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 (2/3)

ONE LOVE. ONE FUTURE.

Sub-problems in ABSA

1. Aspect extraction
   - “*The voice quality of this phone is amazing*”
   - “*I love this phone*” (GENERAL aspect)
2. Aspect-based Opinion Mining
   - “*The voice quality of this phone is amazing*” → Positive
   - “*I love this phone*” → Positive

Aspect-based Opinion Mining

- Supervised approach
  - Use dependency syntax for extracting syntactic features
  - High results but difficult to adjust to new fields
- Classical approach
  - High results on multiple fields
  - Requires knowledge in language and specific field,
  - Contains many rules

Agenda

[1] Sentiment Ontology Tree

[2] Sentiment Analysis on Twitter

[3] Sentiment Analysis on social network

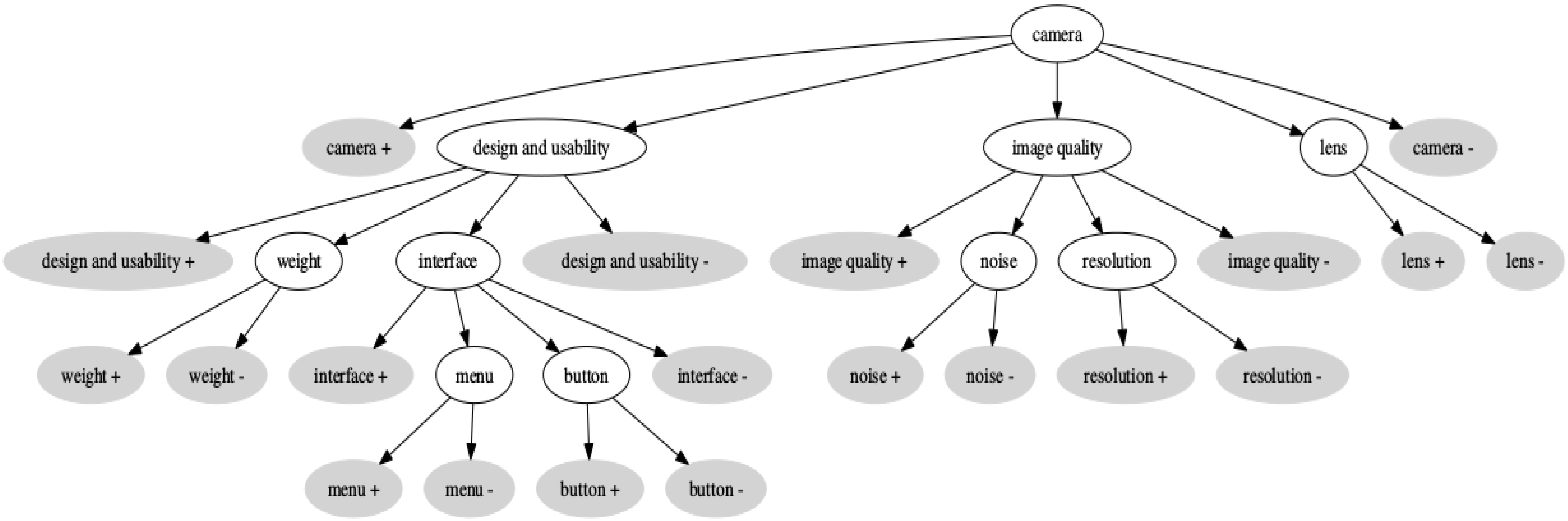
[4] Entity detection and assignment

[5] Comparison Mining

[1] Sentiment Ontology Tree

- Hierarchical classification based on hierarchical learning technique
- Sentiment Ontology Tree (SOT):
  - Demonstrate relationships ancestors - descendants among the aspects in domain
  - Each aspect comes with nodes that express sentiment for that aspect

Example of SOT



SOT definition

- $T(v, v^{+}, v^{-}, \mathbf{T})$
- v: root node express property v
- v^{+}: positive node of property v
- v^{-}: negative node of property v
- $\mathbf{T}$: set of sub-SOT of T: $T'(v', v'^{+}, v'^{-}, \boldsymbol{T}')$

HL-SOT

- sentence $x \in X$, $X = R^d$
- Set of nodes in tree: $Y = \{1, 2, \ldots, N\}$
- Label vector of x: $y = \{y_1, y_2, \ldots, y_N\} \in \{0,1\}^N$
- $\forall i \in Y$,
  - $y_i = 1$ if x is labeled by classifier of node i
  - $y_i = 0$ if x is node labeled by classifier of node i

Problem definition

- $y \in \{0,1\}^N$ *adapt a SOT if and only if*
  - $\forall i \in Y$, $\forall j \in A(i)$: *if* $y_i = 1$ *then* $y_j = 1$, *where* $A(i)$ *is set of ancestor node of node i*
- Let set of label vectors adapt SOT is τ
- Hierarchical classifier f: $X \rightarrow \tau$ generate vector y for each input vector x so that y satisfied a SOT

HL-SOT

- $y = f(x) = g(W \cdot x)$
- $W = (w_1, \ldots, w_N)^T$
  - w_i is weight of linear classifier of node i
- $y_i = w_i^T x \geq \theta_i$ if i is root or $y_j = 1$ where $\forall j \in A(i)$; otherwise $y_i = 0$
  - θ_i *threshold* of classifier of node i

Learning weight

- Training dataset $D = \{(r, l) \mid r \in X, l \in Y\}$
- Weight matrix W is initialized = 0
- Threshold vector θ is initialized = 0

Learning weight (2)

- For each example r_t, weight is updated as follow:

$$w_{i,t} = (I + S_{i,Q(i,t-1)} S_{i,Q(i,t-1)}^\top + r_t r_t^\top)^{-1} \times S_{i,Q(i,t-1)} (l_{i,i_1}, l_{i,i_2}, ..., l_{i,i_{Q(i,t-1)}})^\top$$

- I_{dxd} : identity matrix
- $Q(i, t\text{-}1)$: number of times parent node of node *i* have been set positive before
- $S_{i,Q(i,t-1)} = [r_{i,1}, ..., r_{i,Q(i,t-1)}]$
- Only update weight $w_{i,t}$ of node i if parent node of i is set positive

Learning weight (3)

- Update threshold of classifier

$$\theta_{t+1} = \theta_t + \epsilon(\hat{y}_{r_t} - l_{r_t}),$$

- where ε small real number to control update speed
- If classifier predict correctly, keep as it is
- If classifier predict incorrectly, as positive, increase θ
- If classifier predict incorrectly, as negative, decrease θ

Learning algorithm

Algorithm 1 Hierarchical Learning Algorithm HL-SOT

INITIALIZATION:

Each vector $w_{i,1}, i = 1, ..., N$ of weight matrix W_1 is set to be 0 vector

Threshold vector θ_1 is set to be 0 vector

BEGIN

for $t = 1, ..., |D|$ **do**

  Observe instance $r_t \in \mathcal{X}$

  for $i = 1, ...N$ **do**

    Update each row $w_{i,t}$ of weight matrix W_t by Formula 1

  end for

  Compute $\hat{y}_{r_t} = f(r_t) = g(W_t \cdot r_t)$

  Observe label vector $l_{r_t} \in \mathcal{Y}$ of the instance r_t

  Update threshold vector θ_t by Formula 2

end for

END

[2] Sentiment analysis on Twitter

- Tweet has maximum 140 characters
- 2011: Twitter has 190M users, 65M tweet per day
- User express sentiment on Twitter
- Sentiment analytic tools on Twitter: Tweetfeel, Twendz, Twitter Sentiment

Characteristic of tweet

- Tweet is shorter and more ambiguous then product review
- Product review has known target object; while tweet need an extra step to determine target object
- Relevant tweet provide context information for classifier
- General classifier (for general objects) is not appropriate for tweet classification

Problem definition

- Input: set of tweets consist target objects
- Output: analyze sentiment of tweets for target objects
  - Neutral:
  - Positive
  - Negative

Algorithm

1. Subjective/Objective classification: If tweet is objective → Neutral
2. Positive/Negative classification
3. Optimize graph of relevant tweets

- Classify using linear *SVM*

Preprocess

- Tag part-of-speech using OpenNLP
- Stemming using 20,000 words dictionary (e.g. ‘playing’ → ‘play’)
- Normalize using simple rules (e.g ‘gooood’ → ‘good’, ‘luve’ → ‘love’)
- Analyze dependency syntax using Minimum Spanning Tree

Independent features

- Content features: words, punctuations, emoticon, hashtag
- Vocabulary features: sentiment vocabulary of General Inquirer
- These are typical features used in general sentiment classifier

Extra objects

1) Noun phrases

“*I am passionate about Microsoft technologies, especially Silverlight*”

2) Extend using co-occurrence resolution

“Oh, Jon Stewart. How I love you so.”

3) Top K nouns and noun phrase are related to original object using PMI

Extra objects (2)

$$PMI(w,t) = \log \frac{p(w,t)}{p(w)p(t)}$$

- p(*w*,*t*): probability that *w* and *t* appears in the same corpus
- p(*w*): probability w appears in corpus
- p(*t*): probability *t* appears in corpus
- K = 20, corpus consists of 20M tweet

Extra objects (3)

4) Central words of noun phrase if PMI greater than a threshold

"Microsoft technologies" → *'technologies'*

"the price of iPhone" → *'price'*

"LoveGame by Lady Gaga" → *'LoveGame'*

Object-dependant features

- Object T
- w_i_*arg2*: transitive verb accepts T as object
  "*I love iPhone*" → '*love_arg2*'
- w_i_*arg1*: transitive verb accepts T as subject
- w_i_*it_arg2*: intransitive verb accepts T as subject
- w_i_*arg1*: noun or adjective accepts T as central words (in noun phrase)

Object-dependant features (2)

- *w_i_cp_arg1*: noun or adjective links to T using a copula (e.g. "*to be*")
- *w_i_arg*: adjective or intransitive verb appears as a imdepent senetence and T appears in previous sentence

  "*John did that. Great!*" → '*great_arg*'
- *arg1_v_w_i*: adverb support modifies verb that accepts T as subject

  "*iPhone works better with the CellBand*" → 'arg1_v_better'

Object-dependant features (3)

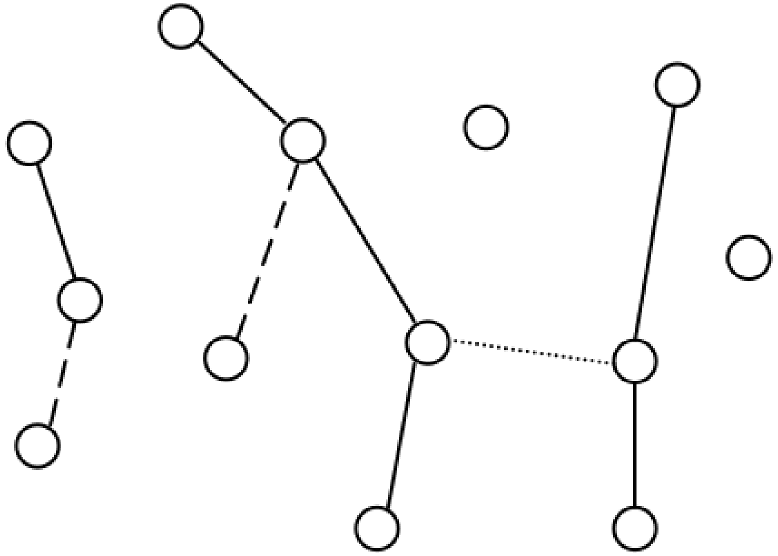
- If the feature is modified by a negative word, add the prefix "neg-"
  “*iPhone does not work better with the CellBand*” → ‘*arg1_v_neg-well*’, ‘*neg-work_it_arg1*’
- Set of negative words: *not*, *no*, *never*, *n’t*, *neither*, *seldom*, *hardly*

Graph optimization

- Using only content to classify sentiments can be incorrect for short tweet
- Requires more context information:
  - Retweet: keep original content
  - User’s tweets have information about object in a short window of time: assume that user do not change sentiment about the object
  - Reply: answer a tweet or feedback

Graph of tweets

- Solid line: tweet of same user
- Dash line: retweet
- Dot line: reply



Model evaluation

- Query set {*Obama*, *Google*, *iPad*, *Lakers*, *Lady Gaga*}
- Data: 459 positive, 268 negative and 1,212 neutral
- Consensus degree: 86% of 100 tweet
  - 1 positive-negative tweet
  - 13 neutral-negative/positive tweet

Feature evaluation

Features	Accuracy (%)
Content features	61.1
+ Sentiment lexicon features	63.8
+ Target-dependent features	**68.2**
Re-implementation of (Barbosa and Feng, 2010)	60.3

Target	Accuracy (%)
Exact target	65.6
+ all extended targets	**68.2**
- co-references	68.0
- targets found by PMI	67.8
- head nouns	67.3

*"No debate needed, heat can't beat **lakers** or celtics"* (negative by TS but positive by human)

*"why am i getting spams from weird people asking me if i want to chat with **lady gaga**"* (positive by TS but neutral by human)

*"Bringing iPhone and **iPad** apps into cars? http://www.speakwithme.com/ will be out soon and alpha is awesome in my car."* (positive by TS but neutral by human)

*"Here's a great article about Monte Veronese cheese. It's in Italian so just put the url into **Google** translate and enjoy http://ow.ly/3oQ77"* (positive by TS but neutral by human)

[3] Sentiment analysis in forums

- Analyze sentiment of product review
- Multilingual: English, French, Dutch
- Label set = {positive, negative, neutral}
- Use linguistic features
- Features based classifier

Vocabulary features

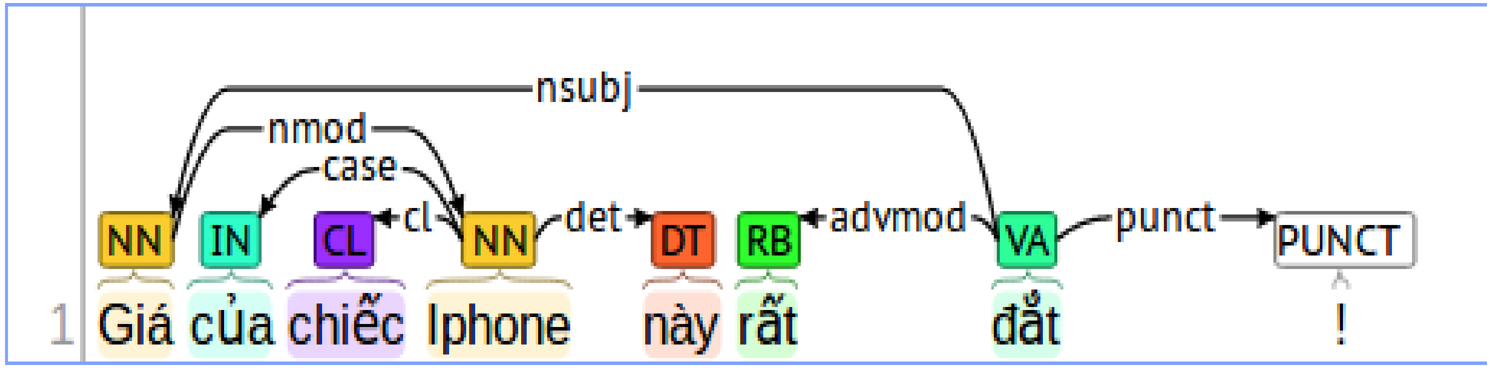
- Unigram: word/token in sentence; remove stopwords (publist.com)
- Stem: using Porter algorithm (e.g. ‘playing’ → ‘play’)
- Negative: vd “*not worth*”
- Document features: e.g. “*<u>même si</u> le film a eu beaucoup de succès, je le trouvais vraiment nul!*” (even though the movie had a lot of success, I really found it nothing!)

Syntax features

- Dept difference: between word/entity features in revert syntax tree and weight of word (English, Dutch)
- Distance : distance (by BFS) between word/entity features in revert syntax tree and weight of word (French)

Example of syntax feature

- Depth(‘giá’) = 1
- Depth(‘Iphone’) = 2
- Path_distance(‘giá’, ‘Iphone’) = 1



http://45.117.171.213/bknlptool/

Difference in languages

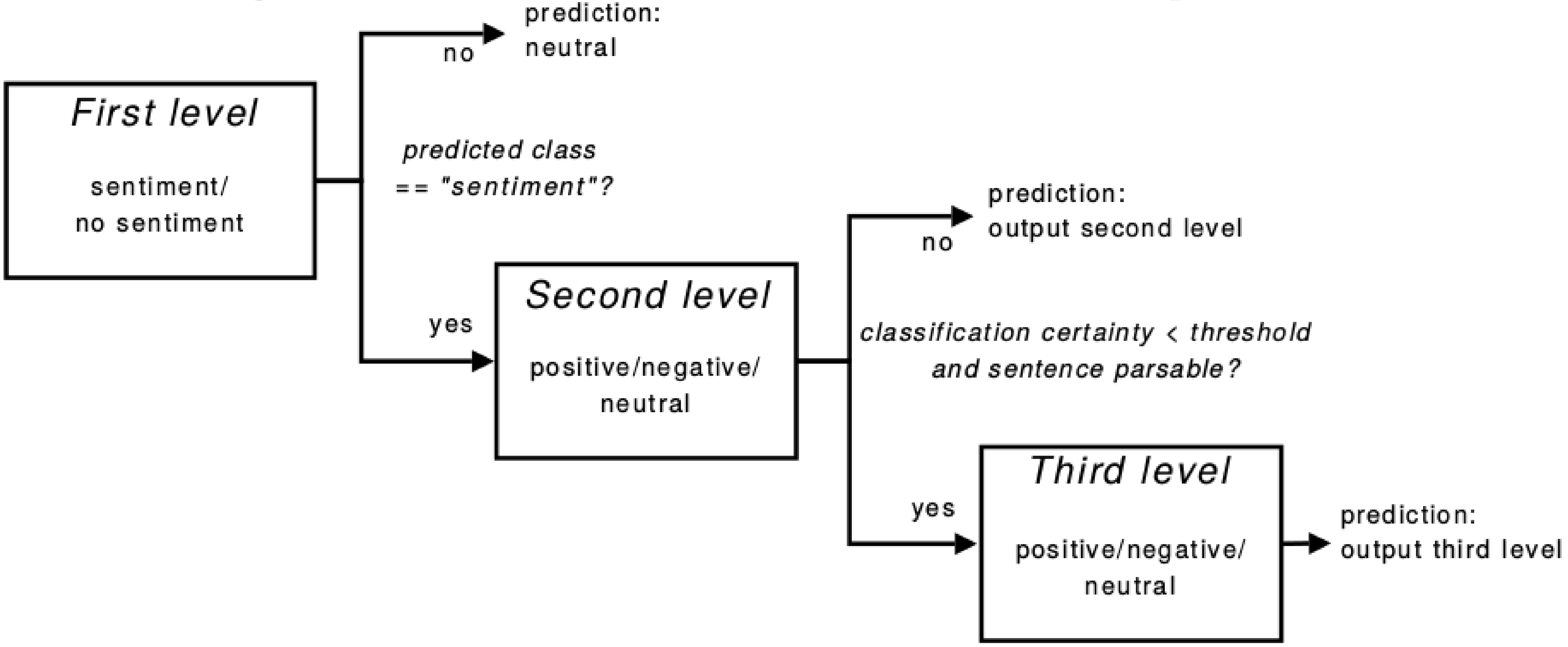
- Compound noun: ‘topfilm’ (top movie)
- Composed verb: “tegenvallen, valt tegen” (to be below expectations), “meevallen, valt mee” (turn out better than expected)
- “je ne suis pas d’accord” (I don’t agree)
- “L é tro bel cet voitur” (Elle est trop belle cette voiture - She is too beautiful, this car)
- j’aime ce film (I love this movie) /ce film est bien (this movie is good)
- de film is goed (this movie is good)

Classifiers

- Support Vectors Machines (SVMs): The classifier learns the support vectors to classify the object into two classes so that the margin is maximum (Weka)
- Multinomial Naive Bayes (MNB): Multiclass classifier based on Bayesian probability with assumption of probabilistic independence between features(Weka)
- Maximum Entropy (ME): The classifier is based on the maximum of entropy (MaxEnt)

Cascade model

Figure 1: Instantiation of the cascade model used in our experiments.



Active learning

- Target: Good quality model learning with minimal amount of labeled data
- Method: Automatic selection of examples to label from an unlabeled set (possibly based on some original labeled examples)

Uncertain sampling

- Criteria: Select examples where the classifier already has uncertainty. Degree of uncertainty based on :
  - Probabilistic Classifiers(MNB, ME)
  - Distance to hyperplane (SVM)
- Target:
  - Reduce redundancy in training data
  - Improved ambiguous examples (various emotions, different feelings for different entities)

Datasets

- User product comments and news articles, product reviews on forums, news sites
- Blog: skyrock.com, livejournal.com, xanga.com, blogspot.com
- Review pages: amazon.fr, ciao.fr, kieskeurig.nl
- Forums: fok.nl, forums.automotive.com
- Outliers: advertisement, spam, personal style

Datasets (2)

- Target objects: cars, movies; replace name with generic label 'CAR' or 'MOVIE'
- Remove questions
- Sentence level classification
- consensus: kappa = 82%
- Each language has 2500 neutral examples, 750 positive examples, and 750 negative examples
- Evaluation using 10-fold cross-validation

Experiment settings

- Cascade model:
  - Level 1: unigram
  - Level 2: + discourse + negation
  - Level 3: + parsed feature
- SC uni-lang (~ level 2) train on all data
- SC uni-lang-dist add features of distance between words and entities
- English: MNB; Ducht: SVM; French: ME

Experimental Results

Impact of features

Features	SVM	MNB	ME
Unigrams	85.45%	81.45%	84.80%
Unigrams & BSubjectivity	**86.35%**	83.95%	**87.40%**
Bigrams	85.35%	83.15%	85.40%
Adjectives	75.85%	82.00%	80.30%

Impact of bagging

Table 4: Results of the the first layer (English corpus) – 10-fold cross-validation.

Features	Precision neu/not neu	Recall neu/not neu	F-measure neu/not neu
Using bagging	96.05/51.69	62.15/94.06	75.47/66.71
No bagging	88.79/78.07	86.20/81.87	87.48/79.92

Experimental Results

Impact of cascade model

(a) English

Architecture	Accuracy	Precision pos/neg/neu	Recall pos/neg/neu	F-measure pos/neg/neu
Cascade with layers 1, 2 and 3	83.30	69.09/85.48/85.93	55.73/82.40/91.84	61.70/83.91/88.79
Cascade with layers 1 and 2	83.10	70.49/87.72/84.61	54.13/79.07/93.00	61.24/83.17/88.61
SC uni-lang	83.03	69.59/86.77/85.08	56.13/79.60/92.12	62.14/83.03/88.46
SC uni-lang-dist	80.23	60.59/78.78/86.57	59.87/82.67/85.60	60.23/80.68/86.08
SC uni	82.73	68.01/85.63/85.53	58.40/78.67/91.24	62.84/82.00/88.29

(b) Dutch

Architecture	Accuracy	Precision pos/neg/neu	Recall pos/neg/neu	F-measure pos/neg/neu
Cascade with layers 1,2 and 3	69.03	63.51/53.30/72.20	42.93/31.20/88.20	51.23/39.36/79.40
Cascade with layers 1 and 2	69.80	66.60/58.31/71.66	41.73/29.47/90.32	51.31/39.15/79.92
SC uni-lang	69.05	60.39/52.59/73.63	49.60/33.87/85.44	54.47/41.20/79.10
SC uni-lang-dist	68.85	61.08/54.52/72.20	43.73/30.53/87.88	50.97/39.15/79.27
SC uni	68.18	58.73/49.58/73.24	48.00/31.73/85.16	52.82/38.70/78.75

(c) French

Architecture	Accuracy	Precision pos/neg/neu	Recall pos/neg/neu	F-measure pos/neg/neu
Cascade with layers 1, 2 and 3	67.68	50.74/55.88/71.90	27.47/38.67/88.44	35.64/45.71/79.32
Cascade with layers 1 and 2	67.47	52.69/53.96/71.56	26.13/38.13/88.68	34.94/44.69/79.21
SC uni-lang	65.97	47.67/50.33/72.18	30.00/40.67/84.36	36.82/44.99/77.79
SC uni-lang-dist	65.97	47.67/50.33/72.18	30.00/40.67/84.36	36.82/44.99/77.79
SC uni	65.83	45.67/50.82/72.23	28.80/41.33/84.28	35.32/45.59/77.79

Experimental Results

Efficiency on neutral example

(a) English

Architecture	Precision	Recall	F-measure
Layer 1 of the cascade	88.79	86.20	87.48
Layer 1 and 2 of the cascade	84.61	93.00	88.61
Layer 2 of the cascade	85.08	92.12	88.46

(b) Dutch

Architecture	Precision	Recall	F-measure
Layer 1 of the cascade	74.49	82.00	78.07
Layer 1 and 2 of the cascade	71.66	90.32	79.92
Layer 2 of the cascade	73.73	85.88	79.34

(c) French

Architecture	Precision	Recall	F-measure
Layer 1 of the cascade	75.95	81.36	78.56
Layer 1 and 2 of the cascade	71.56	88.68	79.21
Layer 2 of the cascade	72.18	84.36	77.79

Experimental Results

Impact of domain

(a) Car domain

Architecture	Acc	Precision pos/neg/neu	Recall pos/neg/neu	F-measure pos/neg/neu
Cascade with layers 1, 2 and 3	70.65	74.04/62.32/71.45	55.78/39.33/87.28	63.62/48.23/78.57
SC uni-lang	70.84	69.67/63.02/72.83	60.22/43.56/84.48	64.60/51.51/78.22
SC uni-lang-dist	70.51	70.23/65.06/71.53	54.00/38.89/87.84	61.06/48.68/78.85
Cascade layers 1, 2 and 3 trained on movie domain	63.95	62.33/48.47/65.72	40.44/17.56/89.12	49.06/25.77/75.65

(b) Movie domain

Architecture	Acc	Precision pos/neg/neu	Recall pos/neg/neu	F-measure pos/neg/neu
Cascade with layers 1, 2 and 3	56.88	46.15/31.76/59.85	12.00/12.00/89.2	19.05/17.42/71.64
SC uni-lang	59.77	44.76/48.09/65.29	31.33/33.56/79.44	36.86/39.53/71.67
SC uni-lang-dist	62.05	48.70/51.81/66.04	29.11/31.78/84.80	36.44/39.39/74.26
Cascade layers 1, 2 and 3 trained on car domain	59.40	55.61/36.18/63.98	25.33/23.56/84.56	34.81/28.53/72.85

Experimental Results

Impact of syntax features on ambiguous examples

Table 6: Results with regard to the classification of ambiguous positive, negative and neutral sentences (Dutch corpus) that can be parsed – 10-fold cross-validation.

Features	Precision pos/neg/neu	Recall pos/neg/neu	F-measure pos/neg/neu
Unigrams	28.57/100/30.43	6.06/ 7.69/93.33	10.00/14.29/45.90
Unigrams + parse features	33.33/100/32.18	9.09/15.38/93.33	14.29/26.67/47.86

Impact on uncertain examples

Table 7: Results with regard to the classification of uncertain positive, negative and neutral sentences (English corpus) that can be parsed – 10-fold cross-validation.

Architecture	Precision pos/neg/neu	Recall pos/neg/neu	F-measure pos/neg/neu
Cascade with layers 1, 2 and 3	50.48/69.05/57.14	71.62/86.14/11.43	59.22/76.65/19.05
Cascade with layers 1 and 2	53.95/78.48/41.11	55.41/61.39/52.86	54.67/68.89/46.25

Experimental Results

Impact on outliers

Table 8: Results with regard to the classification of very noisy sentences that diverge from formal language (French corpus) – 10-fold cross-validation.

Architecture	Precision pos/neg/neu	Recall pos/neg/neu	F-measure pos/neg/neu
Cascade with layers 1, 2 and 3	60.87/52.63/83.41	24.28/15.87/97.11	34.71/24.39/89.74
SC uni-lang	61.25/55.00/83.90	28.32/17.46/96.56	38.74/26.51/89.78
SC uni-lang-dist	61.25/55.00/83.90	28.32/17.46/96.56	38.74/26.51/89.78

Error analysis

Table 9: Error analysis based on examination of 50 misclassified sentences in English, Dutch and French.

Id	Cause	English	Dutch	French	All
1	Features insufficiently known and/or wrong feature connotations	23	21	15	59
2	Ambiguous examples	12	8	8	28
3	Sentiment towards (sub-)entity	3	3	9	15
4	Cases not handled by negation	3	3	4	10
5	Expressions spanning several words	3	5	2	10
6	Understanding of the context or world knowledge is needed	2	2	4	8
7	Domain specific	0	3	3	6
8	Language collocations	2	2	2	6
9	A sentiment feature has multiple meanings	2	1	2	5
10	Language specific	0	2	1	3

Error analysis (2)

1. Lack of training data results in sentences that don't carry emotions but are labeled as positive/negative because they contain words that often appear in positive/negative sentences
2. Ambiguous sentences
3. Sentences with feelings towards another entity
4. Sentences with negative words
5. Emotions are expressed through metaphors
6. Emotions must be inferred from broad context (text) or general knowledge

Error analysis (3)

7. Requires knowledge of a narrow field
8. Emotions expressed through idioms
9. Emotions expressed through homonyms
10. Language-specific problems such as compound words in Ductch or accents in French

Error analysis (4)

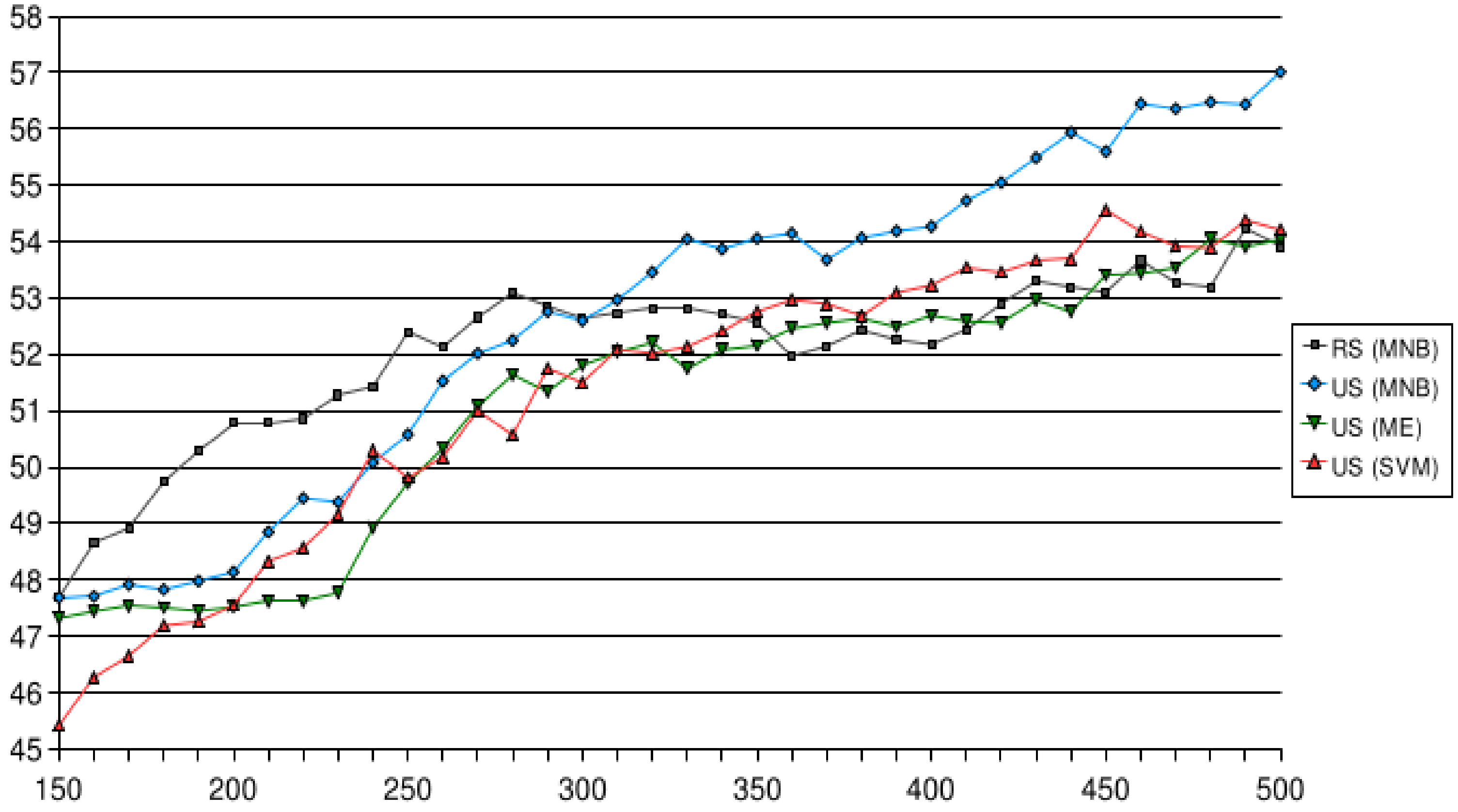
*2. A Good Year is a **fine** example of a **top-notch** director and actor out of their elements in a <u>sappy</u> romantic comedy <u>lacking</u> in …*
3. certainly more comfortable and rewarding than an <u>Audi Q7</u> and …
5. it is een schot in de roos (this is a shot in the bull’s eye) ~ they got it exactly right
6. I don t know maybe it s because I was younger back then but Casino Royale <u>felt more like a connect the dots exercise</u> than a Bond movie.
7. [...] attention pour avoir une chance de ne pas dormir au bout de 10 minutes, mieux vaut connaître les règles du poker […] (dormir ~ sleep)
*8. Casino Royale finally hits full-throttle in its second hour but Bond fans will find the movie hit-and-miss **at best***
9. Not a coincidence–GM used Mercedes' supplier for the new ... the interior plastics and wood trims is REALLY <u>cheap</u>. ... brown seats in a light colored car only make A Ferrari is <u>not cheap</u> to buy or run and residual values weaken if you use the car regularly.

Evaluate active learning

- Sample until have 500 example
- Using MNB classifier with unigram + discourse + negation features
- Evaluation on English
- Validation set: 1703 example: 1151 neutral, 274 positive, and 72 negative

Evaluate active learning (2)

	MNB		ME		SVM	
#Ex	RS	US	RS	US	RS	US
150	47.69	47.69	47.32	47.32	45.82	45.82
200	50.79	48.14	49.54	47.52	46.94	47.55
250	52.40	50.58	51.89	49.72	46.83	49.81
300	52.64	52.60	52.38	51.81	47.55	**51.50**
350	52.57	54.06	51.90	52.16	47.86	**52.76**
400	52.18	54.27	52.32	52.69	48.27	**53.23**
450	53.11	**55.60**	52.61	53.41	48.87	**54.56**
500	53.92	**57.01**	52.08	54.04	48.55	**54.21**



Evaluate active learning (3)

Unvertain sampling and stochastic sampling

Table 11: Comparison of RS and US for the MNB uncertainty sampling method using seed size 150 and batch size 10. The number after ± is the standard deviation – averaged over 5 runs.

	Accuracy		F-measure pos		F-measure neg	
#Ex	RS	US	RS	US	RS	US
150	68.10±00.39	68.10±00.39	35.05±06.70	35.05±06.70	26.64±03.21	26.64±03.21
200	73.45±01.01	70.23±00.60	36.50±08.75	33.74±08.32	30.97±02.32	27.67±03.06
250	75.88±01.20	74.25±01.36	37.41±09.15	35.02±07.98	33.43±01.40	31.46±03.55
300	77.53±00.88	76.74±01.61	36.96±10.48	37.91±02.95	33.65±02.75	33.20±04.99
350	78.40±01.06	77.79±01.46	38.63±09.60	40.51±03.10	31.30±06.08	34.47±07.12
400	78.46±00.71	78.25±01.59	38.26±10.33	41.06±02.17	30.52±06.44	34.38±06.30
450	79.21±00.98	79.42±01.27	39.30±06.95	42.08±03.98	31.87±05.94	36.62±05.24
500	79.54±00.70	80.06±01.04	40.15±06.19	44.40±03.63	33.30±05.40	38.21±05.97

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !