HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

WEB MINING

LECTURE 05: OPINION MINING (3/3)

ONE LOVE. ONE FUTURE.

[4] Entity detection & assignment

- In product reviews, it is often known who is being reviewed
- However, on forums, it is necessary to identify the entity audience the comments are aimed at
  - Task 1: Detect the entity in the sentence
  - Task 2: Assign the entity to the sentence (do not specify the entity to be evaluated)

Assumption of sentiment consistency

Eg.1:

(1) I bought Camera-A yesterday. (2) I took some pictures in the evening in my living room. (3) The images are very clear. (4) They are definitely better than those from my old Camera-B. (5) The battery is very good too.

Eg.2: **(4) → Camera-A > Camera-B**

(1) I bought Camera-A yesterday. (2) I took a few pictures in the evening in my living room. (3) The images were very clear. (4) They were definitely better than those from my old Camera-B. (5) The pictures of that camera were blurring for night shots, but for day shots it was ok

Problem statement

- A thread t containsposts $<p_1, p_2,..., p_n>$
- A post p contains sentences $<s_1, s_2,..., s_m>$
- A sentences s contains an entity set reviews ε is a subset of set of all entities $E = \{e_1, e2,...\}$
- An entity e may be explicitly or implicitly in a sentence s

Problem statement (2)

- Ex: "*Camera-A looks really pretty. The battery lasts very long*"
- Most sentences refer to only one entity ($|\varepsilon|=1$)
- Sentences involving more than one entity are usually comparative sentences ($|\varepsilon|=2$)
  - "*Camera-A is better than Camera-B*"
- Assuming sentences in a post are all meant to evaluate the entity object (in fact, there are also unrelated sentences, e.g. greeting)

Problem statement (3)

- Given a set T threats in domain:
  - Task 1 - Entity Detection: Detect set of entities E in T
  - Task 2 - Entity Assignment: Assign each sentence in T with one or several entities in E

Task 1 - Entity Detection

- Unsupervised method based on sequential pattern mining using an original entity set $E^{(0)} = \{e_1, e_2, \ldots, e_n\}$

Step 1. Prepare data

Step 2. Sequential pattern mining

Step 3. Extract candidate

Step 4. Filter candidates

Step1. Prepare data

- Search for all sentences containing entities in original set; replace entity name (containing one or more words) with the generic name ENTITYXYZ
- Generate series by selecting window 5 from before and after entity; each element is a word/word of type

Hiiiiiiiii/NNP SK/NNP -/: ,/, dont/NN be/VB mad/JJ everyone/NN doesnt/NN have/VBP a/DT **n95**/CD phone/NN fetish/NN ducky/JJ

mad/JJ everyone/NN doesnt/NN have/VBP a/DT **ENTITYXYZ** /CD phone/NN fetish/NN ducky/JJ

<{JJ, mad}{NN, everyone}{NN, doesnt}{VBP, have}{DT, a}{CD, ENTITYXYZ}{NN, phone}{NN, fetish} {JJ, ducky}>

Step 2. Sequential pattern mining

- Min support = 0.01
- Patterns must contain {POS, ENTITYXYZ}
- Patterns must have length >= 2
- E.g.: <{IN}, {DT}, {NNP, ENTITYXYZ }, {is}>

Step 3. Extract candidate

- Search for entities match generated patterns

The/DT misses/VBZ has/VBZ currently/RB got/VBN **a/DT Nokia/NNP 7390/CD** at/IN the/DT end/NN of/IN the/DT day,/VBG all/DT she/PRP does/VBZ is/VBZ text/NN and/CCmake/VB calls,/NN but/CC the/DT reception/NN is/VBZ terrible,/VBG where/WRB my/PRP$ 6233/CD would/MD get/VB full/JJ bars/NNS hers/PRP would/MD only/RB get/VB 1/CD or/CC 2./CD

<{DT}, {NNP, ENTITYXYZ}, {CD}> ~ a/DT **Nokia**/NNP 7390/CD

<{DT}, {NNP}, {CD, ENTITYXYZ}, {IN}> ~a/DT Nokia/NNP **7390**/CD at/IN

Step 4. Filter candidates

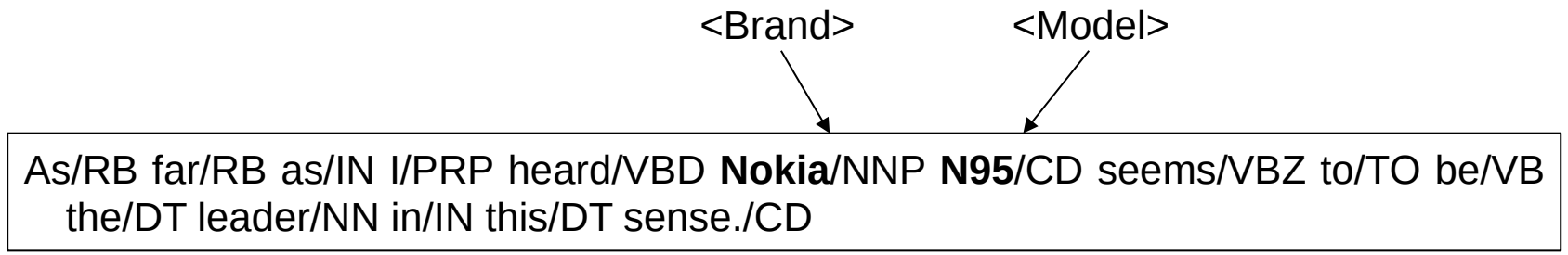
- Eliminate entities whose POS is different from the POS most popular with this candidate
- For example, 'accessories' is usually labeled NNS so 'accessories/CD' will be excluded

You/PRP can/MD also/RB be/VB sure/JJ it/PRP will/MD work/VB **with/IN all/PDT the/DT Sony/NNP Ericsson/NNP walkman/NN phone/NN accessories/CD**

<{IN}{DT}{CD, ENTITYXYZ}> → accessories **(wrong)**

Step 4. Filter candidates (2)

- Use the <Brand Model> (“Moto Razr V3”) to search for brand and model pair
- Use syntax patterns to find competing brands: A and B; A or B; A vs B; A more than B



Task 2 - Entity assignment

- Comparison sentence
  - Comparative: "*Camera-X's battery life is longer than that of Camera-Y*"
  - Equal: "*Camera-X and Camera-Y are of the same size*"
  - Non-comparable: "*Camera-X and Camera-Y have different shapes*"
  - Superlatives: "*Camera-X's battery life is the longest*"

Sentiment consistency

- Suppose entity e first appears in sentence s0 and next sentence is s1 .
- (1) If s0 is a normal sentence
  - If s1 is a normal sentence then it is assigned to e
  - If s1 is a comparison sentence, e will be compared with a new entity (needs to be introduced).
- (2) If s0 is a comparative sentence
  - If s0 is a comparative sentence; s1 represents positive/negative emotion and contains no entity then it is assigned to better/worse entity

Sentiment consistency (2)

- If s0 is an equal or non-comparable sentence, because we are not sure which entity s1 refers to, we assign it to the entity that came before s0.
- If s1 is a comparative, s1 is assigned to the entity in s1
- (3) If s0 is the superlative sentence
  - If s1 is a normal sentence, we assign it to the best entity mentioned in s0 .
  - If s1 is a comparative, s1 is assigned to the entity in s1

Algorithm

- s_i.entity: Entity mentioned in s_i
- s_i.superiorEntity: Better entity in comparative sentence
- s_i.inferiorEntity: worse entity in comparative sentence
- ***opinion***(): Function to determine emotions in normal sentences
- ***compOpinion***(): Function to determine emotions in comparison sentences

for each sentence s_i in sequence in a post **do**
1 **If** s_i is not a comparative sentence **then**
2 **if** s_i contains an explicit entity **then**
3 s_i.Entity ← the explicit entity of the sentence s_i
4 **else** // s_i does not contain an explicit entity
5 **if** s_{i-1} is not a comparative sentence **then**
6 s_i.Entity ← s_{i-1}.Entity
7 **elseif** a superior entity and an inferior entity were discovered in s_{i-1} **then**
8 ***opinion***(s_i); // opinion mining
9 **if** s_i's first clause is a positive clause **then**
10 s_i.Entity ← s_{i-1}.superiorEntity
11 **elseif** s_i's first clause is a negative clause **then**
12 s_i.Entity ← s_{i-1}.inferiorEntity
13 **else** s_i.Entity ← s_{i-1}.superiorEntity
14 **else** s_i.Entity ← s_f.Entity, entities of the last sentence that is not a comparative sentence
15 **else** // s_i is a comparative sentence
16 **if** no entity is mentioned in s_i **then**
17 s_i.Entity ← s_{i-1}.Entity
18 **else** s_i.Entity ← $\{s_{i-1}.\text{Entity}\} \cup \{\text{entities in } s_i\}$;
19 $\langle s_i.\text{inferiorEntity}, s_i.\text{superiorEntity}\rangle$ ← ***compOpinion***(s_i)

Sentiment Analysis

- Analyze sentiment of a sentence towards an entity assigned to sentence based on the evidence::
  - Opinion words: great, good, bad, poor; "*the battery of this camera lasts **long***"/ "*This program takes a **long** time to run*"
  - Opinion phrases: "*cost someone an arm and a leg*", "a good deal of"
  - Negative: not, "not only ... but also"
  - Clause 'but': "*The picture quality is great*, ***but*** *not the battery life*"

Specification language

```
<rule>      :=  <item> "=>" <action>
<item>      :=  <word> | <word> "[" <type> "]"
<word>      :=  [a-z]+
<type>      :=  JJ | RB | NN | VB | …
<action>    :=  Po | Ne | Neu | Ng | But
```

like[VB] => Po

```
<rule>        :=  <pattern> "=>" <action>
<pattern>     :=  <exp> "+" <target> "+" <exp>
                | <exp> "+" <target> | <target> "+" <exp>
<exp>         := <element> | <exp> "+" <element>
                | <exp> "+" <distance> "+" <exp>
                | <exp> "+" <distance>
                | <distance> "+" <exp>
                | !<num> "+" !<item> "+" <exp>
                | <exp> "+" !<num> "+" !<item>
                |<exp> "+" !<num> "+" !<item> "+" <exp>
<element>     :=  <item> | item "/" element
<item>        :=  <indicator> | <word>
<indicator>   :=  <indicatorSym>
                | <indicatorSym> "[" <type> "]"
<target>      :=  <indicator> "[T]" | <word> "[T]"
<indicatorSym> := Po | Ne | Neu | Ng | But
<word>        :=  [a-z]+ | [a-z]+ "[" <type> "]"
<distance>    :=  <num> | <num> - <num>
<num>         :=  0 | [1-9][0-9]*
<action>      :=  <outcome>| !<outcome>
<outcome>     :=  PO | NE | NEU | NG | BUT
<type>        :=  JJ | RB | NN | VB | …
```

Example

The picture quality of this camera is not good, reaction is too slow, but the battery life is long.

*The picture quality is not**[Ng]** good**[Po]**, reaction is too slow**[Neu]**, but**[But]** the battery life is long**[Neu]**.*

too + Neu[JJ][T] => NE

*The picture quality is not[Ng] good[Po], reaction is too slow**[NE]**, but[But] the battery life is long[Neu].*

*The picture quality is not[Ng] good**[Negative]**, reaction is too slow[NE], but[But] the battery life is long[Neu].*

Analyze comparative sentences

- Comparative sentence matches one of patterns:
  - a). pronoun + compkey + prodname,
  - b). prodname + compkey + pronoun,
  - c). prodname + compkey + prodname
  - d). pronoun + superkey
  - e). prodname + superkey
  - f). as + JJ + as (ngoại trừ “as long as” và “as far as”)
- Where compkey is comparison word, prodname is product name, superkey is comparative word

Analyze comparative sentences (2)

- Short adjectives/adjectives are changed to more/most by adding -er/-est (higher/highest)
- Some irregular cases: good/better/best
- Longer adjectives/adverbs add more/most
- Apply rules:
  - more/most + Pos → Positive
  - more/most + Neg → Negative
  - less/least + Pos → Negative
  - less/least + Neg → Positive
- Other words like win, prefer, superior, inferior
  - "*In term of battery life, Camera-X is **superior** to Camera-Y*"

Result evaluation

- Datasets:
  - HowardForums: Movie reviews
  - AVSforums: Plasma/LCD TVs, Projectors and DVD players

Data sets	No. of threads	No. of posts	No. of Product	No. of comparatives	Total no. of sentences
Howard	31	446	171	664	2589
AVS	33	307	180	408	1796
Total	64	753	351	1072	4385

Result evaluation (2)

- CRF: Entity Detection by CRF
- NET: Entity Detector
- Baseline1: Get the last entity of previous sentence if the current sentence does not contain the entity.
- Baseline2: Gets the first entity of previous sentence if the current sentence does not contain an entity.
- ED (k-com): Given comparative sentences
- ED (unk-com): Need to detect comparative sentences

Result evaluation (3)

Table 2: Results of entity discovery

Datasets	CRF		NET		EI (1-3)		EI (1-4)		EI (1-5)	
	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.
Howard	0.40	0.91	0.48	0.35	0.87	0.48	0.86	0.58	0.81	0.83
	F = 0.56		F = 0.40		**F = 0.62**		**F = 0.69**		**F = 0.82**	
AVS	0.37	0.89	0.42	0.29	0.84	0.47	0.84	0.59	0.77	0.80
	F = 0.52		F = 0.34		**F = 0.60**		**F = 0.69**		**F = 0.78**	

Table 3: Experimental results for entity assignment

Data sets	Next Sentences (Accuracy)				All Sentences (Accuracy)				Comp Ident.		
	Baseline1	Baseline2	ED (k-com)	ED (unk-com)	Baseline1	Baseline2	ED (k-com)	ED (unk-com)	Prec.	Recall	F
HowardForums	82.4%	83.3%	93.4%	90.3%	80.3%	82.1%	88.2%	86.7%	85.2%	84.2%	84.7%
AVSforrum	79.6%	80.9%	91.2%	89.6%	76.7%	77.9%	87.2%	85.0%	82.2%	84.9%	83.5%
Average	81.0%	82.1%	92.3%	89.9%	78.5%	80.0%	87.7%	85.9%	83.7%	84.6%	84.1%
Col#	1	2	3	4	5	6	7	8	9	10	11

[5] Exploiting comparative sentences

- User reviews on the Internet for products:
  - 90% din the form of direct reviews ("*the picture quality of Camera X is great*")
  - 10% as comparison ("*the picture quality of Camera X is better than that of Camera Y*")

Problem statement

- Many comparative sentences don't have a direct comparison word, the emotion of the same word depends on the context
- "*the battery life of Camera X is longer than Camera Y*"
- "*Program X's execution time is longer than Program Y*"
- Choosing sentences as context leads to including a lot of irrelevant information

Problem statement (2)

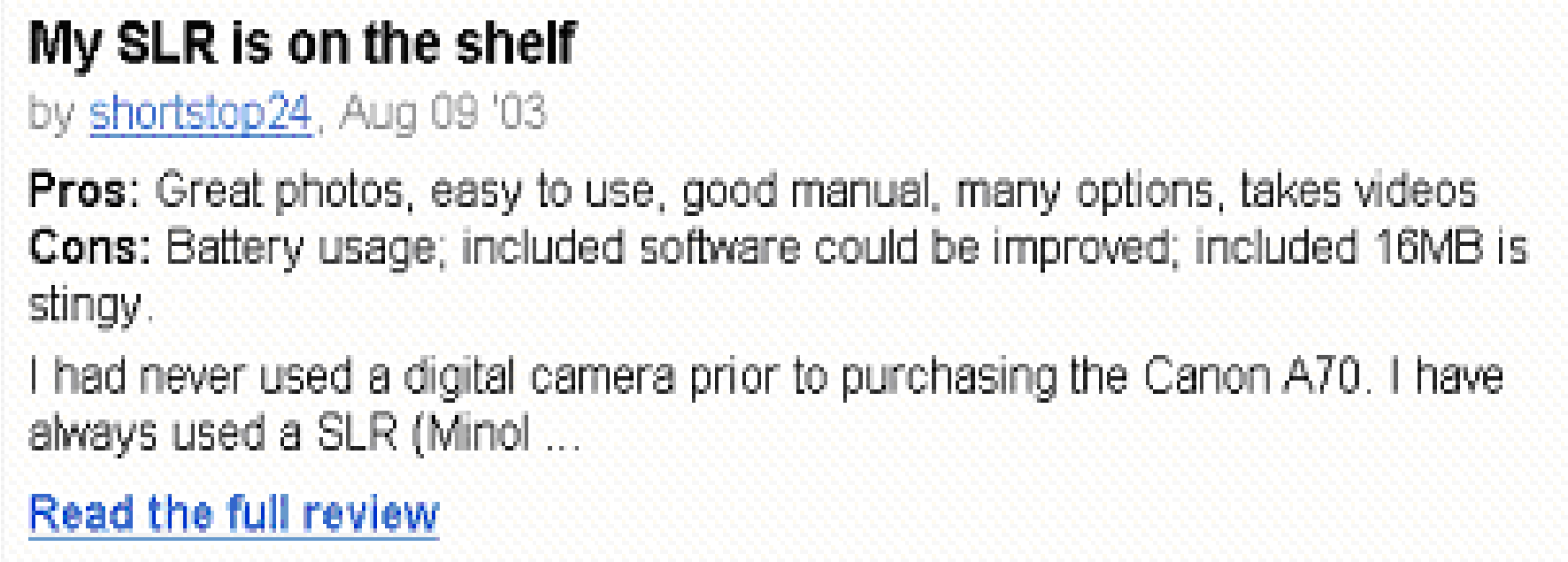
- Context: evaluated entity + comparison word
- How to identify emotions expressed by context?
- Using external knowledge (epinions.com) to identify opinion orientation from context
  - Epinions.com clearly separates positive and negative comments
  - what context often appears in positive or negative comments?

Problem statement (3)

- Given a relation corresponding to the comparative sentence
  - <C (comparison word), F (feature), e1 (entity 1), e2 (entity 2), type (comparison type)>
- "*Camera X has longer battery life than Camera Y*"
  - <longer, battery life, Camera X, Camera Y, comparative>
- Determine which entity is 'better'

Pros and Cons

- **Pros**:
  - great photos <photo>
  - easy to use <use>
  - good manual <manual>
  - many options <option>
  - takes videos <video>



My SLR is on the shelf

by shortstop24, Aug 09 '03

Pros: Great photos, easy to use, good manual, many options, takes videos
Cons: Battery usage; included software could be improved; included 16MB is stingy.

I had never used a digital camera prior to purchasing the Canon A70. I have always used a SLR (Minol ...

Read the full review

Type 1 -er/-est

- Short adjectives/adverbs add -er/est
- C express emotion (better/best): If positive, choose e1, if negative, choose e2
- C no emotion, F express emotion
  - "*Car X generates more noise than Car Y*"
  - C comparative + F positive → e1
  - C comparative (reduction) + F positive → e2
  - C comparative + F negative→ e2
  - C comparative (reduction) + F negative → e1

Type 1 -er/-est (2)

- Both C and F no sentiment

$$OSA(F,C) = log\frac{Pr(F,C)Pr(C|F)}{Pr(F)Pr(C)}$$

  - If $OSA_P(F,C) > OSA_N(F,C)$ → e1; otherwise chooose e2

Type 1 -er/-est (3)

- Calculate OSAP(F,C):
  - Count number times C (and synonyms) and F (and synonyms) appear together in Pros
  - Count number times antonyms of C and F appear in Cons
  - Count the number of times C and antonyms of F appear together in Cons
- Use Wordnet to get synonyms and antonyms
- Do the same with OSAN(F,C)

Type 1 -er/-est (4)

- If C exhibits a feature
  - "*Camera X is smaller than Camera Y*"
  - Count number times C occurs in Pros and Cons and choose the larger value

Type 2 more/less + adj/adv

- Adjectives/adverbs express emotions
  - *"Car X has more beautiful interior than Car Y"*
- Adjectives/adverbs don't express emotions
    - Adjectives/adverbs describe features
    - Adjectives/adverbs don't describe features
- Negative:
  - *"Camera X's battery life is not longer than that of Camera Y"*

Evaluation of results

- The data includes camera ratings, DVD players, MP3 players, Intel vs AMD, Coke vs Pepsi, Microsoft vs Google; laptop, mobile phone

Data Sources	No. of Comparative Sentences
(Jindal and Liu 2006)	418
Reviews and forum posts	419
Total	837

Evaluation of results (2)

- Baseline-84%: Always takes the first entity

	EntityS1 Preferred			**EntityS2 Preferred**		
	Prec.	*Rec.*	*F*	*Prec.*	*Rec.*	*F*
PCS (OSA)	0.967	0.966	0.966	0.822	0.828	0.825
PCS: No Pros & Cons	0.925	0.980	0.952	0.848	0.582	0.690
PCS (PMI)	0.967	0.961	0.964	0.804	0.828	0.816

	EntityS1 Preferred			**EntityS2 Preferred**		
	Prec.	*Rec.*	*F*	*Prec.*	*Rec.*	*F*
PCS (OSA)	0.896	0.877	0.886	0.696	0.736	0.716
PCS: No Pros & Cons	0.722	1.000	0.839	0.000	0.000	0.000
PCS (PMI)	0.894	0.855	0.874	0.661	0.736	0.696

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !