[Proposal] Discovering domain-specific, sentiment-driven text-visual detector in microblog images

Presentor: Aaron Kung
Rongrong Ji
Motivation

• Microblog data often contains multimedia data
• Text data in microblog is usually
  – Extremely sparse
  – With strong sentiment orientation
  – > We cannot use traditional joint text-image methods
• We wish to incorporate such text data with images to derive representative properties in an image-tweet
• No such works associating social media text and image have yet been proposed
Project Goal/Contribution

• Goal 1: Develop a method to extract meaningful and interpretable sentiment-driven text-visual detectors from texted images in microblogs
  – Qualitatively show the detectors are describable
  – Empirically show that the detectors provide more compact and effective representation for sentiment analysis

• Goal 2: Release the first microblog text-image dataset for different domains and corresponding text-visual detectors
Challenges

• **Challenge 1:** How to derive joint text-visual attribute in social media under the notion it’s
  – Biased with personal opinions?
  – Noisy and sparse text?

• **Challenge 2:** How to make such text-visual attribute human-interpretable?

• **Challenge 3:** How to determine top k text-visual attributes from a set of candidates?

• **Challenge 4:** Are there occasions where such attributes are useful over existing methods?
Framework

Preprocessing
- Data collection, labeling

Training
- Derive candidate text-visual attributes
- Rank candidate attributes
- Representative text-visual attributes for the training domain

Testing
- Sentiment Analysis
- Used as detectors for other CV tasks
**Framework**

- **Preprocessing**
  - Data collection, labeling

- **Training**
  - Derive candidate text-visual attributes
  - Rank candidate attributes
  - Representative text-visual attributes for the training domain

- **Testing**
  - Sentiment Analysis
  - Used as detectors for other CV tasks

*(Twitter data is lexically distinct under different emotional state and different topics)*

- Select datasets from domains with balanced sentiment
- Manually label instances according to sentiment polarity
Framework

Preprocessing

Data collection, labeling

Derive candidate text-visual attributes

Representative text-visual attributes for the training domain

Rank candidate attributes

Training

Sentiment Analysis

Used as detectors for other CV tasks

Testing

• Associating text-visual attributes using information graph formulation
• Each joint attribute is equivalent to mining frequent subgraphs
• Use common frequent subgraph algorithms to find candidates
Framework

Preprocessing

Data collection, labeling

Derive candidate text-visual attributes

Rank candidate attributes

Representative text-visual attributes for the training domain

Training

Sentiment Analysis

Used as detectors for other CV tasks

Testing

- There are any structurally similar subgraphs (approximately isomorphic)
- We eliminate similar subgraphs and obtain a diverse and representative list of attributes
Framework

Preprocessing

Data collection, labeling

Derive candidate text-visual attributes

Rank candidate attributes

Representative text-visual attributes for the training domain

Training

Testing

Sentiment Analysis

Used as detectors for other CV tasks

• Joint modality detectors should work well in text mining tasks as well as in visual mining
• The attributes should be compact and efficient
Data Collection

- **Data Source:** *Twitter/Twicaps*
  - Each instance is a tweet with corresponding image URL (flickr, twitpics, yfrog, etc.)
  - Every instance has a corresponding image

- **Domain selection**
  - Many domains are extremely unbalanced in emotion (e.g. movies)
  - Filter by hashtags or keywords to get a group of instances with more evenly balanced emotions based on six basic human emotions (anger, happy, surprise, etc.-through manual checks)
  - Select domains (each about 1000-10000 instances):
    - Weather (lightning, sky, rain)
    - Location specifications (place, event, meeting, room)
    - Neutral expressions (crazy, hell, insane)
    - Scenery expressions (dark)
    - Sensory expressions (hot, cold, burn, freeze)
    - Politics (Obama, Romney)- internal evaluation only
Data Labeling

- Labeling resource: Manual/Mechanical Turk
- Labeling workflow:

  - Pos/neg/neutral
  - Happy/angry/sad

Label types depend on the domain

- Discard
- Golden Standard
- Labeled Dataset
Text/Visual Attribute Generation

• Text attribute generation (feature value-freq):
  – Bag of words text attributes commonly used in Twitter sentiment classification [1,2,3]
  – BoW features (Unigram) + other features
    • # of capitalized characters
    • # of pos/neg emoticons
    • # of hashtags, URL
    • BoW is empty: label “null”

• Visual attribute generation (feature value-sparse [0,1]): use general Classmes detectors[4]
Sub-goal: Discover joint modal attributes

• Fundamental concepts
  – Text attributes are used in assistance
  – Low frequency data attributes should be less important
  – High frequency attributes without label distinction is not informative
  – High frequency attributes with label affinity should be given more weights
Idea 1: Generate information graph

• Each feature should be considered as a node: text-unigram; image-visual attribute

• An edge between two features reflects how likely both occur in affinity to a given sentiment label for an instance
Idea 1: Generate information graph

- There should be a function that maps each instance to a graph $f(i)$, $\mathbb{R}^n \rightarrow G(N,E)$
- Node for attribute $a_i$ $N(a_i) = 1$ (exists) iff $a_i$ occurs in the instance or present in detector
- Edge weight $E_{ij}$ is defined:

$$E_{ij} = \begin{cases} 
  f(i, j) & \text{if } x > \delta \text{ and } i \neq j \\
  0 & \text{otherwise} 
\end{cases}$$

where $i, j$ are features

$$f(i, j) = \frac{L_k \times \log g_k(i)}{\sum_p L_p \times \log(g_p(j))} \times \frac{L_k \times \log g_k(j)}{\sum_p L_p \times \log(g_p(j))}$$

where $k$ denotes the instance considered, $L_k$ is the label of the instance, $g_k(i)$ is the feature function on feature $i$ in instance $k$.

And feature function $g$: ($N_i$ is a normalizing factor on word $i$)

$$g_i = \begin{cases} 
  f_i / 0.5 & \text{if feature } i \text{ is visual} \\
  f_i / N_i & \text{otherwise} 
\end{cases}$$
Idea 2: Mining Frequent Subgraphs

• Given a graph representation of the dataset: **find frequent subgraph structures that most commonly occur among all graph instances**

• Frequent subgraph:
  – Shows relationship between features of different modality and corresponding strength
  – Explicit association
Idea 2-Mining Frequent Subgraphs

• Why not other alternatives-
  – Graph partition on a single aggregate graph: we are looking for important+diverse+overlapping sets of attributes (but graph partition looks for diverse+non-overlapping sets of attributes)
  – Sparsity-enforced factorization (Sparse PCA): PCA focuses more on variance maximization in subspace construction (but we are looking for frequent attributes, and don’t necessarily need to reconstruct the vector space)
Idea 2: Mining Frequent Subgraphs

Graph Database → Frequent Patterns → Graph Patterns

F(g) = support function

gSpan[5]: Conceptually, build a DFS search tree for candidate substructures to examine
Idea 2: Mining Frequent Subgraph

- Alternatively, we can increase support by first clustering text/visual attributes

- Then each cluster is mapped to a new node and do frequent subgraph mining afterward
Idea 3: Selecting Representative Subgraphs

• **Input:** a set of attributes
• **Output:** most relevant attributes

Boosted Attribute Selection

- $F = \{\text{empty}\}$, output attribute set
- $\delta = \text{model improvement threshold}$

Sort the set $S$ of candidate attributes in canonical order

Let additive model $M = \text{null}$, and the model is defined by attributes and corresponding weights

For each attribute $A$ in $S$:

- Train weak model using only $A$
- Learn $w_A = \text{boosted weight of } A \text{ given current } M$

$$M_{\text{new}} = M_{\text{current}} \cup \{A, w_A\}$$

if accuracy($M_{\text{new}}(D)$) – accuracy($M_{\text{current}}(D)$) > $\delta$:

$$F = F \cup \{A\}$$

return $F$
Sub-goal: Empirical Evaluation of Detectors’ Effectiveness

• Evaluation 1: **Qualitative expression of text-visual detectors given specific domain**

• Evaluation 2: **Efficiency of detector generation**

• Evaluation 3: **Using detectors as compact features for sentiment analysis**
Evaluation 1

• Select certain domains and print text-visual detectors and top returning instances:

• Goal: the results should be visually interpretable
Evaluation 2

• Use the derived set of detectors as features to solve the task of **sentiment analysis**
• Evaluation runs: K-fold CV on each data
• Problem formulation: Bi/Ter-nary sentiment classification (pos/neg/neutral)
• Classification Model: SVM
Transforming attribute to feature vector

• Each instance = $G_i(V_i, E_i)$
• Each detector = $G_d(V_d, E_d)$
• Feature vector for instance i and detector d:
  $F(i,d) = (V_{d1} * V^i_{d1}, ..., V_{dn} * V^i_{dn}, ..., E_{d1} * E^i_{d1}, ..., E_{dk} * E^i_{dk})$

  – $V_{d1}$-$V_{dn}$ are matching vertices from 1~n for subgraph detector d
  – $E_{d1}$-$E_{dk}$ are matching edges from 1~k for subgraph detector d
Evaluation 2

• Comparing methods:
  – Pure BoW features:
    • Textual attributes
    • Visual attributes
  – Early fusion of BoW features
    • Append text/visual attributes to a feature vector
    • (Multiple kernel learning)
  – Late fusion of BoW features-average
  – Proposed: append multiple detectors into single feature vector
    • Append detectors without weighting
    • Append detectors with weighting
Evaluation 2

• Comparing number of detectors used and corresponding performance
  – Number of detectors vs prediction accuracy

• Goal: There should be a reduction of total features while providing reasonable results
  – Denoising
  – Evaluating relevance of attribute
  – Improvement is needed here
Evaluation 3

• Run time evaluation: with original visual attributes, we need to check every detector for every instance
  – Check: runtime at testing phase using all detectors
  – Check: runtime using extracted text-visual detectors with varying number of detectors

• Goal: factors of speedup using partial text-visual detectors
Evaluation 3

• To show that it is necessary to provide a separate set of detectors for each domain

• For each domain of twitter-image data, plot the heatmap of constantly present categories
(Evaluation 4)

• Run **object category recognition** task on a selected subset of Twitter-based image tweets

• Comparing methods
  
  – Baselines (method-wise):
    • GIST features for SVM
    • MKL
  
  – Baseline (detector-wise): SVM (Classmes)
  
  – State-of-art: LP-β

  – **Proposed:**
    • SVM on bimodal detectors
    • Boosted model on bimodal detectors (graph boost)
(Evaluation 4)

• Comparing criteria:
  – Feature size
  – Training/testing runtime
  – Classification accuracy
  – Detector size selection vs performance
Sub-goal: Application

• Application 1: Added function to Rongrong’s app, where we can query by domain and return sentiment-ranked tweets and attributes

- Rock, !, : ), Cool, Bright, Exciting, Hot
- Monitor, ink, photo, news
- Freezing, Depressing, Bore, Hungry
Sub-goal: Application

- Application 2 (if time permits): In addition to domain query, we also include time granularity to see if there are differences in discovered detectors under drifting sentiment concepts.

Politics

2011-2012
- Obamacare
- Tax
- Recession
- Greece

2012-
- President
- Election
- Romney
- Obama
Timetable for Executables

- 8/14-8/18: Data labeling, idea implementation, manual verification via evaluation 1, preliminary checking on evaluation 2/3
- 8/18-8/24: Detailed experiments on evaluation 2 and 3, examining different parameters and contrast performance under different domains
- 8/24-9/6: @Kyoto, further consolidation on evaluation 1, 2, 3
- September: Application, drafting ideas
- Deadline?