'negative' sentences	Normalized Count of 'negative' sentences	No. of Stars (1,2,3,4,5)

- Split dataset : Training set - 80%, Testing set - 20%
- Algorithms : J48, Random Forest, SVM

Evaluation

1. As a Classification problem
2. As a Regression problem

Metrics :

- Accuracy
- Precision
- Recall
- F-Measure
- Root mean squared error

Evaluation - Classification problem

J48 Algorithm :

Accuracy	Precision	Recall	F-measure	RMSE
52.9239	0.233	0.482	0.314	0.3661

Summary

- Sentiment Analysis of Stanford NLP parser is very slow
- Results are good for straightforward sentences,
  but not very reliable in other cases
  Eg. "OMG, does any more need to be said about this place???"

- Could have considered more features for machine learning?
  - dependency between sentences of a text
  - n - grams etc

Things we learnt :

- Information Retrieval concepts and models
- NLP and Machine learning concepts and algorithms
- How IR, NLP and ML are inter-dependent
- Application of each and their pros and cons
- Various standard libraries available in each
- How to handle HUGE datasets!

THANK YOU :)