Spotting Sentiments with Semantic Aware Multilevel Cascaded Analysis

Despoina Chatzakou, Nikolaos Passalis, Athena Vakali

Aristotle University of Thessaloniki

Big Data Analytics and Knowledge Discovery, 2015

Outline

Outline

Content generated on the Web

- Numerous individuals express opinions and feelings in the Web.
- Continuous use of popular Social Networks and Web 2.0 technologies has pushed the need for understanding crowd's opinions.

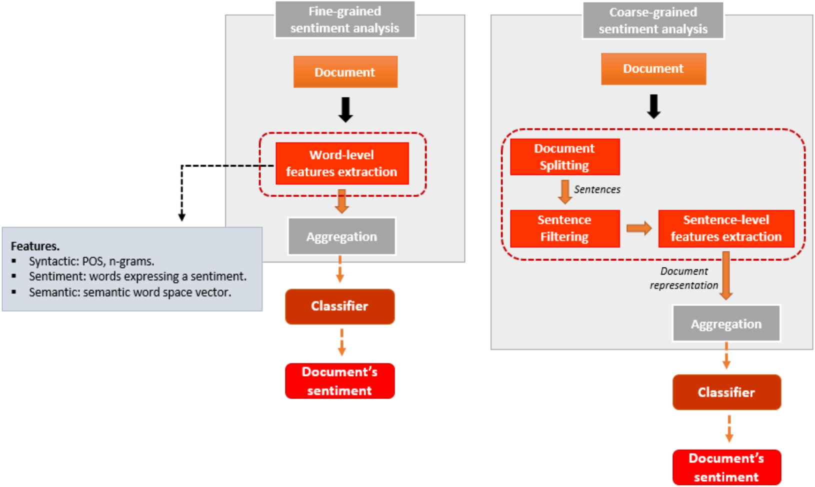
Laptop X is more expensive than laptop Y.

Laptop X is the best laptop in the market!

Capturing sentiment out of textual resources (1/2)

- Machine learning is a popular approach for spotting the sentiment expressed in documents (i.e. document-based sentiment analysis).
- Typically, document-based sentiment analysis processes operate at a particular level:
  - Fine-grained approach: word-level process (i.e. sentiment-based, syntactic-based, semantic-based).
  - Coarse-grained approach: sentence-level process.
- Sentiments' extraction only either out of separated sets of words or at lined sentences leads to information loss.

Capturing sentiment out of textual resources (2/2)



Our goals

- **Given:** a set of documents D; a sentiment label and a representation for each document;
- **Predict:** the expressed sentiment for any new document.

G1. Exploit effectively diverse information from each individual sentence of a document.

G2. Design an effective approach for combining information arising from different text-levels.

Outline

Outline

Recursive Neural Tensor Network

- **RNTN:** suitable for capturing the compositional effects in sentences.
- It learns a semantic vector space & generates a sparse tree to represent a document at different levels.
- Each sentence is represented with a **semantic information vector**.
- It can classify individual sentences and produce a **sentiment probability distribution vector**. **In our case:**
  - A 5-value sentiment probability distribution vector is produced (1 - very negative, 2 - negative, 3 - neutral, 4 - positive, 5 - very positive).

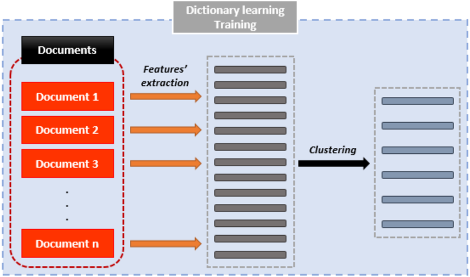
Sentiment probability distribution: $sent_i(s)$, where $i = 1, ..., 5$ and s a sentence of a document.
Semantic space vector: $vec(s)$.

Go

Outline

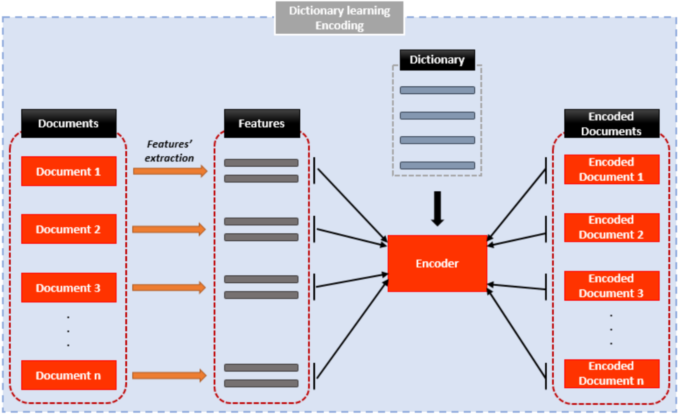
Dictionary learning (1/2)

- Use of a clustering process to construct a dictionary.
- Clustering is applied on the feature vectors that represent the documents' sentences.
- Each "word" of the dictionary corresponds to a set of similar feature vectors.
- K-means and variants are usually used to perform the clustering.



Dictionary learning (2/2)

Encoding of a new document.



Outline

Problem definition

Multilevel sentiment analysis

- **Given:** a training dataset D, a word-level feature extractor $f(w)$, a sentence-level feature extractor $g(s)$, and a sentiment label for each document $t_i \in \{pos, neg\}$;
- **Extract:** the word level and the sentence level features of each document d.
- **Predict:** the sentiment of any new document $d_{test} \notin D$.

Outline

Outline

Word level analysis (1/3)

Step 1. Features extraction: word to vector mapping.
Step 2. The document-level vectors aggregation via a weighting scheme (e.g. term-frequency tf).

Step 1. Features extraction: word to vector mapping.

- A word w is modeled as a vector $v \in \mathbb{R}^k$, where k equals the size of the used dictionary.
- All the elements of v are zero except for the one that corresponds to the word w.

Word-level features.

- Both Bag of Words (BoW) and Naive Bayes bigrams (NB) features were examined.

Word level analysis (2/3)

Step 2. Word vectors aggregation.

- All the vectors of the words in a document are combined into one that describes the whole document.

Word-level feature extractor

The word-level feature extractor f maps each word w of a document d to a vector $f(w) \in \mathbb{R}^l$ (l is the dimensionality of the (output) vector).

Word level analysis (3/3)

Aggregation Example (BoW features).

- Dictionary: $\{bag, of, words\}$.

Word	Vector
bag	(1,0,0)
of	(0,1,0)
words	(0,0,1)

- Binary weighting scheme.
  - Vector of the phrase *bag bag words*: (1,0,1).
- Term-frequency weighting scheme.
  - Vector of the phrase *bag bag words*: (2,0,1).

Outline

Sentence level analysis (1/6)

Step 1. Features extraction.
Step 2. Aggregation phase under a weighting scheme.

Step 1. **Features extraction.**

- Use of the RNTN model as a sentence-level feature extractor.
  - Sentiment distribution vector, $sent(s)$.
  - Semantic vector, $vec(s)$.
- Other methods could also be used for extracting the sentiment distribution and the semantic vector for each sentence.
- RNTN was the first to achieve 85.4% accuracy on binary sentence-level classification.

Sentence level analysis (2/6)

Step 2. Aggregation phase.

Two approaches were examined:

- Sentiment center estimation.
- Semantic center estimation.

Sentence level analysis: Sentiment center estimation (3/6)

Sentiment center vector.

$$sent_{center}(d) = \sum_{s \in d} sent(s)/|d|$$

where $|d|$ is the number of sentences of document d.

Sentiment variance vector.

$$sent_{var}(d) = \sum_{s \in d} (sent(s) - sent_{center}(d))^2/|d|$$

which contains the squared differences from the document's center.

Sentence level analysis: Semantic center estimation (4/6)

Semantic center vector.

$$vec_{center}(d) = \sum_{s \in d} vec(s)/|d|$$

where $|d|$ is the number of sentences of document d.

Semantic variance vector.

$$vec_{var}(d) = \sum_{s \in d} (vec(s) - vec_{center}(d))^2/|d|$$

which contains the squared differences from the document's center.

Sentence level analysis: Semantic CenterBook (5/6)

- Builds on vectors that merge semantically similar sentences.

CenterBook process.

Given a training set of documents $D = \{d_1, d_2, ..., d_n\}$
Do

- Split all documents into a set of sentences.
- Clustering the set of all sentences appearing in D.

Done
Output: a collection of clusters.

Sentence level analysis: Semantic CenterBook (6/6)

- Each sentence in a document is represented by its nearest cluster.
- The overall document is modeled by the set of centroids.

Sentence encoding function, $h(s)$.

$$h(s) = y_i = \begin{cases} 1, & i == arg_j min(||c_j - vec(s)||_2^2) \\ 0, & otherwise \end{cases}$$

where y_i is the i-th element of $y(s)$ vector.

Document representation, CenterBook.

$$code(d) = \sum_{s \in S_d} h(s)$$

where s is each sentence of a document d and S_d the set of all sentences of document d.

Outline

MultiSpot pipeline (1/2)

Given a document d

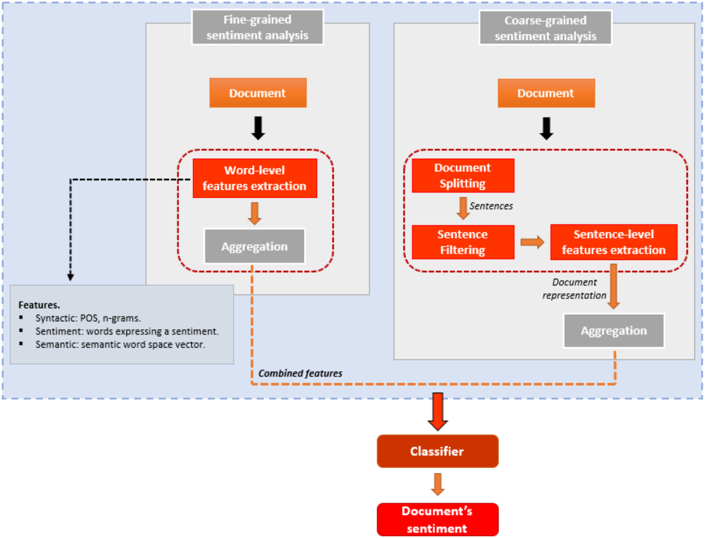
Do

- **Phase 1.** Word level analysis (*Fine-grained word features*).
- **Phase 2.** Sentence level analysis (*Coarse-grained word features*).
- **Phase 3.** Combination of word and sentence level aggregated features.

Done

Spot the document's sentiment.

MultiSpot pipeline (2/2)



Outline

Outline

Dataset overview

- Both datasets contain movie reviews collected from the Internet Movie Database (IMDB).

Dataset	# Reviews	Pos / Neg
Large Movie Review Dataset (IMDB)	50k + 50k (unlab)	50% - 50%
Polarity dataset v2.0 (RT-2k)	2.000	50% - 50%

Outline

Fundamental Characteristics

- **Extracted word-level features.**
  - BoW (*top 10.000 unigrams*) & NB bigrams.
  - Weighting scheme: Term-frequency for the IMBD dataset, Binary weighting for the RT-2k.
- **Extracted sentence-level features:** Sentiment distribution & Semantic vector based on RNTN model.
- **Clustering:** k-means algorithm for 15 iterations, 10 times repetition of the clustering process and selection of the minimum energy configuration.
- **Classification:** linear SVM, selection of best SVM model based on 10-fold cross validation.

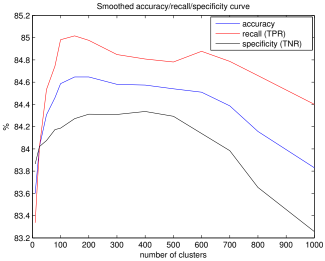
Outline

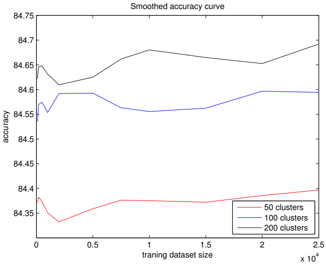
CenterBook evaluation

How is the classification quality affected by:

Q1. the number of the clusters?

Q2. the size of available training data?





Results for the IMDB dataset.

Go

Evaluation of sentence-based approaches

Parameters.

- Use of 200 clusters for the IMDB dataset and 100 clusters for the RT-2k dataset.

Features	**IMDB**			**RT-2k**		
	Accuracy	**Recall**	**F1**	**Accuracy**	**Recall**	**F1**
Rule-based	76.21	56.36	70.32	59.7	19.80	32.95
Sentiment Center	84.02	80.97	83.52	83.30	86.10	83.75
Sentiment Center (var)	84.01	80.96	83.51	83.05	86.30	83.58
Semantic Center	84.90	83.45	84.68	**85.10**	86.80	**85.35**
Semantic Center (var)	85.27	83.53	85.01	84.90	85.50	84.99
CenterBook	84.76	85.05	84.80	83.15	84.10	83.34
Sentiment Center (var) + Semantic Center	85.27	83.53	85.01	85.05	85.40	85.10
Sentiment Center (var) + Semantic Center + CenterBook	**85.35**	84.30	**85.20**	84.85	87.10	85.18

- Approaches that involve semantic features yield better classification results.

Evaluation of the MultiSpot method using BoW and NB bigrams features (1/2)

Features	IMDB			RT-2k		
	Accuracy	**Recall**	**F1**	**Accuracy**	**Recall**	**F1**
BoW	87.77	88.01	87.80	87.15	88.40	87.31
BoW + Sentiment Center	88.99	89.01	88.99	87.45	88.70	87.60
BoW + Semantic Center (var)	89.36	89.18	89.34	88.20	89.60	88.36
BoW + CenterBook	89.29	89.26	89.29	88.85	90.80	89.06
BoW + Sentiment Center + Semantic Center (var)	89.38	89.22	89.36	88.25	89.70	88.42
BoW + Sentiment Center + Semantic Center (var) + CenterBook	**89.48**	89.19	**89.45**	**89.05**	91.50	**89.31**

- 1.71% improvement for the IMDB dataset and 1.9% improvement for the RT-2k dataset.

Features	IMDB			RT-2k		
	Accuracy	**Recall**	**F1**	**Accuracy**	**Recall**	**F1**
NB bi	91.43	92.13	91.49	89.45	90.80	89.59
NB bi+ Sentiment Center	91.72	91.77	91.73	90.00	91.30	90.13
NB bi+ Semantic Center (var)	91.76	91.49	91.73	90.90	91.90	90.99
NB bi+ CenterBook	91.72	91.90	91.73	**91.30**	93.50	**91.49**
NB bi+ Sentiment Center + Semantic Center (var)	**91.78**	91.53	**91.76**	90.85	91.90	90.95
NB bi+ Sentiment Center + Semantic Center (var) + CenterBook	91.60	91.33	91.58	90.65	93.10	90.87

- 0.35% improvement for the IMDB dataset and 1.85% improvement for the RT-2k dataset.

Evaluation of the MultiSpot method using BoW and NB bigrams features (2/2)

Observations.

- The combination of word & sentence level features improves the classification accuracy.
- The quality of the word-level features significantly affects the overall classification accuracy.

Friedman test.

- It is used to explore differences in treatments across multiple test attempts.
- **Null hypothesis:** Multilevel cascaded sentiment analysis does not increase the accuracy of the baseline (BoW / NB bigrams) classifier ➔ Rejected.

Comparison of MultiSpot with state-of-the-art approaches

Method	IMDB	RT-2k
MultiSpot method		
NB bi + CenterBook	91.72	**91.30**
NB bi + Sentiment Center (var) + Semantic Center (var)	**91.78**	90.85
State-of-the-art approaches		
Full + Unlabeled + BoW (Maas2011)	88.89	88.90
BoW SVM (Pang2004)	-	87.15
tfΔidf (Martineau2009)	-	88.10
Appr. Taxonomy (Whitelaw2005)	-	**90.20**
Word Repr. RBM + BoW (Dahl2012)	89.23	-
NB SVM bigrams (Wang2012)	91.22	89.45
Paragraph Vector (Le2014)	**92.58**	-

- **RT-2k:** exceeds the existing classification accuracy by 1.1%.
- **IMDB:** surpasses the existing classification accuracy for 0.8% (not combined with the paragraph vector method).

Outline

Conclusions

G1. Exploit effectively diverse information from each individual sentence of a document.

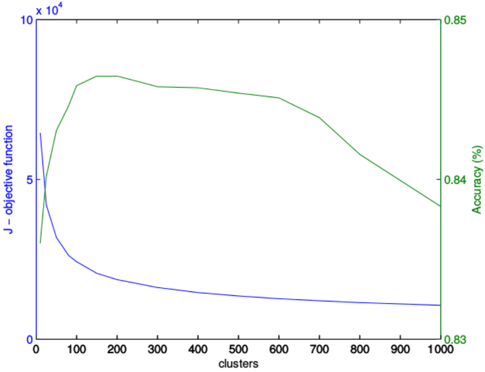
- Exploitation of sentiment and/or semantic information via the center-based methodologies.

G2. Design an effective approach for combining information arising from different text-levels.

- MultiSpot is an affective pipeline which combines both word and sentence level information.

Questions?

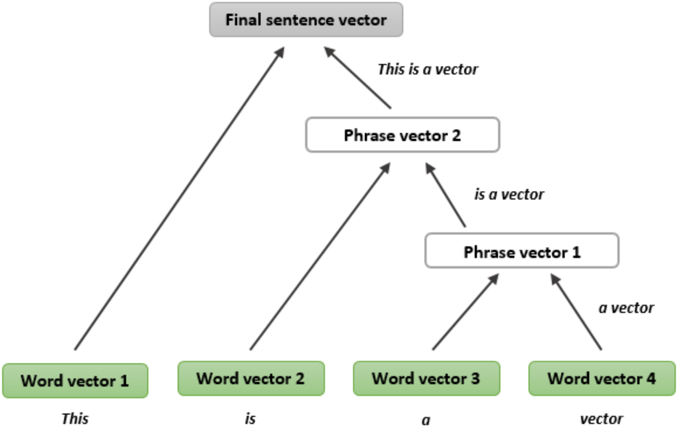
Evaluation of the k-means objective function.



Results for the IMDB dataset.

Back

Appendix: Tree structure



Back