Chapter 11: Opinion Mining

Bing Liu
Department of Computer Science
University of Illinois at Chicago
liub@cs.uic.edu

Introduction – facts and opinions

- Two main types of textual information on the Web.
  - Facts and Opinions
- Current search engines search for facts (assume they are true)
  - Facts can be expressed with topic keywords.
- Search engines do not search for opinions
  - Opinions are hard to express with a few keywords
    - How do people think of Motorola Cell phones?
  - Current search ranking strategy is not appropriate for opinion retrieval/search.

Introduction – user generated content

- Word-of-mouth on the Web
  - One can express personal experiences and opinions on almost anything, at review sites, forums, discussion groups, blogs ... (called the user generated content.)
  - They contain valuable information
  - Web/global scale: No longer – one's circle of friends
- Our interest: to mine opinions expressed in the user-generated content
  - An intellectually very challenging problem.
  - Practically very useful.

Introduction – Applications

- Businesses and organizations: product and service benchmarking. Market intelligence.
  - Business spends a huge amount of money to find consumer sentiments and opinions.
    - Consultants, surveys and focused groups, etc
- Individuals: interested in other's opinions when
  - Purchasing a product or using a service,
  - Finding opinions on political topics,
- Ads placements: Placing ads in the user-generated content
  - Place an ad when one praises a product.
  - Place an ad from a competitor if one criticizes a product.
- Opinion retrieval/search: providing general search for opinions.

Two types of evaluation

- Direct Opinions: sentiment expressions on some objects, e.g., products, events, topics, persons.
  - E.g., “the picture quality of this camera is great”
  - Subjective
- Comparisons: relations expressing similarities or differences of more than one object. Usually expressing an ordering.
  - E.g., “car x is cheaper than car y.”
  - Objective or subjective.

Opinion search (Liu, Web Data Mining book, 2007)

- Can you search for opinions as conveniently as general Web search?
- Whenever you need to make a decision, you may want some opinions from others,
  - Wouldn’t it be nice? you can find them on a search system instantly, by issuing queries such as
    - Opinions: “Motorola cell phones”
    - Comparisons: “Motorola vs. Nokia”
- Cannot be done yet! (but could be soon …)

Typical opinion search queries

- Find the opinion of a person or organization (opinion holder) on a particular object or a feature of the object.
  - E.g., what is Bill Clinton's opinion on abortion?
- Find positive and/or negative opinions on a particular object (or some features of the object), e.g.,
  - customer opinions on a digital camera.
  - public opinions on a political topic.
- Find how opinions on an object change over time.
- How object A compares with Object B?
  - Gmail vs. Hotmail

Find the opinion of a person on X

- In some cases, the general search engine can handle it, i.e., using suitable keywords.
  - Bill Clinton's opinion on abortion
- Reason:
  - One person or organization usually has only one opinion on a particular topic.
  - The opinion is likely contained in a single document.
  - Thus, a good keyword query may be sufficient.

Find opinions on an object

We use product reviews as an example:

- Searching for opinions in product reviews is different from general Web search.
  - E.g., search for opinions on "Motorola RAZR V3"
- General Web search (for a fact): rank pages according to some authority and relevance scores.
  - The user views the first page (if the search is perfect).
  - **One fact = Multiple facts**
- Opinion search: rank is desirable, however
  - reading only the review ranked at the top is not appropriate because it is only the opinion of one person.
  - **One opinion ≠ Multiple opinions**

Search opinions (contd)

- Ranking:
  - produce two rankings
    - Positive opinions and negative opinions
    - Some kind of summary of both, e.g., # of each
  - Or, one ranking but
    - The top (say 30) reviews should reflect the natural distribution of all reviews (assume that there is no spam), i.e., with the right balance of positive and negative reviews.
- Questions:
  - Should the user reads all the top reviews? OR
  - Should the system prepare a summary of the reviews?

Reviews are similar to surveys

- Reviews can be regarded as traditional surveys.
  - In traditional survey, returned survey forms are treated as raw data.
  - Analysis is performed to summarize the survey results.
    - E.g., % against or for a particular issue, etc.
- In opinion search,
  - Can a summary be produced?
  - What should the summary be?

Roadmap

- **Opinion mining – the abstraction**
- Document level sentiment classification
- Sentence level sentiment analysis
- Feature-based opinion mining and summarization
- Comparative sentence and relation extraction
- Summary

Opinion mining – the **abstraction**

(Hu and Liu, KDD-04; Liu, Web Data Mining book 2007)

- Basic components of an opinion
  - Opinion holder: The person or organization that holds a specific opinion on a particular object.
  - Object: on which an opinion is expressed
  - Opinion: a view, attitude, or appraisal on an object from an opinion holder.
- Objectives of opinion mining: many ...
- **Let us abstract the problem**
  - put existing research into a common framework
- We use consumer reviews of products to develop the ideas. Other opinionated contexts are similar.

Object/entity

- **Definition** (**object**): An object *O* is an entity which can be a product, person, event, organization, or topic. *O* is represented as
  - a hierarchy of components, sub-components, and so on.
  - Each node represents a component and is associated with a set of **attributes** of the component.
  - *O* is the root node (which also has a set of attributes)
- An opinion can be expressed on any node or attribute of the node.
- To simplify our discussion, we use "**features**" to represent both components and attributes.
  - The term "feature" should be understood in a ***broad sense***,
    - Product feature, topic or sub-topic, event or sub-event, etc
- Note: the object *O* itself is also a feature.

Model of a review

- An object O is represented with a finite set of features, $F = \{f_1, f_2, \ldots, f_n\}$.
  - Each feature f_i in F can be expressed with a finite set of words or phrases W_i, which are **synonyms**.

  That is to say: we have a set of corresponding synonym sets $W = \{W_1, W_2, \ldots, W_n\}$ for the features.
- **Model of a review**: An opinion holder j comments on a subset of the features $S_j \subseteq F$ of object O.
  - For each feature $f_k \in S_j$ that j comments on, he/she
    - chooses a word or phrase from W_k to describe the feature, and
    - expresses a positive, negative or neutral opinion on f_k.

Opinion mining tasks

- At the document (or review) level:

  Task: sentiment classification of reviews
    - Classes: positive, negative, and neutral
    - Assumption: each document (or review) focuses on a single object (not true in many discussion posts) and contains opinion from a single opinion holder.
- At the sentence level:

  Task 1: identifying subjective/opinionated sentences
    - Classes: objective and subjective (opinionated)

  Task 2: sentiment classification of sentences
    - Classes: positive, negative and neutral.
    - Assumption: a sentence contains only one opinion
      - not true in many cases.
    - Then we can also consider clauses or phrases.

Opinion mining tasks (contd)

- At the feature level:

  Task 1: Identify and extract object features that have been commented on by an opinion holder (e.g., a reviewer).

  Task 2: Determine whether the opinions on the features are positive, negative or neutral.

  Task 3: Group feature synonyms.

  - Produce a feature-based opinion summary of multiple reviews (more on this later).
- Opinion holders: identify holders is also useful, e.g., in news articles, etc, but they are usually known in the user generated content, i.e., authors of the posts.

More at the feature level

- **Problem 1**: Both *F* and *W* are unknown.
  - We need to perform all three tasks:
- **Problem 2**: *F* is known but *W* is unknown.
  - All three tasks are still needed. Task 3 is easier. It becomes the problem of matching the discovered features with the set of given features *F*.
- **Problem 3**: *W* is known (*F* is known too).
  - Only task 2 is needed.

F: the set of features

W: synonyms of each feature

Roadmap

- Opinion mining – the abstraction
- **Document level sentiment classification**
- Sentence level sentiment analysis
- Feature-based opinion mining and summarization
- Comparative sentence and relation extraction
- Summary

Sentiment classification

- Classify documents (e.g., reviews) based on the overall sentiments expressed by opinion holders (authors),
  - Positive, negative, and (possibly) neutral
  - Since in our model an object *O* itself is also a feature, then sentiment classification essentially determines the opinion expressed on *O* in each document (e.g., review).
- Similar but different from topic-based text classification.
  - In topic-based text classification, topic words are important.
  - In sentiment classification, sentiment words are more important, e.g., great, excellent, horrible, bad, worst, etc.

Unsupervised review classification
(Turney, ACL-02)

- Data: reviews from epinions.com on automobiles, banks, movies, and travel destinations.
- The approach: Three steps
- Step 1:
  - Part-of-speech tagging
  - Extracting two consecutive words (two-word phrases) from reviews if their tags conform to some given patterns, e.g., (1) JJ, (2) NN.

- Step 2: Estimate the semantic orientation (SO) of the extracted phrases
  - Use Pointwise mutual information

$$PMI(word_1, word_2) = \log_2\left(\frac{P(word_1 \wedge word_2)}{P(word_1)P(word_2)}\right)$$

  - Semantic orientation (SO):

SO(phrase) = PMI(phrase, “excellent”)
- PMI(phrase, “poor”)

  - Using AltaVista near operator to do search to find the number of hits to compute PMI and SO.

- Step 3: Compute the average SO of all phrases
  - classify the review as recommended if average SO is positive, not recommended otherwise.
- Final classification accuracy:
  - automobiles - 84%
  - banks - 80%
  - movies - 65.83
  - travel destinations - 70.53%

Sentiment classification using machine learning methods (Pang et al, EMNLP-02)

- This paper directly applied several machine learning techniques to classify movie reviews into positive and negative.
- Three classification techniques were tried:
  - Naïve Bayes
  - Maximum entropy
  - Support vector machine
- Pre-processing settings: negation tag, unigram (single words), bigram, POS tag, position.
- SVM: the best accuracy 83% (unigram)

Review classification by scoring features

(Dave, Lawrence and Pennock, WWW-03)

- It first selects a set of features $F = f_1, f_2, \ldots\ldots$
  - Note: machine learning features, but product features.
- Score the features
  - C and C' are classes

$$score(f_i) = \frac{P(f_i \mid C) - P(f_i \mid C')}{P(f_i \mid C) + P(f_i \mid C')}$$

- Classification of a review d_j (using sign):

$$class(d_j) = \begin{cases} C & eval(d_j) > 0 \\ C' & eval(d_j) < 0 \end{cases}$$

$$eval(d_j) = \sum_i score(f_i)$$

- Accuracy of 84-88%.

Other related works

- Using PMI, syntactic relations and other attributes with SVM (Mullen and Collier, EMNLP-04).
- Sentiment classification considering rating scales (Pang and Lee, ACL-05).
- Comparing supervised and unsupervised methods (Chaovalit and Zhou, HICSS-05)
- Using semi-supervised learning (Goldberg and Zhu, Workshop on TextGraphs, at HLT-NAAL-06).
- Review identification and sentiment classification of reviews (Ng, Dasgupta and Arifin, ACL-06).
- Sentiment classification on customer feedback data (Gamon, Coling-04).
- Comparative experiments (Cui et al. AAAI-06)

Roadmap

- Opinion mining – the abstraction
- Document level sentiment classification
- **Sentence level sentiment analysis**
- Feature-based opinion mining and summarization
- Comparative sentence and relation extraction
- Summary

Sentence-level sentiment analysis

- Document-level sentiment classification is too coarse for most applications.
- Let us move to the sentence level.
- Much of the work on sentence level sentiment analysis focuses on identifying subjective sentences in news articles.
  - Classification: objective and subjective.
  - All techniques use some forms of machine learning.
  - E.g., using a naïve Bayesian classifier with a set of data features/attributes extracted from training sentences (Wiebe et al. ACL-99).

Using learnt patterns (Rilloff and Wiebe, EMNLP-03)

- A bootstrapping approach.
  - A high precision classifier is first used to automatically identify some subjective and objective sentences.
    - Two high precision (but low recall) classifiers are used,
      - a high precision subjective classifier
      - A high precision objective classifier
      - Based on manually collected lexical items, single words and n-grams, which are good subjective clues.
  - A set of patterns are then learned from these identified subjective and objective sentences.
    - Syntactic templates are provided to restrict the kinds of patterns to be discovered, e.g., <subj> passive-verb.
  - The learned patterns are then used to extract more subject and objective sentences (the process can be repeated).

Subjectivity and polarity (orientation) (Yu and Hazivassiloglou, EMNLP-03)

- For subjective or opinion sentence identification, three methods are tried:
  - Sentence similarity.
  - Naïve Bayesian classification.
  - Multiple naïve Bayesian (NB) classifiers.
- For opinion orientation (positive, negative or neutral) (also called polarity) classification, it uses a similar method to (Turney, ACL-02), but
  - with more seed words (rather than two) and based on log-likelihood ratio (LLR).
  - For classification of each word, it takes the average of LLR scores of words in the sentence and use cutoffs to decide positive, negative or neutral.

Other related work

- Consider gradable adjectives (Hatzivassiloglou and Wiebe, Coling-00)
- Semi-supervised learning with the initial training set identified by some strong patterns and then applying NB or self-training (Wiebe and Riloff, CICLing-05).
- Finding strength of opinions at the clause level (Wilson et al. AAAI-04).
- Sum up orientations of opinion words in a sentence (or within some word window) (Kim and Hovy, COLING-04).

Let us go further?

- Sentiment classification at both document and sentence (or clause) levels are useful, but
  - They do not find what the opinion holder liked and disliked.
- An negative sentiment on an object
  - does not mean that the opinion holder dislikes everything about the object.
- A positive sentiment on an object
  - does not mean that the opinion holder likes everything about the object.
- **We need to go to the feature level.**

But before we go further

- Let us discuss **Opinion Words or Phrases** (also called polar words, opinion bearing words, etc). E.g.,
  - Positive: beautiful, wonderful, good, amazing,
  - Negative: bad, poor, terrible, cost someone an arm and a leg (idiom).
- They are instrumental for opinion mining (obviously)
- Three main ways to compile such a list:
  - Manual approach: not a bad idea, only an one-time effort
  - Corpus-based approaches
  - Dictionary-based approaches
- **Important to note:**
  - **Some opinion words are context independent (e.g., good).**
  - **Some are context dependent (e.g., long).**

Corpus-based approaches

- Rely on syntactic or co-occurrence patterns in large corpora. (Hazivassiloglou and McKeown, ACL-97; Turney, ACL-02; Yu and Hazivassiloglou, EMNLP-03; Kanayama and Nasukawa, EMNLP-06; Ding and Liu SIGIR-07)
  - Can find domain (not context!) dependent orientations (positive, negative, or neutral).
- (Turney, ACL-02) and (Yu and Hazivassiloglou, EMNLP-03) are similar.
  - Assign opinion orientations (polarities) to words/phrases.
  - (Yu and Hazivassiloglou, EMNLP-03) is different from (Turney, ACL-02)
    - use more seed words (rather than two) and use log-likelihood ratio (rather than PMI).

Corpus-based approaches (contd)

- Use constraints (or conventions) on connectives to identify opinion words (Hazivassiloglou and McKeown, ACL-97; Kanayama and Nasukawa, EMNLP-06; Ding and Liu, 2007). E.g.,
- Conjunction: conjoined adjectives usually have the same orientation (Hazivassiloglou and McKeown, ACL-97).
    - E.g., “This car is *beautiful* **and** *spacious*.” (conjunction)
  - AND, OR, BUT, EITHER-OR, and NEITHER-NOR have similar constraints.
  - Learning using
    - log-linear model: determine if two conjoined adjectives are of the same or different orientations.
    - Clustering: produce two sets of words: positive and negative
  - Corpus: 21 million word 1987 Wall Street Journal corpus.

Corpus-based approaches (contd)

- (Kanayama and Nasukawa, EMNLP-06) takes a similar approach to (Hazivassiloglou and McKeown, ACL-97) but for Japanese words:
  - Instead of using learning, it uses two criteria to determine whether to add a word to positive or negative lexicon.
  - Have an initial seed lexicon of positive and negative words.
- (Ding and Liu, 2007) also exploits constraints on connectives, but with two differences
  - It uses them to assign opinion orientations to product features (more on this later).
    - One word may indicate different opinions in the same domain.
      - “The battery life is *long*” (+) and “It takes a *long* time to focus” (-).
    - **Find domain opinion words is insufficient.**
  - It can be used without a large corpus.

Dictionary-based approaches

- Typically use WordNet's synsets and hierarchies to acquire opinion words
  - Start with a small seed set of opinion words.
  - Use the set to search for synonyms and antonyms in WordNet (Hu and Liu, KDD-04; Kim and Hovy, COLING-04).
  - Manual inspection may be used afterward.
- Use additional information (e.g., glosses) from WordNet (Andreevskaia and Bergler, EACL-06) and learning (Esuti and Sebastiani, CIKM-05).
- Weakness of the approach: Do not find context dependent opinion words, e.g., small, long, fast.

Roadmap

- Opinion mining – the abstraction
- Document level sentiment classification
- Sentence level sentiment analysis
- **Feature-based opinion mining and summarization**
- Comparative sentence and relation extraction
- Summary

Feature-based opinion mining and summarization (Hu and Liu, KDD-04)

- Again focus on reviews (easier to work in a concrete domain!)
- Objective: find what reviewers (opinion holders) liked and disliked
  - Product features and opinions on the features
- Since the number of reviews on an object can be large, an opinion summary should be produced.
  - Desirable to be a structured summary.
  - Easy to visualize and to compare.
  - Analogous to but different from multi-document summarization.

The tasks

- Recall the three tasks in our model.

  Task 1: Extract object features that have been commented on in each review.

  Task 2: Determine whether the opinions on the features are positive, negative or neutral.

  Task 3: Group feature synonyms.

  - Produce a summary
- Task 2 may not be needed depending on the format of reviews.

Different review format

Format 1 - Pros, Cons and detailed review: The reviewer is asked to describe Pros and Cons separately and also write a detailed review. Epinions.com uses this format.

Format 2 - Pros and Cons: The reviewer is asked to describe Pros and Cons separately. Cnet.com used to use this format.

Format 3 - free format: The reviewer can write freely, i.e., no separation of Pros and Cons. Amazon.com uses this format.

Format 1

My SLR is on the shelf

by camerafun4. Aug 09 '04

Pros: Great photos, easy to use, very small
Cons: Battery usage; included memory is stingy.

I had never used a digital camera prior to purchasing
have always used a SLR ... **Read the full review**

Format 3

GREAT Camera., Jun 3, 2004

Reviewer: **jprice174** from Atlanta, Ga.

I did a lot of research last year before I bought this camera... It kinda hurt to leave behind my beloved nikon 35mm SLR, but I was going to Italy, and I needed something smaller, and digital.

The pictures coming out of this camera are amazing. The 'auto' feature takes great pictures most of the time. And with digital, you're not wasting film if the picture doesn't come out.

Format 2

User rating
Perfect
10
out of 10

"It is a great digitbal still camera for this century"
September 1, 2004

Pros:
It's small in size, and the rotatable lens is great. It's very easy to use, and has fast response from the shutter. The LCD has increased from 1.5 in to 1.8, which gives bigger view. It has lots of modes to choose from in order to take better pictures.

Cons:
It almost has no cons, it would be better if the LCD is bigger and it's going to be best if the model is designed to a smaller size.

Feature-based opinion summary (Hu and Liu, KDD-04)

GREAT Camera., Jun 3, 2004
Reviewer: **jprice174** from Atlanta, Ga.

I did a lot of research last year before I bought this camera... It kinda hurt to leave behind my beloved nikon 35mm SLR, but I was going to Italy, and I needed something smaller, and digital.

The pictures coming out of this camera are amazing. The 'auto' feature takes great pictures most of the time. And with digital, you're not wasting film if the picture doesn't come out. …

....

Feature Based Summary:

Feature1: **picture**
Positive: 12

- The pictures coming out of this camera are amazing.
- Overall this is a good camera with a really good picture clarity.

…

Negative: 2

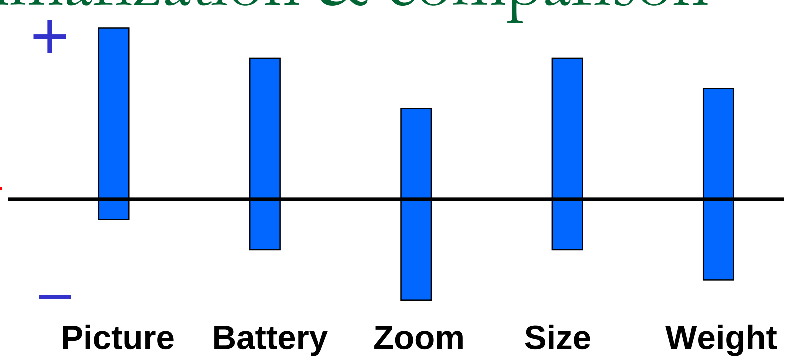
- The pictures come out hazy if your hands shake even for a moment during the entire process of taking a picture.
- Focusing on a display rack about 20 feet away in a brightly lit room during day time, pictures produced by this camera were blurry and in a shade of orange.

Feature2: **battery life**
…

Visual summarization & comparison

- Summary of reviews of

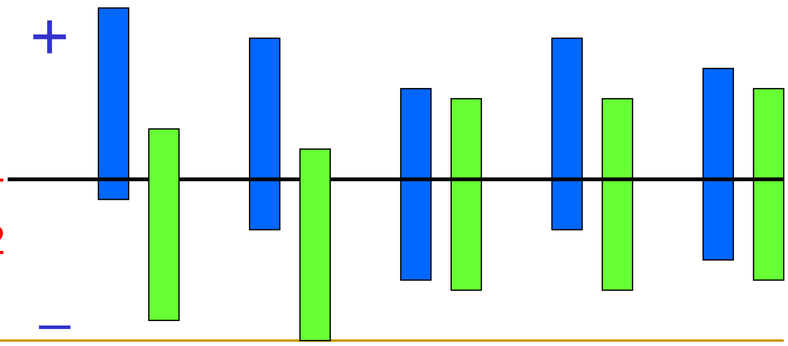
Digital camera 1



- Comparison of reviews of

Digital camera 1

Digital camera 2



Feature extraction from Pros and Cons of Format 1 (Liu et al WWW-03; Hu and Liu, AAAI-CAAW-05)

- **Observation**: Each sentence segment in Pros or Cons contains only one feature. Sentence segments can be separated by commas, periods, semi-colons, hyphens, '&''s, 'and''s, 'but''s, etc.

- Pros in Example 1 can be separated into 3 segments:

great photos	<photo>
easy to use	<use>
very small	<small> ⇒ <size>

- Cons can be separated into 2 segments:

battery usage	<battery>
included memory is stingy	<memory>

Extraction using label sequential rules

- Label sequential rules (LSR) are a special kind of sequential patterns, discovered from sequences.
- LSR Mining is supervised (Liu's Web mining book 2006).
- The training data set is a set of sequences, e.g.,

  "*Included memory is stingy*"

  is turned into a sequence with POS tags.

  ⟨{included, VB}{memory, NN}{is, VB}{stingy, JJ}⟩

  then turned into

  ⟨{included, VB}{$feature, NN}{is, VB}{stingy, JJ}⟩

Using LSRs for extraction

- Based on a set of training sequences, we can mine label sequential rules, e.g.,

  ⟨{easy, JJ }{to}{*, VB}⟩ → ⟨{easy, JJ}{to}{$feature, VB}⟩
  [sup = 10%, conf = 95%]

Feature Extraction

- Only the right hand side of each rule is needed.
- The word in the sentence segment of a new review that matches $feature is extracted.
- We need to deal with conflict resolution also (multiple rules are applicable.

Extraction of features of formats 2 and 3

- Reviews of these formats are usually complete sentences

  e.g., "the pictures are very clear."
  - Explicit feature: picture
- "It is small enough to fit easily in a coat pocket or purse."
  - Implicit feature: size
- Extraction: Frequency based approach
  - Frequent features
  - Infrequent features

Frequency based approach

(Hu and Liu, KDD-04; Liu, Web Data Mining book 2007)

- Frequent features: those features that have been talked about by many reviewers.
- Use sequential pattern mining
- Why the frequency based approach?
  - Different reviewers tell different stories (irrelevant)
  - When product features are discussed, the words that they use converge.
  - They are main features.
- Sequential pattern mining finds frequent phrases.
- Froogle has an implementation of the approach (no POS restriction).

Using part-of relationship and the Web

(Popescu and Etzioni, EMNLP-05)

- Improved (Hu and Liu, KDD-04) by removing those frequent noun phrases that may not be features: better precision (a small drop in recall).
- It identifies part-of relationship
  - Each noun phrase is given a pointwise mutual information score between the phrase and **part discriminators** associated with the product class, e.g., a scanner class.
  - The part discriminators for the scanner class are, "of scanner", "scanner has", "scanner comes with", etc, which are used to find components or parts of scanners by searching on the Web: the KnowItAll approach, (Etzioni et al, WWW-04).

Infrequent features extraction

- How to find the infrequent features?
- Observation: the same opinion word can be used to describe different features and objects.
  - "The pictures are absolutely amazing."
  - "The software that comes with it is amazing."

Identify feature synonyms

- Liu et al (WWW-05) made an attempt using only WordNet.
- Carenini et al (K-CAP-05) proposed a more sophisticated method based on several similarity metrics, but it requires a taxonomy of features to be given.
  - The system merges each discovered feature to a feature node in the taxonomy.
  - The similarity metrics are defined based on string similarity, synonyms and other distances measured using WordNet.
  - Experimental results based on digital camera and DVD reviews show promising results.
- Many ideas in information integration are applicable.

Identify opinion orientation on feature

- For each feature, we identify the sentiment or opinion orientation expressed by a reviewer.
- We work based on sentences, but also consider,
  - A sentence can contain multiple features.
  - Different features may have different opinions.
  - E.g., The battery life and picture quality are *great* (+), but the view founder is *small* (-).
- Almost all approaches make use of opinion words and phrases. But notice again:
  - Some opinion words have context independent orientations, e.g., "great".
  - Some other opinion words have context dependent orientations, e.g., "small"
- Many ways to use them.

Aggregation of opinion words

(Hu and Liu, KDD-04; Ding and Liu, 2007)

- Input: a pair (f, s), where f is a product feature and s is a sentence that contains f.
- Output: whether the opinion on f in s is positive, negative, or neutral.
- Two steps:
  - Step 1: split the sentence if needed based on BUT words (but, except that, etc).
  - Step 2: work on the segment s_f containing f. Let the set of opinion words in s_f be w_1, .., w_n. Sum up their orientations (1, -1, 0), and assign the orientation to (f, s) accordingly.
- In (Ding and Liu, SIGIR-07), step 2 is changed to $\sum_{i=1}^{n} \frac{w_i.o}{d(w_i, f)}$ with better results. $w_i.o$ is the opinion orientation of w_i. $d(w_i, f)$ is the distance from f to w_i.

Context dependent opinions

- Popescu and Etzioni (EMNLP-05) used
  - constraints of connectives in (Hazivassiloglou and McKeown, ACL-97), and some additional constraints, e.g., morphological relationships, synonymy and antonymy, and
  - relaxation labeling to propagate opinion orientations to words and features.
- Ding and Liu (2007) used
  - constraints of connectives both at intra-sentence and inter-sentence levels, and
  - additional constraints of, e.g., TOO, BUT, NEGATION, ….

  to directly assign opinions to (*f*, *s*) with good results (> 0.85 of F-score).

Some other related work

- Morinaga et al. (KDD-02).
- Yi et al. (ICDM-03)
- Kobayashi et al. (AAAI-CAAW-05)
- Ku et al. (AAAI-CAAW-05)
- Carenini et al (EACL-06)
- Kim and Hovy (ACL-06a)
- Kim and Hovy (ACL-06b)
- Eguchi and Lavrendo (EMNLP-06)
- Zhuang et al (CIKM-06)
- Many more

Roadmap

- Opinion mining – the abstraction
- Document level sentiment classification
- Sentence level sentiment analysis
- Feature-based opinion mining and summarization
- **Comparative sentence and relation extraction**
- Summary

Extraction of Comparatives

(Jinal and Liu, SIGIR-06, AAAI-06; Liu's Web Data Mining book)

- Recall: Two types of evaluation
  - Direct opinions: "This car is bad"
  - Comparisons: "Car X is not as good as car Y"
- They use different language constructs.
- Direct expression of sentiments are good. Comparison may be better.
  - Good or bad, compared to what?
- Comparative Sentence Mining
  - Identify comparative sentences, and
  - extract comparative relations from them.

Linguistic Perspective

- Comparative sentences use morphemes like
  - *more/most*, *-er/-est*, *less/least* and *as*.
- *than* and *as* are used to make a 'standard' against which an entity is compared.

Limitations

- Limited coverage
  - *Ex: "In market capital, Intel is way ahead of Amd*"
- Non-comparatives with comparative words
  - Ex1: "*In the context of speed, faster means better*"
- For human consumption; no computational methods

Types of Comparatives: Gradable

- *Gradable*
  - *Non-Equal Gradable*: Relations of the type *greater* or *less than*
    - *Keywords like better, ahead, beats, etc*
    - *Ex: "optics of camera A is better than that of camera B"*
  - *Equative*: Relations of the type *equal to*
    - Keywords and phrases like *equal to*, *same as*, *both*, *all*
    - Ex: "*camera A and camera B both come in 7MP*"
  - *Superlative*: Relations of the type *greater* or *less than all others*
    - Keywords and phrases like *best, most, better than all*
    - Ex: "*camera A is the cheapest camera available in market*"

Types of comparatives: non-gradable

- Non-Gradable: Sentences that compare features of two or more objects, but do not grade them. Sentences which imply:
  - Object A is similar to or different from Object B with regard to some features.
  - Object A has feature F_1, Object B has feature F_2 (F_1 and F_2 are usually substitutable).
  - Object A has feature F, but object B does not have.

Comparative Relation: gradable

- **Definition**: A gradable comparative relation captures the essence of a gradable comparative sentence and is represented with the following:

  (relationWord, features, entityS1, entityS2, type)

  - *relationWord*: The keyword used to express a comparative relation in a sentence.
  - *features*: a set of features being compared.
  - *entityS1* and *entityS2*: Sets of entities being compared.
  - *type*: *non-equal gradable*, *equative* or *superlative*.

Examples: Comparative relations

- Ex1: "*car X has better controls than car Y*"
  (relationWord = better, features = controls, entityS1 = car X, entityS2 = car Y, type = non-equal-gradable)
- Ex2: "*car X and car Y have equal mileage*"
  (relationWord = equal, features = mileage, entityS1 = car X, entityS2 = car Y, type = equative)
- Ex3: "*Car X is cheaper than both car Y and car Z*"
  (relationWord = cheaper, features = null, entityS1 = car X, entityS2 = {car Y, car Z}, type = non-equal-gradable)
- Ex4: "company X produces a variety of cars, but still best cars come from company Y"
  (relationWord = best, features = cars, **entityS1** = company Y, entityS2 = null, type = superlative)

Tasks

Given a collection of evaluative texts

Task 1: Identify comparative sentences.

Task 2: Categorize different types of comparative sentences.

Task 2: Extract comparative relations from the sentences.

Identify comparative sentences

(Jinal and Liu, SIGIR-06)

Keyword strategy

- An observation: It is easy to find a small set of keywords that covers almost all comparative sentences, i.e., with a very high recall and a reasonable precision
- We have compiled a list of **83 keywords** used in comparative sentences, which includes:
  - Words with POS tags of JJR, JJS, RBR, RBS
    - POS tags are used as keyword instead of individual words.
    - Exceptions: more, less, most and least
  - Other indicative words like *beat*, *exceed*, ahead, etc
  - Phrases like *in the lead*, *on par with*, etc

2-step learning strategy

- **Step1**: Extract sentences which contain at least a keyword (recall = 98%, precision = 32% on our data set for gradables)
- **Step2**: Use the naïve Bayes (NB) classifier to classify sentences into two classes
  - comparative and non-comparative.
  - Attributes: **class sequential rules** (CSRs) generated from sentences in step1, e.g.,

    $\langle\{1\}\{3\}\{7, 8\}\rangle \rightarrow class_i$ [sup = 2/5, conf = 3/4]

1. Sequence data preparation

- Use words within radius *r* of a keyword to form a sequence (words are replaced with POS tags)
- ….

2. CSR generation

- Use different minimum supports for different keywords (multiple minimum supports)
- 13 manual rules, which were hard to generate automatically.

3. Learning using a NB classifier

- Use CSRs and manual rules as attributes to build a final classifier.

Classify different types of comparatives

- Classify comparative sentences into three types: non-equal gradable, equative, and superlative
  - SVM learner gave the best result.
  - Attribute set is the set of keywords.
  - If the sentence has a particular keyword in the attribute set, the corresponding value is 1, and 0 otherwise.

Extraction of comparative relations

(Jindal and Liu, AAAI-06; Liu's Web mining book 2006)

Assumptions

- There is only one relation in a sentence.
- Entities and features are nouns (includes nouns, plural nouns and proper nouns) and pronouns.
  - Adjectival comparatives
  - Does not deal with adverbial comparatives

3 steps

- Sequence data generation
- Label sequential rule (LSR) generation
- Build a sequential cover/extractor from LSRs

Sequence data generation

- Label Set = {$entityS1, $entityS2, $feature}
- Three labels are used as pivots to generate sequences.
  - Radius of 4 for optimal results
- Following words are also added
  - Distance words = {*l*1, *l*2, *l*3, *l*4, *r*1, *r*2, *r*3, *r*4}, where "*li*" means distance of *i* to the left of the pivot.
    "*ri*" means the distance of *i* to the right of pivot.
  - Special words *#start* and *#end* are used to mark the start and the end of a sentence.

Sequence data generation example

The comparative sentence

"*Canon/NNP has/VBZ better/JJR optics/NNS*" has *$entityS1* "*Canon*" and *$feature* "*optics*".

Sequences are:

- ⟨{#start}{l1}{**$entityS1,** NNP}{*r*1}{has, VBZ }{r2 } {better, JJR}{*r*3}{$Feature, NNS}{r4}{#end}⟩

- ⟨{#start}{l4}{**$entityS1**, NNP}{l3}{has, VBZ}{*l*2} {better, JJR}{*l*1}{**$Feature**, NNS}{*r*1}{#end}⟩

Build a sequential cover from LSRs

LSR: ⟨{*, NN}{VBZ}⟩ → ⟨{$entityS1, NN}{VBZ}⟩

- Select the LSR rule with the highest confidence. Replace the matched elements in the sentences that satisfy the rule with the labels in the rule.
- Recalculate the confidence of each remaining rule based on the modified data from step 1.
- Repeat step 1 and 2 until no rule left with confidence higher than the *minconf* value (we used 90%).

(Details skipped)

Experimental results (Jindal and Liu, AAAI-06)

- Identifying Gradable Comparative Sentences
  - precision = 82% and recall = 81%.
- Classification into three gradable types
  - SVM gave accuracy of 96%
- Extraction of comparative relations
  - LSR (label sequential rules): F-score = 72%

Roadmap

- Opinion mining – the abstraction
- Document level sentiment classification
- Sentence level sentiment analysis
- Feature-based opinion mining and summarization
- Comparative sentence and relation extraction
- **Summary**

Summary

Two types of evaluations have been discussed

- Direct opinions
  - Document level, sentence level and feature level
  - Structured summary of multiple reviews
- Comparisons
  - Identification of comparative sentences
  - Extraction of comparative relations
- Very challenging problems
  - Current techniques are still primitive
- Industrial applications are coming …