UNIVERSITY
OF
CALIFORNIA
MERCED

Multi-Class Sentiment Analysis with Clustering and Score Representation

Mohsen Farhadloo
Erik Rolland

mfarhadloo@ucmerced.edu

CONTENT

- Introduction
- Applications
- Related works
- Our approach
- Experimental results
- Questions

INTRODUCTION

- Sentiment analysis (opinion mining):
  - Computational and automatic study of people's opinions expressed in written language or text.
- Two types of information are in text data:
  - Objective information: facts.
  - Subjective information: opinions.
- The focus of sentiment analysis:
  - subjective part of text → identify opinionated information rather than mining and retrieval of factual information.
- Sentiment analysis brings together various fields of research: text mining, Natural Language Processing, Data mining.

APPLICATIONS

- Review summarizations.
- Review-oriented search engines.
  - Search for people's opinions: How do people think about iPhone 5s?
- Recommendation systems.
  - If you can do sentiment analysis, then the recommendation system can recommend items with positive feedback and not recommend items with negative feedback.
- Information extraction systems.
  - These systems focus on objective parts to extract factual information.
  - They can discard subjective sentences.
- Question-answering systems.
  - Different types of questions: definitional and opinion oriented questions.
- Both individuals and organizations can take advantage of sentiment analysis.

LEVELS OF SENTIMENT ANALYSIS

- Document level
  - Identify the opinion orientation of the **whole document.**
- Sentence level
  - Identify whether the sentence is **subjective or objective**.
  - Identify the opinion orientation of subjective **sentences**.
- Aspect level
  - Identify the **aspects** that the users are commenting on.
  - Identify the opinion orientation about each aspect.

RELATED WORKS

- Classical methods for aspect-based sentiment analysis address the problem in two steps: aspect identification and sentiment identification.
- Recently there are some works based on topic modeling that identify both aspects and sentiments simultaneously.
- Hu and Liu [2004]:
  - Aspect identification: frequent nouns and association mining for pruning.
  - Sentiment identification: find the closest adjective to the noun and use a lexicon for determining the opinion polarity.
- Gamon et al. [2005]:
  - Idea: using clustering over sentences to identify aspects.
  - Reported: none of the clustering algorithms produced satisfactory results.
  - Aspect identification: applying a weighting scheme to the frequent nouns.
  - Sentiment identification: Naïve Bayes classifier with bootstrapping from a small set of labeled data to a large set of unlabeled data.

RELATED WORK (CONT.)

- Goldensohn et al. [2008]:
  - Aspect identification:
    - Dynamic aspects (string-based and specific aspects): using frequent nouns.
    - Static aspects (generic and coarse-grained aspects): designing classifiers for each one of them using hand-labeled sentences.
  - Sentiment identification:
    - Computing a single score for each term: starting from a seed set of words with arbitrary scores and propagate them to the other words.
    - Compute a score for each sentence and also for its neighbors.
    - Design maximum entropy classifiers for positive and negative sentences.

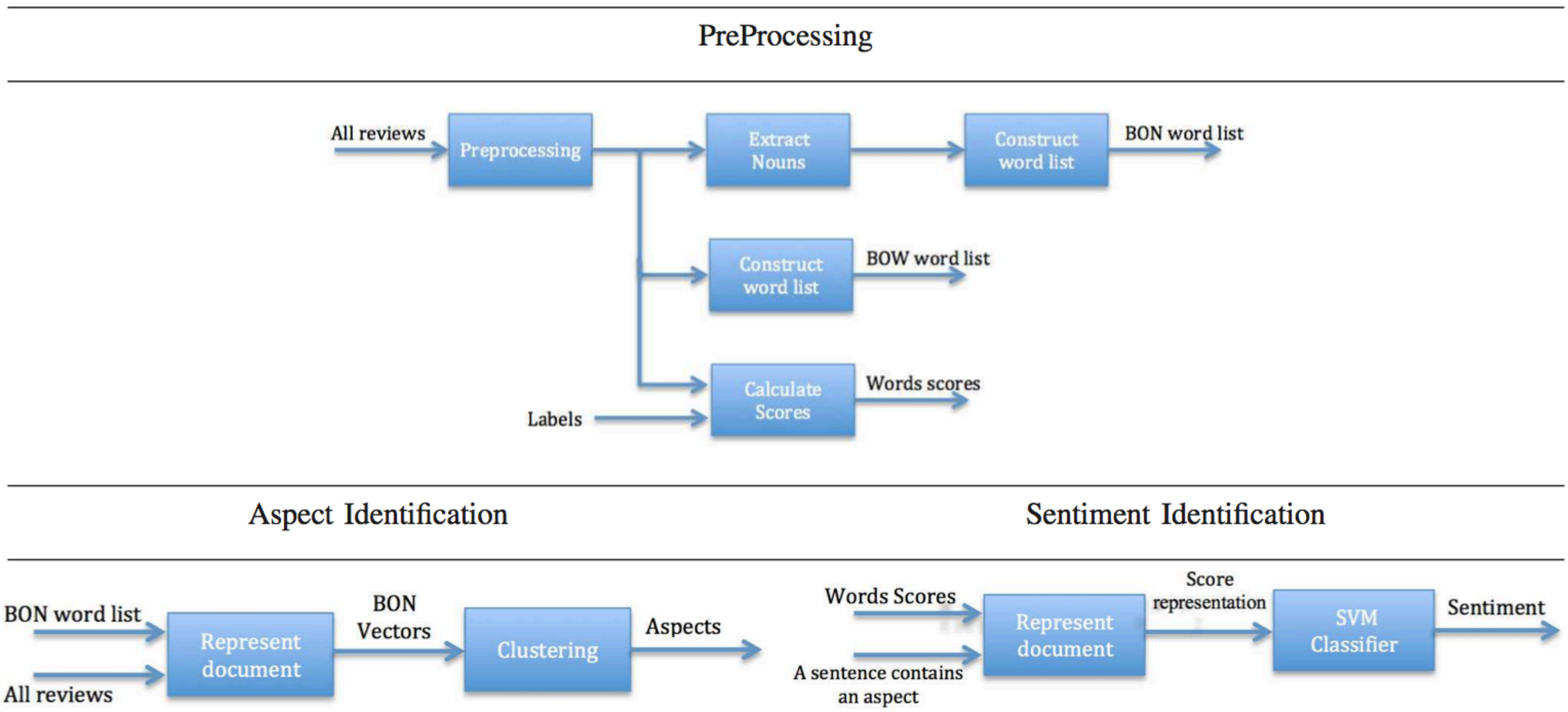
RELATED WORK (CONT.)

- There are papers that have reported improvement in sentiment analysis using domain ontology.
- Concept-based approaches
  - Use Web ontologies.
  - Represent each sentence with bag-of-concepts instead of bag-of-words.
  - Each concept is just a multi-term expression.
  - For sentiment identification they use a lexicon of concepts that contains the affective labels of concepts (SenticNet).

OUR CONTRIBUTIONS

- **Aspect identification** with sentence **clustering** using **Bag of Nouns** instead of Bag of Words.
- Proposing **score representation** as feature set for classification.
  - It is based on positivity, neutrality and negativity of terms that we learn from data.
- Considering the sentiment identification as a **three-class** classification problem rather than two-class problem.
- Using this new feature set for classification we improve the performance of state-of-the-art 3-class sentiment classification of sentences by 20% in terms of average f1 score.

SYSTEM DIAGRAM



ASPECT IDENTIFICATION

- Using clustering to find similar sentences. **It is likely that similar sentences are about similar aspects.**
- For sentence clustering the method that we use for representing each sentence is important.
- The major reason that regular clustering algorithms did not work (Gamon et al [2005]) is the lack of proper method to represent each sentence.
- Sentences representation
  - BOW representation: considers all terms in the sentence.
  - BON representation: considers only nouns of the sentence.

ASPECT IDENTIFICATION (CONT.)

- Consider three sentences
  - The **screen** is great (s1).
  - The **screen** is awful (s2).
  - The **voice** is great (s3).
- BOW vs BON representations

TABLE I: BOW representation

Word list	s1	s2	s3
screen	1	1	0
great	1	0	1
awful	0	1	0
voice	0	0	1

TABLE II: BON representation

Word list	s1	s2	s3
screen	1	1	0
voice	0	0	1

- In BOW representation s1 differs in two positions from s2 and s3.
- In BON representation s1 and s2 that are about **screen** are similar.

SENTIMENT IDENTIFICATION

- Machine learning approach sees the sentiment identification problem as a classification problem.
- Make use of manually labeled training data.
- Two major tasks in designing a classifier
  - Feature extraction: come up with a set of features that represents your problem properly.
  - Classifier selection: choose a classifier among KNN, Naïve Bayes, SVM, Maximum Entropy.
- Our approaches are related to feature extraction steps.
- Support Vector Machines are widely used in text classification. We use SVM as well.

BOW REPRESENTATION

- BOW representation
  - Construct a vocabulary list using all the documents in the corpus.
  - Represent each document with a vector indicating the existence of terms.
  - Different weighting schemes: binary, term occurrence, tf-idf.
  - We compute the tf-idf as:

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{1 + \max_{w \in d} \mathrm{f}(w,d)} \tag{1}$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{2}$$

$$\text{tf-idf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D) \tag{3}$$

SCORE REPRESENTATION

- Compute three scores for each term in the vocabulary list.

$$s_i^+ = \frac{f_i^+}{f_i^+ + f_i^0 + f_i^-},$$
$$s_i^0 = \frac{f_i^0}{f_i^+ + f_i^0 + f_i^-},$$
$$s_i^- = \frac{f_i^-}{f_i^+ + f_i^0 + f_i^-}$$

- The scores of each sentence are the weighted sum of the scores of its terms.

$$S^+ = \sum_{i \in \mathbf{x}} w_i s_i^+,$$
$$S^0 = \sum_{i \in \mathbf{x}} w_i s_i^0,$$
$$S^- = \sum_{i \in \mathbf{x}} w_i s_i^-$$

- Represent each sentence with a 3-dim vector

$$\mathbf{S} = [S^+, S^0, S^-]^T$$

SCORE REPRESENTATION (CONT.)

- The scores of each term are not some arbitrary scores assigned to them.
- These scores reflect the positivity, neutrality and negativity of the terms in the related context.
- Instead of working with high-dim vectors we work with 3-dim vectors.
- We use SVM classifier to classify the sentiment of each sentence.
- In basic SVM the goal is to find a hyper plane that separates the two classes and its distance to the nearest point in each side is maximized.

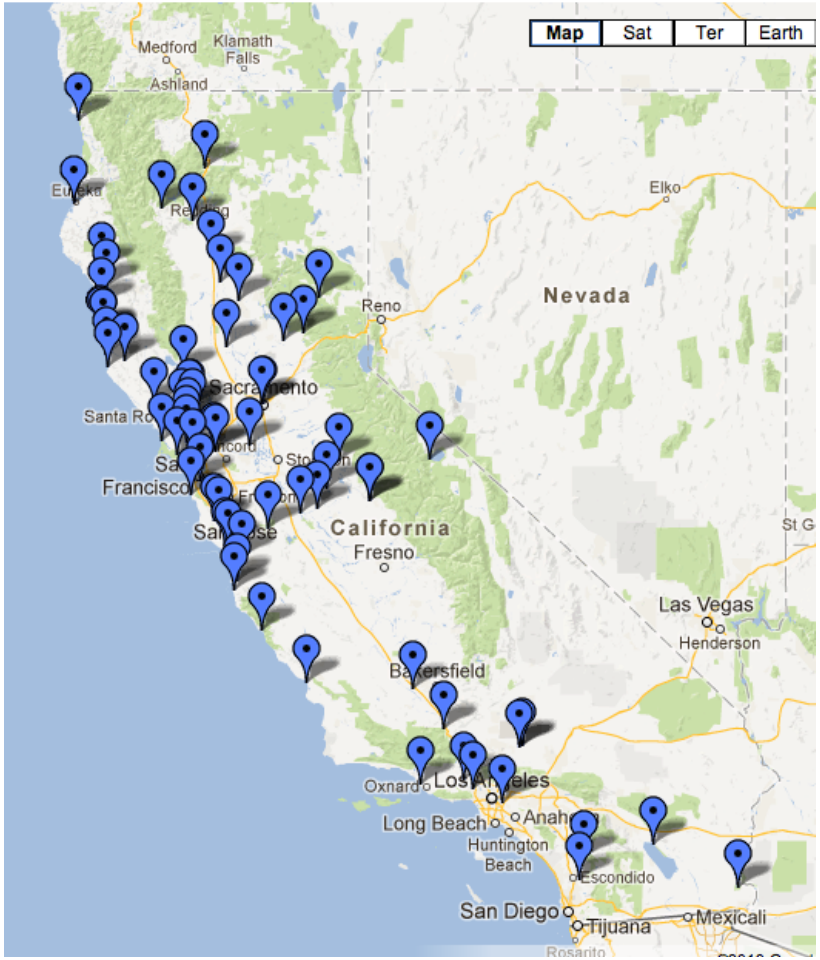
SCORE REPRESENTATION (CONT.)

Table VIII: The computed scores for some of the terms.

Term	Positiveness (s_i^+)	Neutralness (s_i^0)	Negativeness (s_i^-)
beach	0.569	0.276	0.154
not	0.411	0.301	0.288
great	0.937	0.029	0.033
beauti	0.936	0.037	0.028
view	0.760	0.154	0.087
place	0.821	0.098	0.082
trail	0.467	0.386	0.147
waterfal	0.575	0.365	0.060
big	0.572	0.380	0.048
walk	0.613	0.31	0.077
time	0.5	0.318	0.182
area	0.637	0.298	0.064
fall	0.517	0.392	0.092
lot	0.483	0.408	0.108
hike	0.691	0.227	0.082
nice	0.843	0.065	0.093
state	0.408	0.531	0.062
site	0.357	0.452	0.191
day	0.469	0.469	0.062
park	0.219	0.563	0.219
walk	0.573	0.369	0.058
water	0.489	0.304	0.207
campground	0.625	0.273	0.102
easi	0.883	0.078	0.039
short	0.711	0.276	0.013
good	0.784	0.122	0.095
ocean	0.548	0.384	0.068
night	0.347	0.389	0.264
redwood	0.696	0.290	0.014
well	0.794	0.143	0.063

EXPERIMENTAL RESULTS

- Data
  - Reviews from TripAdvisor.com.
  - Reviews of 6 state parks with a beach on the Pacific Ocean.
  - 992 positive, 992 neutral and 421 negative sentences.
  - Labels have been provided manually.
  - 21 sentences from each category as test set.
  - Discard terms that occur fewer than 5 times.
  - Size of word list for BOW 662 and for BON 340.



BOW VS. BON FOR CLUSTERING

- BON leads **to lower dimensional vectors**.
- Performance measure is **normalized recall**: it measures what fraction of a desired list the clustering algorithm covers.
- We use the list of **all nouns in our corpus** as the desired list.
- After clustering some terms are selected from each cluster as **representative terms** using the centroid of the cluster.

TABLE V: Normalized recall ($\tilde{r}$) for different number of clusters.

k	5	10	15	20	25
BOW $\tilde{r}$	0.18	0.21	0.16	0.21	0.26
BON $\tilde{r}$	0.29	0.32	0.39	0.37	0.34

ASPECT IDENTIFICATION VIA SENTENCE CLUSTERING

- Using BON approach, the extracted terms are more meaningful and closer to the desired list.
- Latent Semantic Analysis reduces the unrelated terms from the clustering process.

TABLE IV: Aspect identification with clustering the sentences. The first row is the desired list of aspects. The rest are the extracted representative terms after clustering using BOW and BON with/without LSA.

desired_list	beach, park, view, place, trail, waterfal, walk, sur, time, state, area, fall, lot, site, day, hike, camp, water, campground, night, redwood, nice, highway, spot, way, pfeiffer, road, peopl, famili, year, sand, bay, coast, fee, kid, trip, car, hour, ranger, river, tree, stop, wave, mile, quot, bit, rock, minut, sunset, shower
BOW	beach, <u>beat</u>, bike, <u>cool</u>, entranc, <u>great</u>, <u>love</u>, <u>need</u>, <u>nice</u>, <u>not</u>, phone, <u>pick</u>, <u>reach</u>, roost, <u>share</u>, sunset, surfer, wave
BON	bathroom, beach, coffe, coupl, day, end, famili, fun, kid, minut, park, princeton, site, state, surfer, time, town, trail, view, water
BOW with LSA	beach, <u>big</u>, camp, day, fall, hike, lot, <u>not</u>, park, place, site, state, sur, time, view, walk, waterfal
BON with LSA	area, beach, camp, campground, coast, day, julia, kid, lot, oak, park, pfeiffer, poison, redwood, sand, site, summer, time, wave, way, weekend

EFFECT OF LATENT SEMANTIC ANALYSIS

- In order to address the synonymy problem we investigated the effects of Latent Semantic Analysis.
- By virtue of dimension reduction, it is possible for documents with somewhat different profiles of terms usage to be mapped into the same vector of factor values.
- It is possible to find a lower dimensional space that gives better performance than using the original high dimensional data in terms of the coverage of the desired list of aspects.

TABLE III: Normalized recall ($\tilde{r}$) for different dimension in the low-dimensional space using LSA.

svd-dimension	2	3	5	10	30	50	70	90	110	130	150	170	200
BOW $\tilde{r}$	0.06	0.08	0.08	0.24	0.29	0.16	0.20	0.18	0.18	0.18	0.22	0.18	0.22
BON $\tilde{r}$	0.06	0.06	0.18	0.29	0.33	0.27	0.41	0.29	0.41	0.29	0.31	0.41	0.29

SENTIMENT IDENTIFICATION

- 3-class classification problem.
- We adopted the one-against-all scheme i.e. three binary classifiers (one for each class).
- Each classifier is a non-linear binary classifier with Radial Basis Functions.
- The parameters are chosen through 5-fold cross-validation.
- We have compared sentiment identification using BOW as features to score-representation as features.
- We have compared term-occurrence and tf-idf weighting schemes.

RESULTS

TABLE VI: Performance of sentiment classification with BOW as feature set. The table compares the tf-idf weighting scheme with term occurrence scheme.

	occurrence			tf-idf		
	precision	recall	f1-score	precision	recall	f1-score
Negative class	0.40	0.29	0.33	0.33	0.24	0.28
Neutral class	0.53	0.76	0.63	0.46	0.76	0.57
Positive class	0.67	0.57	0.62	0.77	0.48	0.59
Average performance	0.53	0.54	0.53	0.52	0.49	0.48

TABLE VII: Performance of sentiment classification with score representation as feature set. The table compares the tf-idf weighting scheme with term occurrence scheme.

	occurrence			tf-idf		
	precision	recall	f1-score	precision	recall	f1-score
Negative class	0.63	0.57	0.60	0.64	0.43	0.51
Neutral class	0.58	0.71	0.64	0.57	0.57	0.57
Positive class	0.89	0.76	0.82	0.77	0.81	0.79
Average performance	0.70	0.68	0.69	0.64	0.63	0.63

Precision = the fraction of retrieved instances that are relevant.
Recall = the fraction of relevant instances that are retrieved.

$$F = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

RESULTS (CONT.)

- Classifying the **negative sentences is more challenging** than the positive and neutral sentences.
- Better f1-score would be achieved with **term occurrence** as weighting scheme.
- To the best of our knowledge the best result reported in the literature for 3-class sentiment classification is with average f1-score of 49%.
- The state-of-the-art in 3-class sentiment analysis can be improved more, by selecting better feature set.
- Using our proposed **score representation** as feature vectors, the average f1-score that we achieved is 69%.

QUESTIONS

- Thank you.