Topic modeling of job reviews
Andrew Hall

Abstract (92 words)
The goal of this project is to implement topic modeling to inform businesses of relevant clusters in a corpus of job reviews from an anonymous job review website. Textual data were preprocessed using NLTK and spaCy then topic modeling was conducted across LDA, NMF, and LSA approaches. The final LDA model extracted three discrete topics: Work life balance, Culture and teaming, and Company aspects. Additional adjective analysis compared *Pros* and *Cons* reviews to determine the most common adjectives present when employees are asked to write positive and negative things about their workplace.

Design
To determine the main latent topics in job reviews, I used a corpus of job reviews from the Blind app, an anonymous platform that enables verified employees to post reviews of their workplace. The corpus includes reviews from 25 different companies across multiple industries. Reviews were scraped by Harsh Bardhan Mishra using Python, Selenium and Beautiful Soup and include reviews posted between 2020 and June 2022. The analysis in this project was broken into three parts: an analysis of the general written reviews, an analysis of the reviews written when prompted for a "Pros," and the reviews written when prompted for "Cons." Course learnings were demonstrated through a) vectorization of the textual data using both the NLTK and spaCy packages, b) a comparison of count vectorization and TF-IDF approaches, c) a comparison of different dimensionality reduction techniques (LDA, NMF, and LSA) for topic modeling, and d) de-tangling of the extracted topics using both subjective interpretation and visualization libraries.

Data
The corpus of job reviews includes 131,409 individual documents, split into "General," "Pros," and "Cons" written by 43,803 individuals across 25 companies. All textual data columns included raw text in the form of short reviews akin to tweets. In preparation for topic modeling, the data had to be manipulated using Pandas and Numpy to convert between wide and long datasets, combine across datasets, ensure text data were in the correct string format, and establish the correct column headers.

Algorithms
Feature engineering and hyperparameter tuning:
- ***Stop words****:* English stop words were removed from the text data after tokenization to reduce the number of tokens that lacked signal.
- ***N-Grams****:* The number of permissible n-grams was tuned, from unigrams to trigrams. Initially, only unigrams and bigrams were included, but due to the prominence of the phrase "Work life balance" in multiple reviews, trigrams were included as well.
- ***Min DF****:* The minimum document frequency for a given token was tuned between 0 and 4,000 with 200 ultimately providing the most interpretable topics across techniques.
- ***Max DF****:* The maximum document frequency was tuned between .6 and .95, with .8 selected as the optimal balance point.
- ***Number of topics****:* The number of topics extracted by the topic modeling algorithms was tuned between 2 and 6, with 3 topics providing the best balance between ease of interpretation and comprehensiveness with the least amount of overlap.

Type of vectorizer:
- **CountVectorizer** and **TfidfVectorizer** were compared across three topic modeling algorithms. CountVectorizer resulted in the more interpretable topics.

Topic modeling algorithms:

- Three topic modeling algorithms were compared to find the most interpretable topics for use: **LDA**, **NMF**, and **LSA**. All three produced similar topics, but LDA consistently provided more discrete, separable topics more useful for business applications.

Final model

- The final model was selected out of the above options based on the interpretability and perceived separability of the generated topics. The LDA model produced the most discrete, interpretable topics (desirable for business applications), so the relatively higher computational load was deemed worthwhile. If this model were to be scaled up in the future, the NMF model may be deemed superior due to its relatively lighter computational load.
- The 3 topics (and related terms) were as follows:
    1. **Work Life Balance**: [*'life balance', 'work life', 'balance', 'life'*]
    2. **Culture and teaming**: ['culture', 'team', 'growth', good']
    3. **Company aspects**: [*'great', 'company', 'place', 'work'*]

Adjective analysis

- *Pros* and *Cons* reviews were compared using the spaCy pipe method. The most common adjectives for *Pros* included "good," "great," "smart," "nice" and "interesting," whereas the most common adjectives for *Cons* included "bad," "slow," "other," and "many."

NLP-relevant tools

- **spaCy** – Python library for text analysis and processing, targeted towards business applications. Arguably less flexible than NLTK but offers greater usability.
- **NLTK** – Python library for text analysis and processing, created for academic applications.
- **TF-IDF Vectorizer, Count Vectorizer** – Vectorizers for text tokens that extracts values for each document-term combination (TF-IDF) or simple counts (Count).
- **Gensim** – Python library for topic modeling, used here to create the requisite input matrices for LDA
- **Scattertext** – Visualization library to compare the language of two "populations in a dataset. Used here to compare the reviews written a "Pros" to those written as "Cons."
- **pyLDAvis** – Visualization library to visualize the topics extracted from an LDA model.

Communication

In addition to the slide presentation, the final model is visualized using pyLDAvis to highlight the key topics extracted. Additionally, scattertext was used to create an html with interactive visuals.