Extracting Product Feature Assessments from Reviews

Ana-Maria Popescu
Oren Etzioni
http://www.cs.washington.edu/homes/amp

Overview

Motivation & Terminology
Opinion Mining Work
Overview of OPINE
Product Feature Extraction
Customer Opinion Extraction
Experimental Results
Conclusion and Future Work

Motivation

Reviews abound on the Web
- consumer electronics, hotels, etc.
- Automatic extraction of customer opinions can benefit both manufacturers and customers

Other Applications
- Automatic analysis of survey information
- Automatic analysis of newsgroup posts

Terminology

Reviews contain features and opinions.

Product features include:

Parts	*the cover of the scanner*
Properties	*the size of the Epson3200*
Related Concepts	*the image from this scanner*
Properties & Parts of Related Concepts	*the image size for the HP610*

Product features can be:

Explicit	*the size is too big*
Implicit	*the scanner is not small*

Terminology

Reviews contain features and opinions.

Opinions can be expressed by:

Adjectives	*noisy scanner*
Nouns	*scanner is a disappointment*
Verbs	*I love this scanner*
Adverbs	*the scanner performs beautifully*

Opinions are characterized by polarity (+, -) and strength *(great > good).*

Opinion Mining Work

Extract positive/negative opinion words
- Hatzivassiloglou & McKeown'97, Turney'03, etc.

Opinion Mining Work

Extract positive/negative opinion words

Hatzivassiloglou & McKeown'97, Turney'03, etc.

Classify reviews as positive or negative

Turney'02, Pang'02, Kushal'03

Opinion Mining Work

Extract positive/negative opinion words

Hatzivassiloglou & McKeown'97, Turney'03, etc.

Classify reviews as positive or negative

Turney'02, Pang'02, Kushal'03

Identify feature-opinion pairs together with the polarity of each opinion

Hu & Liu'04, Hu & Liu'05

Opinion Mining Work

Extract positive/negative opinion words

Hatzivassiloglou & McKeown'97, Turney'03, etc.

Classify reviews as positive or negative

Turney'02, Pang'02, Kushal'03

Identify feature-opinion pairs together with the polarity of each opinion

Hu & Liu'04, Hu & Liu'05

OPINE: High-precision feature-opinion extraction, opinion polarity and strength extraction

The OPINE System

Hotel Majestic, Barcelona:			HotelNoise
OpinionPhrase	Rank	Polarity	Frequency
Deafening	1	-	2
Loud	2	-	7
Silent	3	+	3
Quiet	4	+	4

Sample OPINE output in the Hotel domain

KIA Overview

OPINE is built on top of KIA, a domain-independent IE system which extracts concepts and relationships from the Web.

Given relation R and pattern P
- KIA instantiates P into extraction rules for R
- KIA extracts candidate facts from the Web
- Each fact is assessed using a form of PMI:

$$\text{PMI}(\textit{Seattle, is a city}) = \frac{\text{Hits}(\textit{"Seattle is a city"})}{\text{Hits}(\textit{"Seattle"})}$$

is a city = discriminator for the IS-A relationship

OPINE Overview

Input: product class C, reviews R

Output: set of feature-opinion pairs {(f,o)}.

- R' ← parseReviews(R)
- E ← findExplicitProductFeatures(R', C)
- O ← findOpinions(R', E)
- CO ← clusterOpinions(O)
- I ← findImplicitFeatures(CO, E)
- RO ← solveOpinionRankingCSP(CO)
- {(f, o)} ← outputFeatureOpinionPairs(RO, I ∪ E)

Explicit Feature Extraction

Given product class C

1. **Extract parts and properties of C**
   Recursively extract parts and properties of C's parts and properties, etc.
2. **Extract related concepts of C**
   (Popescu & all, 2004)
   Extract parts and properties of related concepts

13

Parts and Properties

Extract review noun phrases with frequency f > k as potential meronyms.
Assess candidates using discriminators D derived from patterns P:

Example: C=*scanner*, M=*size*, P= *[M] of C*

P = *[M] of C* → D_0= *[M] of scanner* ... D_k= *[M] of Epson 3200.*

$$\text{PMI}(\textit{size, [M] of scanner}) = \frac{\text{Hits}(\textit{“size of scanner”})}{\text{Hits}(\textit{“ of scanner”}) * \text{Hits}(\textit{“size”})}$$

...

$$\text{PMI}(\textit{size, [M] of Epson3200}) = \frac{\text{Hits}(\textit{“size of Epson 3200”})}{\text{Hits}(\textit{“ of Epson 3200 ”}) * \text{Hits}(\textit{“size”})}$$

Compute $PMI_T(M, P) = f(PMI(M,D_0), \ldots PMI(M, D_k))$.
Convert $PMI_T(M, P_0) \ldots PMI_T(M, P_j)$ **into binary features for a NB classifier (NBC).**
Retain meronyms M with p(meronym(M, C)) > t.
Separate parts from properties using WordNet and Web information.

14

OPINE Overview

Input: product class C, reviews R
Output: set of feature-opinion pairs {(f,o)}.

R' ← parseReviews(R)
E ← findExplicitFeatures(R', C)
O ← findOpinions(R', E)
CO ← clusterOpinions(O)
I ← findImplicitFeatures(CO, E)
RO ← solveOpinionRankingCSP(CO);
{(f, o)} ← outputFeatureOpinionPairs(RO, I ∪ E)

15

Opinion Extraction

Given feature f and sentence s containing f
Extract phrases whose head modifies head(f)

Example

f = resolution	*s = ... great resolution ...*
f = scanner	*s = ... scanner is white ...*
f = scanner	*s = ... scanner is a horror ...*
f = scanner	*s = I hate this scanner.*
f = scanner	*s = The scanner works well.*

OPINE then determines the polarity of each potential opinion phrase.

16

Polarity Extraction

Each potential opinion op has a semantic orientation label L(op): +, -, |

Initial SO Label Assignment
OPINE derives an initial label for each potential opinion:
SO(op) = PMI(op, good) - PMI(op, bad).
If SO(op) < t or Hits(op) < t_1, L(op) = “|” (neutral).
Else
If SO(op) > 0, L(op) = “+”.
Else L(op) = “-”.

Final SO Label Assignment
OPINE uses constraints to derive a final set of labels
WordNet constraints — antonym(operative, inoperative)
Conjunction/disjunction constraints — attractive, but expensive
Iteration i :
$L_i(op) = f(L_{i-1}(op_0), L_{i-1}(op_1) \ldots L_{i-1}(op_k))$
Termination Condition:
Labels remain constant over consecutive iterations.

17

OPINE Overview

Input: product class C, reviews R
Output: set of feature-opinion pairs {(f,o)}.

R' ← parseReviews(R)
E ← findExplicitFeatures(R', C)
O ← findOpinions(R', E)
CO ← clusterOpinions(O)
I ← findImplicitFeatures(CO, E)
RO ← solveOpinionRankingCSP(CO)
{(f, o)} ← outputFeatureOpinionPairs(RO, I ∪ E)

18

Implicit Properties

Adjectival opinions refer to implicit or explicit properties
Example: *slow driver speed, slow driver*

OPINE extracts properties corresponding to adjectives and uses them to derive implicit features

Clarity:	*intuitive understandable clear straightforward*
Noise:	*silent noisy quiet loud deafening*
Price:	*cheap inexpensive affordable expensive*

Implicit Features:

the interface is intuitive	*clarity(interface): intuitive*
straightforward interface	*clarity(interface): straightforward*

Clustering Adjectives

Generate initial clusters using WordNet syn/antonyms.
Clusters A_i and A_j are merged if there exist multiple elements a_i , a_j s.t. a_i is similar to a_j with respect to WordNet:
similar(a1, a2): derived(a1, C), att(C, a2).
similar(a1, a2): att(C1, a1), att(C2, a2), subclass(C1, C2), etc.
For each cluster A_i
OPINE uses queries such as
[a_1, a_2 and X] [a_1, even X] , [a_1, or even X], etc.
to extract additional related adjectives a_r from the Web.
If multiple a_r are elements of cluster A_r
$A_i + A_r = A'$ {intuitive} + {clear, straightforward}
Generate adjective cluster labels
WordNet: big=valueOf(size)
Add suffixes to cluster elements -iness, -ity

Rank Opinion Phrases

Initial opinion phrase ranking
Derived from the magnitude of the SO scores:
|SO(great)| > |SO(good)|: great > good
Final opinion phrase ranking
Given cluster A
Use patterns such as
[a, even a'] [a, just not a'] [a, but not a'], etc.
to derive set S of constraints on relative opinion strength
c = silent > quiet c=deafening > loud
Augment S with antonymy/synonymy constraints
Solve CSP_S to find final opinion phrase ranking

HotelNoise: *deafening > loud > silent > quiet*

Opinion Sentences

Opinion sentences are sentences containing at least one product feature and at least one corresponding opinion.

Determining Opinion Sentence Polarity
Determine the average strength s of sentence opinions op
If s > t,
Sentence polarity is indicated by the sign of s
Else
Sentence polarity is that of the previous sentence

Experimental Results

Datasets: 7 product classes, 1621 reviews
5 product classes from Hu&Liu'04
2 additional classes: Hotels, Scanners

Experiments:

Feature Extraction:	Hu&Liu'04 vs. OPINE
Opinion Sentences:	Hu&Liu'04 vs. OPINE
Opinion Phrase Extraction & Ranking:	OPINE

OPINE vs. Hu&Liu

Feature Extraction
OPINE improves precision by 22% with a 3% loss in recall.
Increased precision is due to Web-based feature assessment.

Opinion Sentence Extraction
OPINE outperforms Hu & Liu on opinion sentence extraction: 22% higher precision, 11% higher recall
OPINE outperforms Hu & Liu on sentence polarity extraction: 8% higher accuracy
OPINE handles adjectives, noun, verb, adverb opinions and limited pronoun resolution. OPINE also uses a more restrictive definition of opinion sentence than Hu & Liu.

OPINE Experiments

Extracting opinion phrases for a given feature:
- **P = 86%, R = 82%**
- Parser errors reduce precision
- Some neutral adjectives can acquire a pos/neg polarity in context - these adjectives can lead to reduced precision/recall

Opinion Phrase Polarity Extraction
- **P = 91%**
- Precision is reduced by adjectives which can acquire either a positive or a negative connotation: visible

Ranking Opinion Phrases Based on Strength
- **P = 93%**

25

Conclusion & Future Work

- OPINE is a high-precision opinion mining system which extracts fine-grained features and associated opinions from reviews.
- OPINE successfully uses the Web in order to improve precision.

Future Work
- Use OPINE's output to generate review summaries at different levels of granularity.
- Augment the opinion vocabulary.
- Allow comparisons of different products with respect to a given feature.

26