Generating Cafe Suggestions with Topic Extraction and Sentiment Analysis of Yelp Reviews

Anurag Prasad and Jarrod Lewis
CS591 Data Analytics – Spring 2017
Instructor: Babis Tsourakakis

Can we use Yelp reviews to generate tips for restaurants?

- Motivation
  - Yelp currently has “tips” from users to help other users
  - But reviews also hold valuable information for businesses!
- Project Goals
  - Look specifically at cafes: Yelp’s “Coffee & Tea” category
  - Discover topics relevant to a particular cafes based on its reviews
    - Analyze relationship between…
      - topics and star-ratings
      - topics and reviewer sentiment
    - Create a model to predict topics of unseen reviews
      - Generate feedback based on topic stars and sentiments

Related Work: tf-idf

- **Term frequency–inverse document frequency** [1]
  - Standard "bag of words" text information retrieval method
  - Measures the number of occurrences of word in entire text corpus
  - Benefits
    - Extracts most descriptive terms in dataset
    - Good for lexical features
  - Limitations
    - Reveals little about intra/inter-document structure
    - Not good for capturing semantics
      - Does not capture position in text,
Co-occurrences in different documents, etc.

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

tf: *"How often does term t occur in doc d?"*

idf: *"But how common is term t across all docs?"*

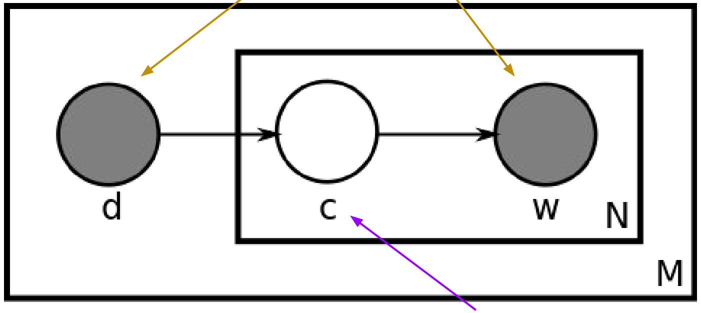
Related Work: pLSI

- **Probabilistic latent semantic indexing** [2]
  - Pr[word-doc co-occurrence] is mixture of conditionally independent multinomial distributions

$$p(d, w_n) = p(d) \sum_z p(w_n \mid z) p(z \mid d).$$

  - Limitations
    - Not flexible enough to handle unseen text (relies on training set)
    - No probabilistic model at document-level (ignores uncertainty)
      - # topics grows linearly with size of corpus → overfitting
    - “Bag of words” approach ignores word and document order

Doc *d*, word *w*:
observable variables



Topic *c*:
latent variable

Related Work: LDA

- **Latent Dirichlet Allocation** [3]
  - Two-level *generative probabilistic* model
    1. Choose θ ~ Dir(α)

       * θ : topic distribution for document *M*

       * Prior α : per-document topic dist.
    2. For each of *N* words w_n:

       choose a word w_n from $p(w_n \mid \theta, \beta)$

       * Prior β : per-topic word dist.

  - Improvements over standard pLSI
    - Modified to fit a Dirichlet distribution
    - Does not overfit on small datasets
  - Limitations
    - Topics can be hard to interpret ("supervised" part)

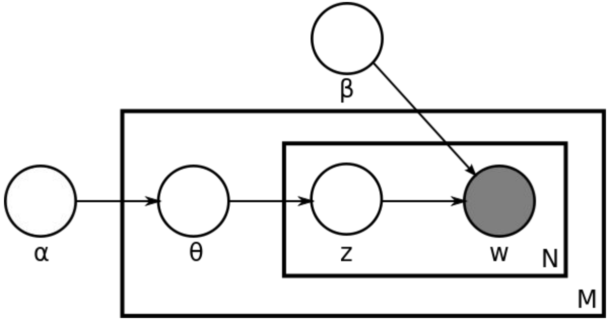
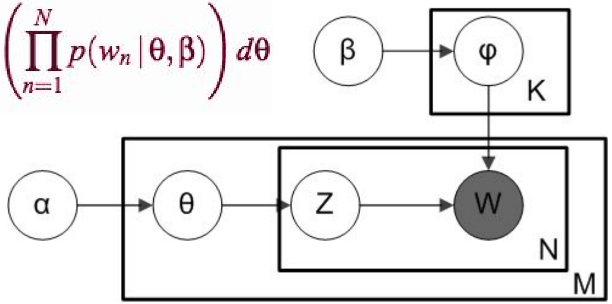


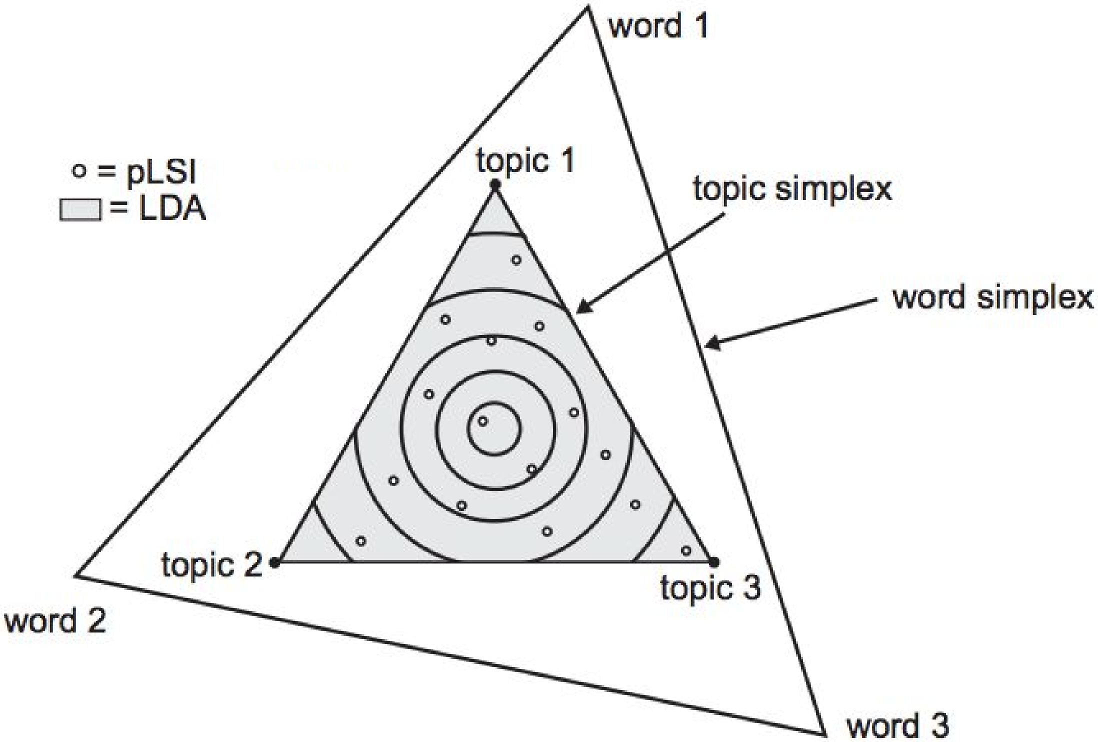
Plate notation for LDA model

$$p(\mathbf{w} \mid \alpha, \beta) = \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} p(w_n \mid \theta, \beta) \right) d\theta$$



LDA with Dirichlet-distributed topic-word distributions

Geometric Comparison: pLSI vs LDA



- pLSI
  - Multiple docs per topic, but…
  - Pr[documents] with pLSI are points
- LDA
  - Can place docs within *any point* within topic simplex

LDA Toy Example

1. Babis knows many things.
2. Ben would like to learn TensorFlow.
3. Babis knows TensorFlow.

Topic 1: ‘Babis’, ‘knows’, ‘things’

Topic 2: ‘Ben’, ‘learn’, ‘TensorFlow’

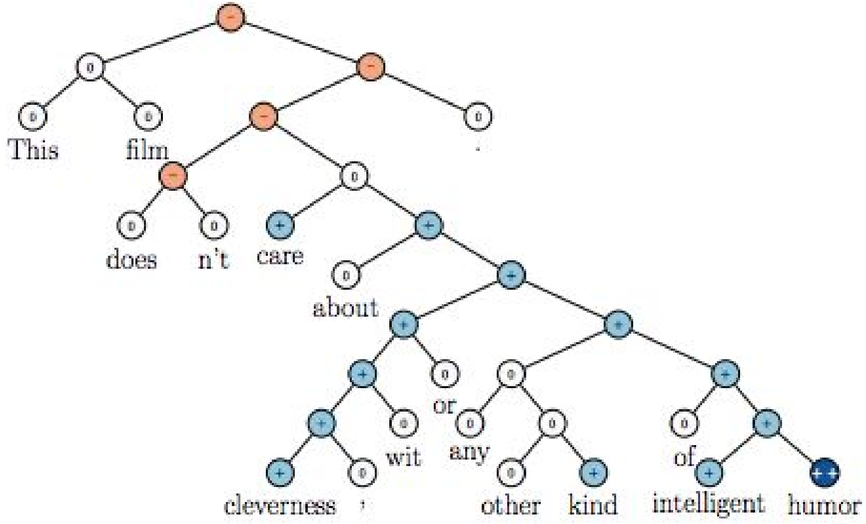
1. 100% Topic 1
2. 100% Topic 2
3. 67% Topic 1, 33% Topic 2

Related Work: Subtopic Extraction

- Subtopic Extraction from Reviews [4]
  - Used LDA model to extract subtopics from 158,000 reviews
  - Application focused on correlation between subtopics
  - Influenced our decisions for using LDA and Gensim library for topic modeling
- Gensim (topic modeling library)
  - It provides tools for transforming the text data into the relevant structure for the model
  - The model generated can be used to predict the topic distribution of an unseen document

Related Work: Sentiment Treebanks

- Sentiment Treebank [5]
  - Implementation of the Stanford Core NLP
  - Attempt to get more meaningful sentiment scores
    - The treebank looks at the overall sentence structure, not just a bag of words



Methodology

1. Process reviews in dataset to create usable corpus
2. Train LDA model on dataset
3. Extract topics using model
   a. per review
   b. per cafe
4. Get star ratings and sentiments of each topic
   a. Sentiment analysis with Sentiment Treebank
   b. Rank the topics based on star rating and sentiment score
5. Predict topics of unseen reviews
   a. Given a cafe Yelp url, return suggestions based on the cafe's reviews
   b. Produce suggestions for a cafe based on its reviews

Developing the LDA Model

- LDA model trained using the Yelp Academic Dataset
  - ‘Coffee & Tea’ category– 179,409 reviews on cafes
- Before the data could be fed into the LDA model it had to be cleaned
  - Stopwords filtered out of the data
  - Words stemmed so that their structure could be easily matched
  - Part-of-speech tagging used to filter for nouns, the best identifiers for subtopics
  - Words transformed into the gensim corpus format which maps words to values

Experiments: LDA Topic Extraction

- Topic Extraction
  - LDA model categorizes subtopics based on 10 key nouns
  - Based on experimentation we decided to use K=25 subtopic groups
    - K=50 topics produced poorly defined topics
    - Even at K=25 there was still some trouble with interpreting the groups

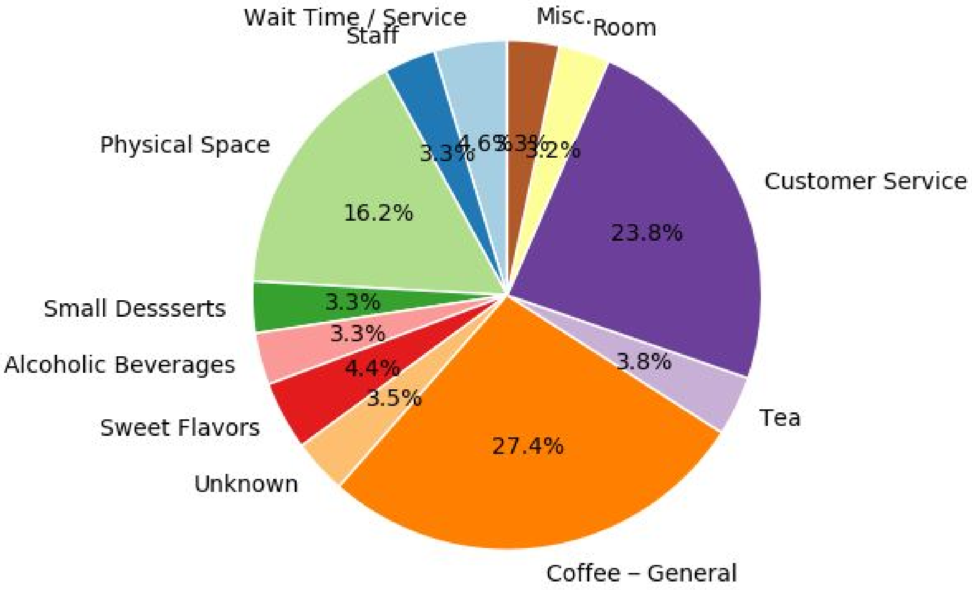
Coffee - General	Atmosphere	Wait Time/Service	Baked Bread Items
coffee (20.4%)	place (8.9%)	time (4.5%)	bagel (8.6%)
shop (3.9%)	staff (5.2%)	service (3.1%)	pastry (6.4%)
bean (1.3%)	music (2.1%)	order (3.4%)	bread (2.3%)
espresso (%)	fun (1.3%)	wait (1.3%)	baguette (1.6%)

Topic word distributions

Experiments: Sentiment Analysis Application

- Modified Python wrapper to call Stanford NLP Sentiment Treebank
- Reviews were broken up into their individual sentences and fed into the treebank
- { 0 : Very Negative, 1 : Negative, 2 : Neutral, 3 : Positive, 4 Very Positive }
- Combined sentiment scores used to assign aggregate sentiment score to topics
    - On review the use of the treebanks seems to be a significant improvement over traditional sentiment scoring

Experiments: Topic Breakdown for an Unseen Review

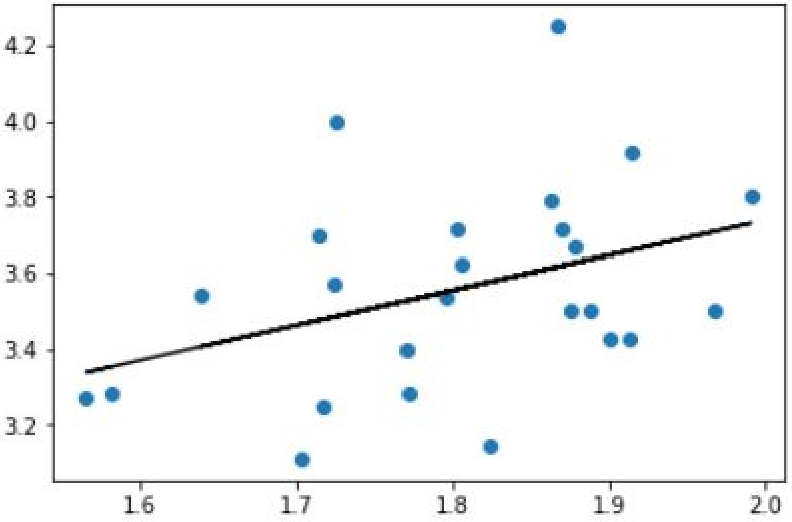


Experiments: Generating Recommendations for Cafes

- Example cafe recommendation: 3 Little Figs (https://www.yelp.com/biz/3-little-figs-somerville)
- Produced recommendation
    - Keep up the good work with…
        - Food & Meals (sentiment: slightly positive (2.44) / avg. stars: 4.86)
        - Physical Space (sentiment: slightly positive (2.35) / avg. stars: 4.67)
        - Atmosphere (sentiment: neutral (2.0) / avg. stars: 4.6)
    - May need to make improvements with...
        - Tea (sentiment: neutral (1.78) / avg. stars: 4.43)
        - Baked Bread Items (sentiment: slightly negative (1.71) / avg. stars: 4.56)
        - Small Desserts (sentiment: slightly negative (1.69) / avg. stars: 4.56)

Experiments: Predicting Star Rating from Sentiments

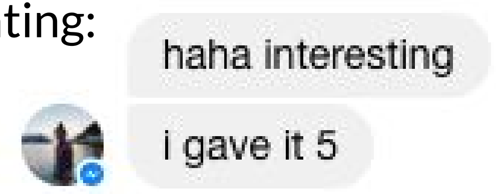
- Star Rating Prediction
  - Significant correlation for Star Rating and Sentiment Score on every cafe page tested
  - This simple regression model can be used to predict Star Rating from Sentiment Score
  - The prediction performs worst with low rated reviews as often the intercept of the regression is above 2
- Prediction takes two inputs
  - Cafe Yelp url
  - Unseen review text



Fun Experiment: Predicting star rating for friend's review

"Absolutely my favorite place to go to on weekdays (the lines go out the door on weekends) – coffee is consistently great and the sandwiches are fab"

- Predicted rating: 4.5
- Actual rating:



Technical Challenges

- Finding substantial training data
  - Attempt 1: webscraping
  - Attempt 2: Yelp dataset
- Figuring out appropriate K topics to use for LDA
- Interpreting “cloudy” subtopics
- Dealing with infrequent subtopics
  - Improvisation: Factored in weight of subtopic
- Web-scraping to acquire reviews given cafes
  - Could acquire first page of 20 reviews per cafe due to Yelp’s restrictions

Findings

- There were topics produced by the LDA model which were easily defined and fit our data well
- Rating of the subtopics could be predicted using sentiment score
- The combination of LDA and Sentiment Scoring could be a viable option for ranking what subtopics a business should improve

Potential Improvements

- Look at more categories of restaurants
  - Create multiple LDA models for different categories of food
  - How do their topics relate?
- Train on a larger and more representative dataset
  - The Yelp Academic Dataset was focused in Las Vegas so some groups of subtopics were biased to this geographic area
- Train the Sentiment Treebank on a more related set of data
  - Currently the treebank is based on a set of movie reviews
- More sophisticated star-rating prediction such as MLE

Conclusion

- LDA allowed us to extract implicit subtopics that standard text mining would neglect
- Sentiment treebanks are an improvement on traditional SA
- This combination allows us to
  - Learn what customers care about in their reviews
  - Identify priority points for restaurants
- Implications for Yelp – offer recommendation tools to its listed businesses

References

[1] J. Ramos. “Using TF-IDF to Determine Word Relevance in Document Queries”

[2] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. ”Indexing by latent semantic analysis.” Journal of the American Society of Information Science, 41(6):391407, 1990.

[3] D. Blei, A. Ng, and M. Jordan. ”Latent Dirichlet Allocation.” Journal of Machine Learning Research, 3:9931022, January 2003.

[4] J. Huang, S. Rogers, and E. Joo. “Improving Restaurants by Extracting Subtopics from Yelp Reviews”, Spring 2013.

[5] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, C. Potts “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank”