Semi-supervised Consistent Labeling of Short Text

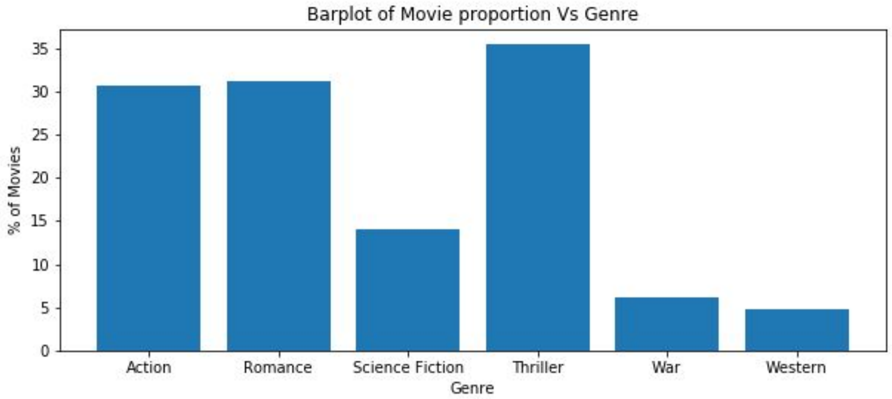
Motivation

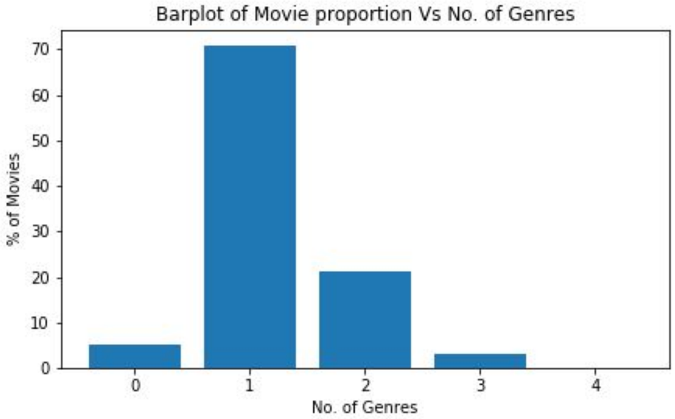
- Companies have access to large amount of free text such as customer reviews, call center transcripts, social media comments, tweets etc.
- Mostly unused due to them being **unlabeled**
- Resort to manual labeling of a small subset (~10%) of data for supervised learning
- Manual labeling is **tedious** and **subjective** (inconsistent labeling)
- Short text (< 100 words post-preprocessing) difficult for textual models

Project Objective

- Develop a semi-supervised method for consistent labeling of short text based on a dictionary
- **Proof of concept:** Label movie genres based on plot synopsis (multi-label data)
- **Dataset**: ~21500 movies across 6 genres (**Action, Romance, Science Fiction, Thriller, War, Western**) used

Exploratory Data Analysis





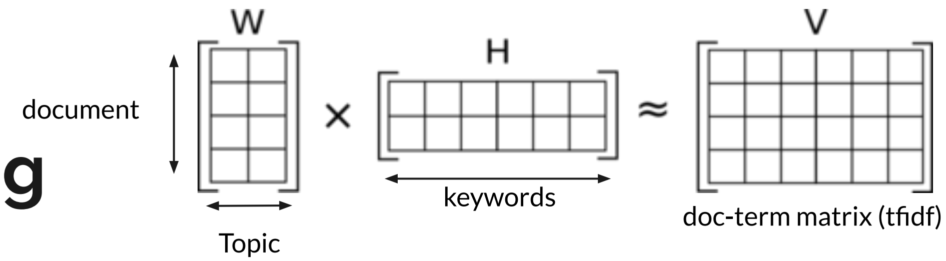
Cardinality: 1.225
Label density: 0.204
Mean Imbalance Ratio: 3.171

Approach

- **Step 1: Preliminary Topic Modeling**
  - Mix and match keywords from preliminary topic modeling to generate an initial dictionary with desired categories
  - Domain experts/business users just need to agree on this dictionary instead of manually labeling documents one-by-one
  - Keyword matching-based classification using initial dictionary as baseline for comparison
- **Step 2: Dictionary Enrichment**
  - Add related keywords from the corpus to each categories using **seeded topic modeling**
- **Step 3: Automatic Labeling**
  - Label top (most likely to belong in category) and bottom (least likely to belong in category) documents for each category with enriched dictionary using **matrix factorization**
  - Classify the remaining documents using standard classification models

Preliminary Topic Modeling

Non-negative matrix factorization (NMF)



Coherence score [1] to select best number of topics

Closer to human interpretation compared to perplexity score or log likelihood.

7 Topics: 0.335
8 Topics: 0.345
9 Topics: 0.362
10 Topics: 0.348
11 Topics: 0.347
12 Topics: 0.356
13 Topics: 0.365
14 Topics: 0.366
15 Topics: 0.368
16 Topics: 0.364
17 Topics: 0.375
18 Topics: 0.352

Topic 0: paris, **date**, **wedding**, apartment, **married**, **best_friend**, party, **romantic**, perfect, **sex**, **friendship**, ...
Topic 1: **terrorist**, hostage, bomb, president, nuclear, international, london, prevent, special, security, unit, **cia**, ...
Topic 2: band, gold, **outlaw**, **western**, **sheriff**, stranger, **west**, mexican, desert, **bandit**, **ranch**, gun, **texas**, …
Topic 3: **ii**, japanese, camp, **nazi**, prisoner, **jewish**, france, **wwii**, resistance, japan, allied, pilot, true_story, …
...
Topic 5: ship, **space**, captain, ... , pilot, survivor, **astronaut**, cargo, aboard, **moon**, ...
…
Topic 7: **high_school**, teacher, kid, class, **teenage**, crush, …, popular, **summer,** ...
Topic 8: **indian**, **cavalry**, tribe, india, **reservation**, **chief**, fort, white, territory, peace, raise, **west**, **wagon**, ...
…
Topic 13: **spy**, **assassin**, …, **master**, **cia**, **undercover**, chinese, **martial_art**, **ruthless**, north, **international**, soviet,...
…
Topic 15: **serial_killer**, **murderer**, late, **fbi_agent**, terrorize, **true_story**, stalk, **police_officer**, **justice**, **young_girl**, **maniac**, ...

Action, **Romance**, **Science Fiction**, **Thriller**, **War**, **Western**

1. Mimno D. et al. Optimizing semantic coherence in topic models, 2011

Seeded Topic Modeling

- Variant of standard topic modeling
- Keywords to converge around seeded topics instead of convergence based on latent distributions observed in the corpus
- Allow some control of the the resulting topics from model
- **Seeded LDA paper**: *"Importantly, we only encourage the model to follow the seed sets and do not force it. So if it has compelling evidence in the data to overcome the seed information then it still has the freedom to do so."* [2]
- **Python library**: GuidedLDA (https://github.com/vi3k6i5/GuidedLDA)
- Not used here because:
  - Latent Dirichlet Allocation not suitable for short text
  - Library built on LDA library, only allow integer-value matrix (tf matrix only, cannot use tfidf matrix)
- Develop custom version of seeded topic modeling based on NMF

2. Jagarlamudi J. et. al. Incorporating Lexical Priors into Topic Models, 2012

Normalized Pointwise Mutual Information (NPMI)

- 2 words are closely related if they tend to occur together in documents at a comparatively high frequency with respect to their own individual occurrence frequency in the corpus.
- Pointwise mutual information arguably the best statistical way to perform this normalization [3]:

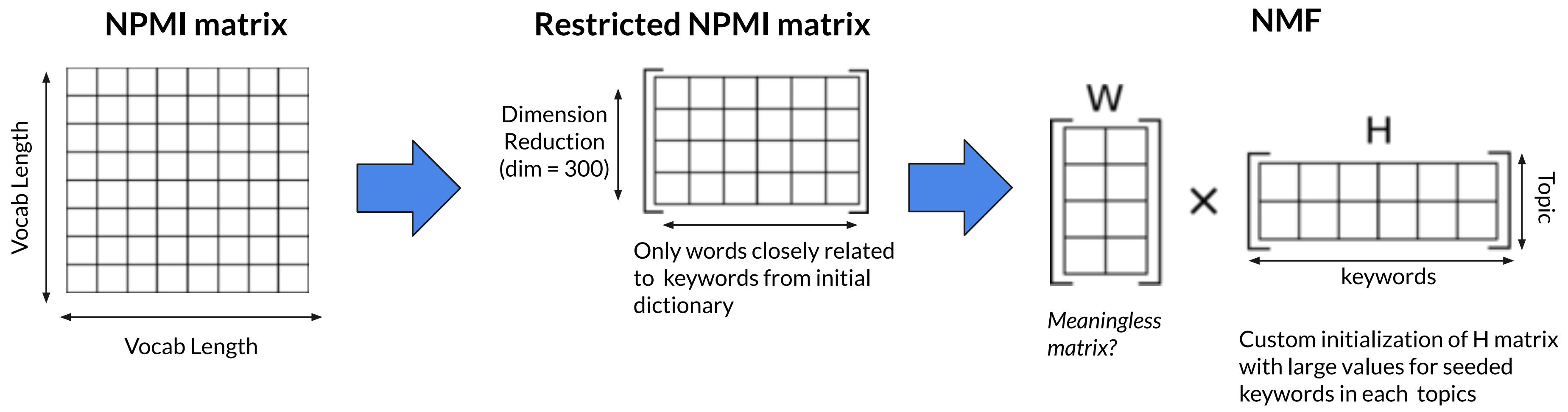
$$\log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\,p_Y(y)}\right)$$

- Normalized pointwise mutual information(NPMI) between -1 and 1
- Words with high NPMI should be in the same topic

3. Jan Van Eck N. et. al. How to Normalize Co-Occurrence Data? An Analysis of Some Well-Known Similarity Measures , 2009

Dictionary Enrichment

Get keywords from the corpus to converge around seeded topics (categories from initial dictionary) using custom version of seeded topic modeling with NMF



New words in Enriched Dictionary

Add top new keywords (highest topic score) from the H matrix to the dictionary

Action: fu, kung, martial ,china, ching, sword, swordsman, dynasty, karate, ...

Romance: crush, teacher, attend, music, dance, senior, torrid

Science Fiction: population, creature, crash, orbit, spacecraft, mutate, ...

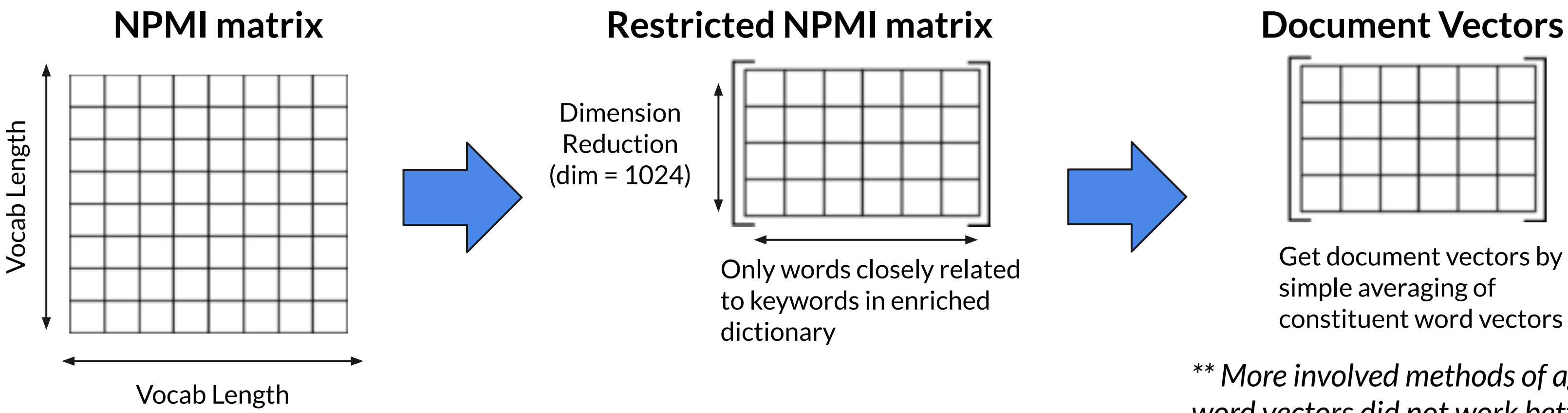
Thriller: mafia, undercover, cia, operative, hitman, mob, ..

War: command, raid, germany, occupied, partisan, lt, refugee, gestapo, reich, ...

Western: horse, mexican, posse, rancher, territory, frontier, herd, confederate

Automatic labeling (Part 1)

Compute a document vectors for each document



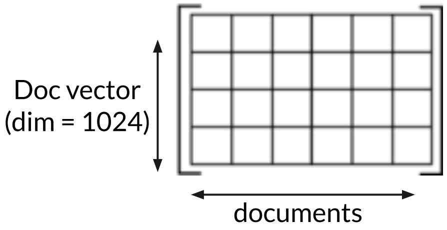
** *More involved methods of aggregating word vectors did not work better here* [4,5]

4. Arora S. et. al. A Simple but Tough-to-Beat Baseline for Sentence Embeddings, 2017
5. Rücklé A. et. al. Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations, 2018

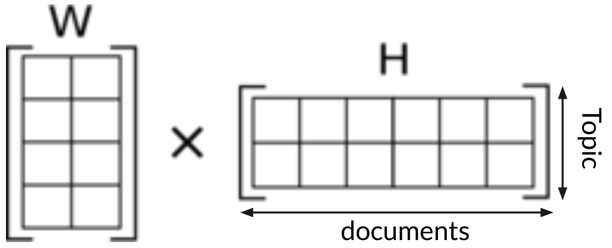
Automatic labeling (Part 2)

Label top (most likely to belong in category) and bottom (least likely to belong in category) documents for each category using document vectors

Document Vectors



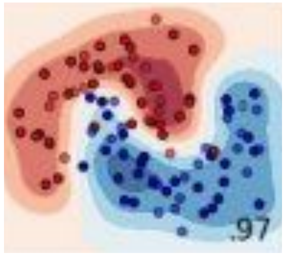
NMF



Meaningless matrix?

Custom initialization of H matrix with large values for seeded documents (contain most number of keywords) in each topics

Classification



Label top and bottom documents for each category using topic score from H matrix and classify remaining documents with these labels

Evaluation

- Naive bayes as a simple classifier for classifying the remaining documents to test this method
- Same 20% test data
- Compare with baseline (keyword matching of initial dictionary)
- Compare with supervised models (10% train data, mimic availability of manually labeled data)

Metrics	Baseline	This method	Simple NN	GBM	Random Forest
Precision	0.525	0.461	0.647	0.710	**0.774**
Recall	0.383	**0.547**	0.405	0.303	0.280
Macro F1	0.398	*0.461*	**0.484**	0.422	0.378

Close to performance of best supervised model

~ 6% improvementi n marco F1 over the baseline

Future Work

- Testing on more datasets to fine-tune method to be more stable and generalizable
- Better document vectors eg. embeddings from BERT language model?
- Some way to guide generation of better initial dictionary from preliminary topic modeling keywords

Thank you.