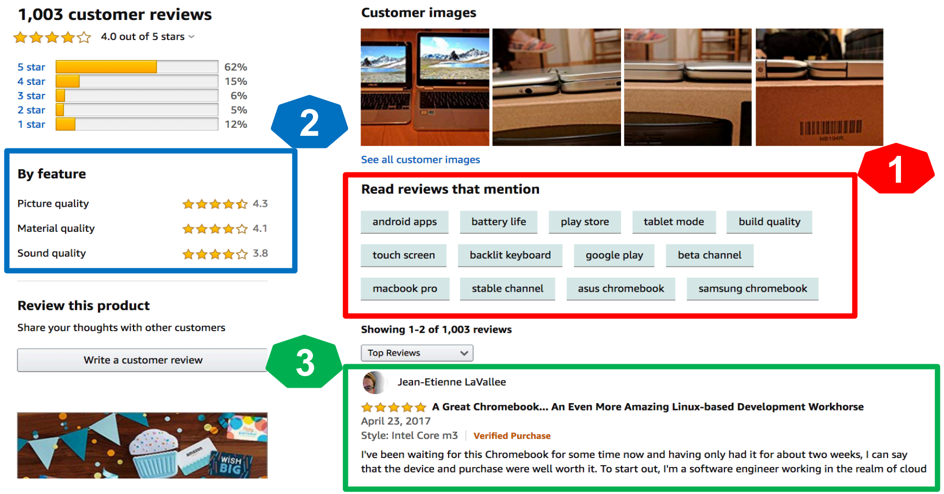
Aspect Term Extraction in Product Reviews

Susanta Ghosh

Outline

- **Project description**
- **Data: Statistics**
- **Related work**
- **Logics and results**
  - **Logic – 1:** Unsupervised approach(Rules based)
  - **Logic – 2:** Supervised approach(Extension of logic-1)
  - **Logic – 3:** CRF
  - **Logic – 4:** Ensemble
- **Future work**

Use case



General Use case

Case 1: Find most important aspects. Aspect based review filtering.

Case 2: Aspect based rating.

Case 3: Combine customer review based on aspect terms.

Other Use case

Case 1: New marketing strategy.
Case 2: Product improvement.

Problem Description

Aspect Term(AT):

- Opinionated expressions.
- Example : “I expected so as it's an Apple product.” In this sentence no AT present.

Description

- It is the task of automatically extract aspect term from product review text.
- For each word of a review, the model should predict if the word is an aspect term or not of the reviewed product.
- It is a **multi-label** classification problem. Multi-label = {**O** : not an aspect, **B** : first word of an aspect (Beginning), **I** : second, third, … word of an aspect (Inside)}
- It is a subtask of aspect-based sentiment analysis.

What need to do?

- **Given:** Product reviews and aspect terms
- **Need to do:** Build an AT extractor model which should predict the word is a class {O, B, I} of the reviewed product.

Problem: Example

Consider the product review and aspect terms:

Review text: The **battery life** is really good and its **size** is reasonable

Aspect terms: Battery life, size.

BIO format(It will helpful for train a classifier or evaluate model):

The	battery	life	is	really	good	and	its	size	is	reasonable
O	B	I	O	O	O	O	O	B	O	O

- ATE is to assign to each word one of the three possible classes:
  - O = not an aspect (Outside)
  - B = first word of an aspect
  - I = second, third, ... word of an aspect (Inside)

Challenges

Consider the Laptop reviews:

Examples:

1. I expected so as it's an **Apple product**.
2. I love India.
3. The **battery life** is really good and its size is reasonable.

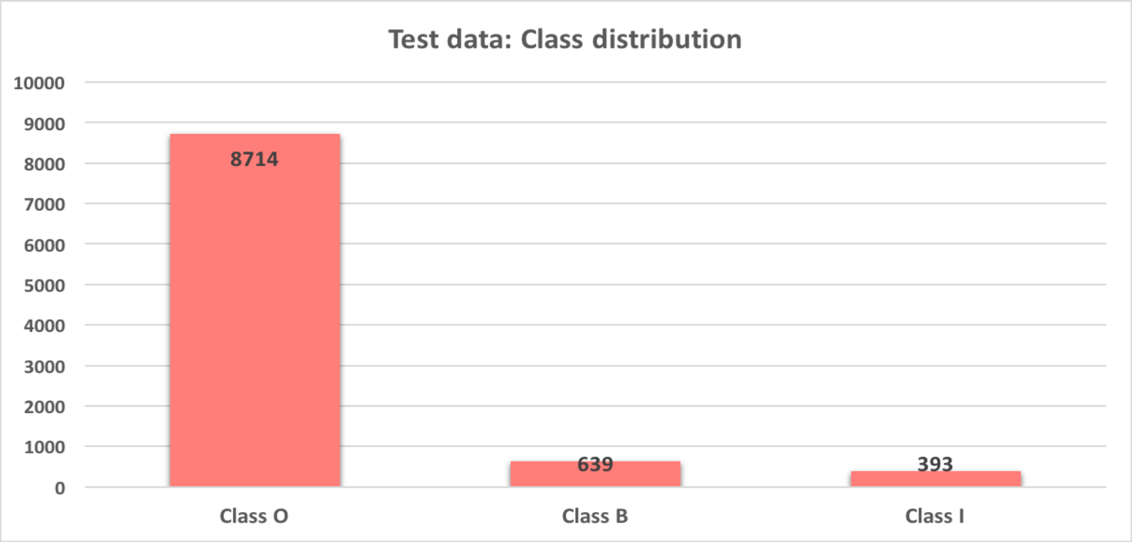
Challenge 1: AT is opinionated expression. See example 1.

Challenge 2: Domain product aspect terms. See example 2.

Challenge 3: Aspect term can be composed of more than one token. See example 3

Data: Statistics

Data type	Number of review	#aspect (single word)	#aspect (multi word)
Training	3045	365	590
Test	800	167	226



Related work

1. **Frequency and Relation based Approaches:**
    - Feature-based Summarization (Hu et al. 2004)
    - A Rule-Based Approach to Aspect Extraction from Product Reviews(my implementation)

2. **Model-based Approaches:**
    - **Supervised Learning:**
        - Naïve Bayes
        - Hidden Markov model(HMM)
            - Opinion Miner [Jin et al. 2009]
        - Conditional Random Field (CRF)
            - CRF (Li et al. 2010)
    - **Topic modeling techniques:**
        - MG-LDA (Titov et al. 2008a)

Outline

- Project description
- Data: Statistics
- Related work
- **Logics and results**
  - **Logic – 1:** Unsupervised approach(Rules based)
  - **Logic – 2:** Supervised approach(Extension of logic-1)
  - **Logic – 3:** CRF
  - **Logic – 4:** Ensemble
- Future work

Logic 1: Unsupervised approach

- I try to solve this problem completely unsupervised way.
- Extract aspect terms using syntactic dependencies of words in a sentence.

Workflow with example

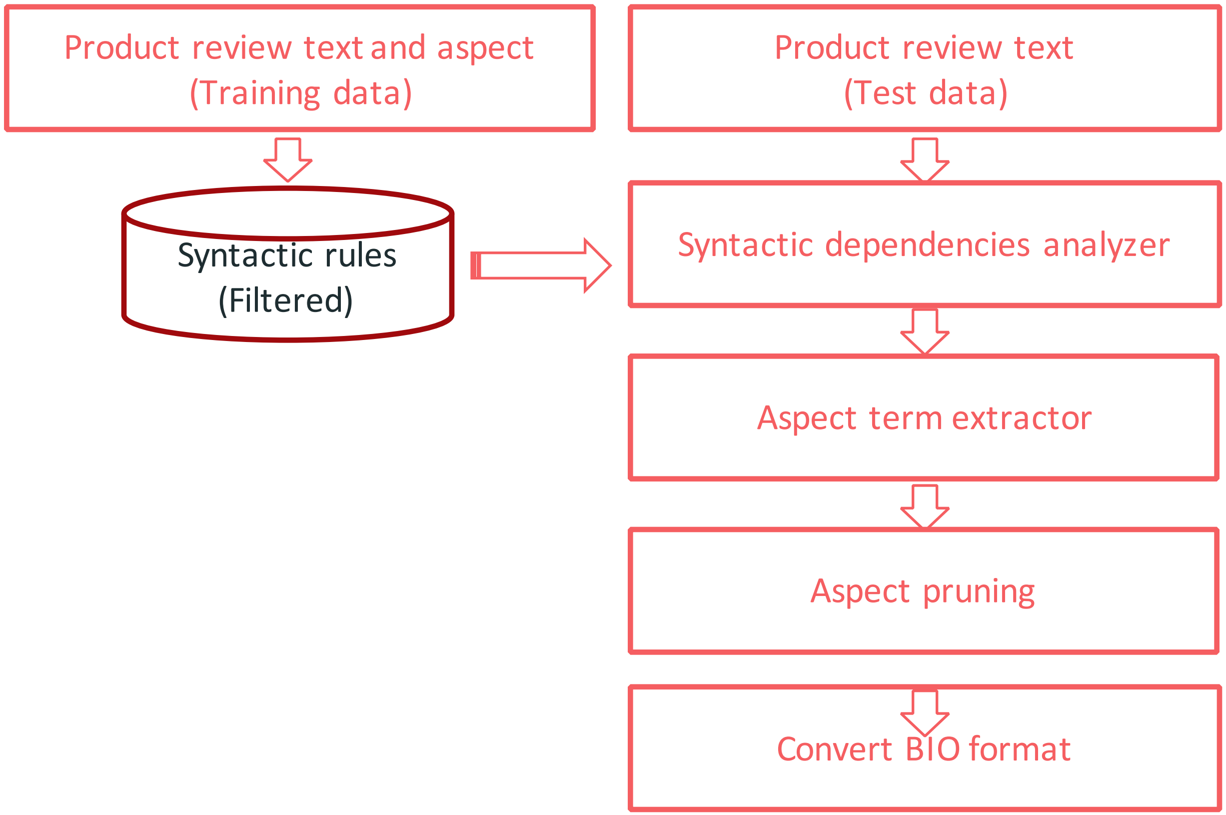
Review text: The battery life is really good and its size is reasonable

Aspect - opinion pair: [(battery, good), (size , reasonable), **(battery, life)**]

Aspect terms : [battery life, size]

Convert to BIO format: [(The, O), (battery, B), (life, I), (is, O), (really, O), (good, O), (and, O), (its, O), (size, B), (is, O), (reasonable, O)]

Logical workflow



Algorithm: Syntactic rules

1. **Nominal subject:** The aspect is the subject of the opinion.

2. **Noun compound modifier:** Noun that serves to modify the head noun.

3. **Adjectival modifier:** The opinion is an adjectival modifier of the aspect.

4. **Direct object:** The direct is the direct object of the opinion.

From training data, I find what syntactic rules and pattern can be useful.

Stanford Parser: In build syntactic rules tool.

Stanford Parser notation:

(('good', 'JJ'), 'nsubj', ('life', 'NN'))
(('reasonable', 'JJ'), 'nsubj', ('size', 'NN'))
(('life', 'NN'), 'compound', ('battery', 'NN'))

From above notations we an create aspect-opinion pair.

This type of several syntactic rules I use to create aspect-opinion pair.

Algorithm: Aspect terms extractor

Consider the product review and aspect terms:

Review text: The battery life is really good and its size is reasonable

Stanford Parser notation	Extracted aspect
(('good', 'JJ'), **'nsubj'**, ('life', 'NN'))	life
(('reasonable', 'JJ'), **'nsubj'**, ('size', 'NN'))	size
(('life', 'NN'), **'compound'**, ('battery', 'NN'))	battery life

Final aspect terms: {battery life, size}

In the above text **“good”** and **“reasonable”** is opinion word and **“battery life”** and **“size”** are aspects.

Algorithm: Aspect pruning

Some cases, redundant aspect could be extracted, let's consider below aspect term list:

Extracted aspect terms: [**battery life**, **life**, size]

Now, you can see “life” aspect already included in “battery life”.

After removing redundant aspect: [battery life, size]

Evaluation

Logic	Class	Precision	Recall	F1-score
Logic-1	O	0.94	0.89	0.91
Logic-1	B	0.23	0.43	0.30
Logic-1	I	0.34	0.19	0.25

- **Logic-1 F1-score(macro avg.): 0.49**

This is a class imbalance problem, as you can see in the data set almost 90% of word belongs to class O.

Class I: recall is very low, that mean this model only 19% correctly find out. Also, precision not good.

Class B: recall is not bad but precision is too low, that means wrong classification is high

Class O: Just because of class imbalance results looks good but actually it is not.

Logic 1: Pro and Cons

Pro:

1. Totally unsupervised
2. Can find low-frequency aspects.

Cons:

1. Not consider domain product aspect(challenge 2).
2. Produce many non-aspects matching with the patterns
3. Class I: recall is too low.

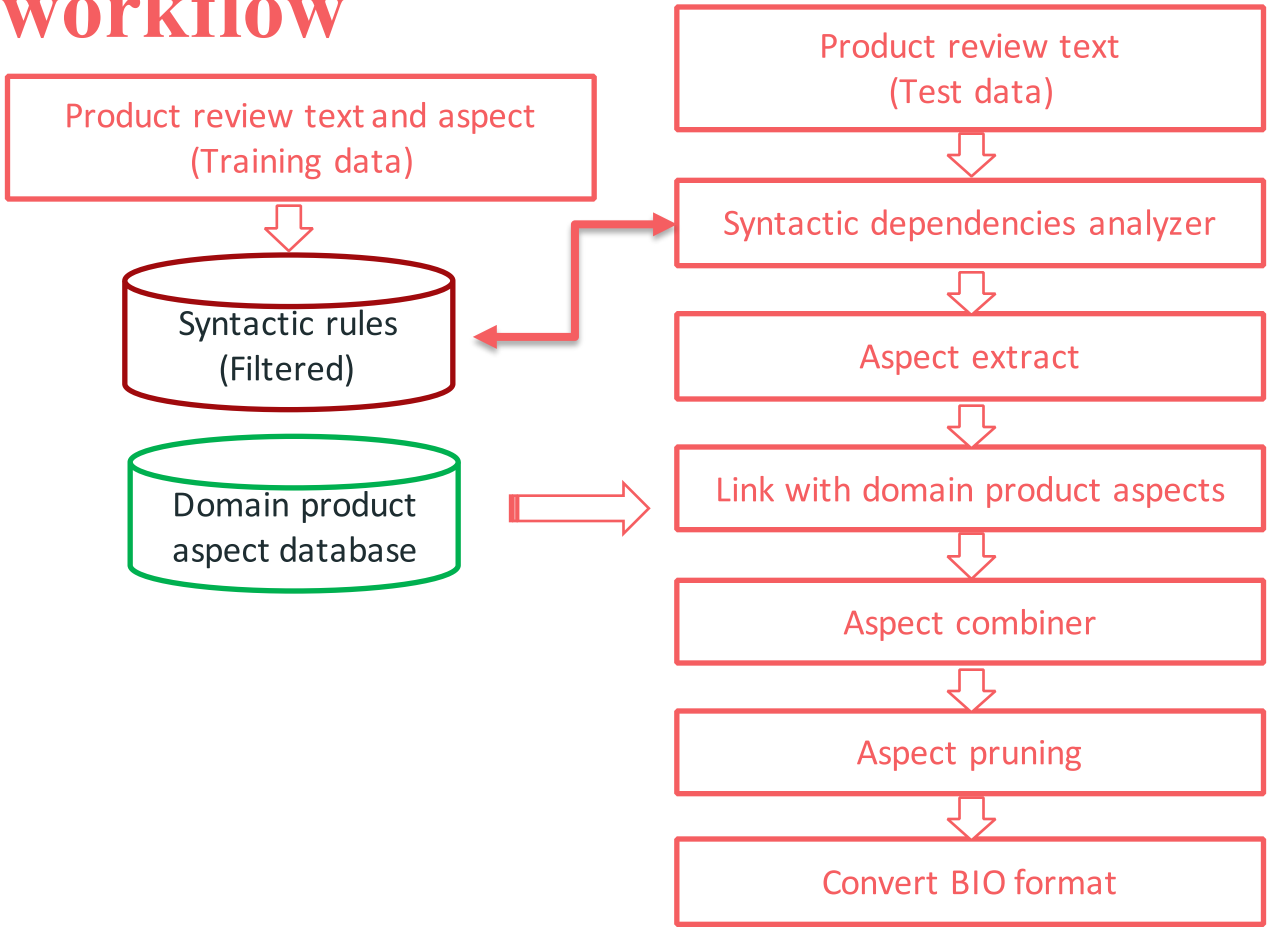
Logic 2: Extension of Logic-1

Goal:

- Remove all the cons of logic-1 and get good results.

Idea:

Logical workflow



Algorithm: Syntactic rules

1. **Nominal subject:** The aspect is the subject of the opinion.
2. **Noun compound modifier:** Noun that serves to modify the head noun.
3. **Adjectival modifier:** The opinion is an adjectival modifier of the aspect.
4. **Direct object:** The target is the direct object of the opinion.

From training data, I find what syntactic rules and pattern can be useful.

Stanford Parser: In build syntactic rules tool.

Stanford Parser notation:

(('good', 'JJ'), 'nsubj', ('life', 'NN'))
(('reasonable', 'JJ'), 'nsubj', ('size', 'NN'))
(('life', 'NN'), 'compound', ('battery', 'NN'))

From above notations we an create aspect-opinion pair.

This type of several syntactic rules we use to create aspect-opinion pair.

Algorithm: Aspect terms extractor

Consider the product review and aspect terms:

Review text: The battery life is really good and its size is reasonable

Stanford Parser notation	Extracted aspect
(('good', 'JJ'), **'nsubj'**, ('life', 'NN'))	life
(('reasonable', 'JJ'), **'nsubj'**, ('size', 'NN'))	size
(('life', 'NN'), **'compound'**, ('battery', 'NN'))	battery life

Final aspect terms: {battery life, size}

In the above text **"good"** and **"reasonable"** is opinion word and **"battery life"** and **"size"** is aspects.

Algorithm: Link with domain aspect

- All aspects got from aspect extractor just filter it.
- Take only those aspects, also present in domain product aspects database.

Algorithm: Aspect combiner

Aspect term can be composed of more than one token so how I deal with?

Syntactic dependency rule :

- (('life', 'NN'), **'compound'**, ('battery', 'NN')) → **battery life**

Sentence POS tag sequence:

1. **Noun followed by Noun:**
   - [('The', 'DT'), **('battery', 'NN'), ('life', 'NN')**, ('is', 'VBZ'),('really', 'RB'),('good', 'JJ')]. → **Battery life**
2. **Noun followed by Cardinal number:**
   - [('I', 'PRP'), ('like', 'VBP'), **('windows', 'NNS'), ('8', 'CD')**]. → **Windows 8**
3. **Noun before Noun**

Algorithm: Aspect pruning

Some cases, redundant aspect could be extract, lets consider below aspect term list:

Extracted aspect terms: [**battery life**, **life**, size]

Now, you can see “life” aspect already included in “battery life”.

After removing redundant aspect: [battery life, size]

Last: Convert to BIO format and evaluate

Evaluation

Logic	Class	Precision	Recall	F1-score
Logic-1	O	0.94	0.89	0.91
Logic-1	B	0.23	0.43	0.30
Logic-1	I	0.34	0.19	0.25
Logic-2	O	0.96	0.99	0.97
Logic-2	B	0.72	0.64	0.68
Logic-2	I	0.79	0.43	0.56

- **Logic-2 F1-score(macro avg.): 0.74**

Class I: Now recall is reasonable and precision is good.

Class B: recall is good and precision also better then logic-1

Evaluation Metrics: SemEval 2014 Task-4

$$Precision = \frac{|ExtractedAspects \bigcap GoldStandardAspects|}{|ExtractedAspects|}$$

$$Recall = \frac{|ExtractedAspects \bigcap GoldStandardAspects|}{|GoldStandardAspects|}$$

$$F - measure = \frac{2 \times Recall \times Precision}{(Recall + Precision)}$$

Results: SemEval 2014 Task-4

No.	Team	F1-score
1	HIS_RD.	74.55
2	DLIREC	73.78
3	DLIREC	70.4
4	NRC-Can.	68.56
5	UNITOR	67.95
6	XRCE	67.24
7	SAP_RI	66.6
8	IITP	66.55
9	UNITOR	66.08
10	SeemGo	65.99
11	ECNU	65.88
12	SNAP	62.4

No.	Team	F1-score
13	Logic-2	62.01
14	DMIS	60.59
15	UWB	60.39
16	JU_CSE.	59.37
17	lsis_lif	56.97
17	USF	52.58
19	Blinov	52.07
20	UFAL	48.98
21	UBham	47.49
22	UBham	47.26
23	SINAI	45.28
24	EBDG	41.52

Logic 3: Conditional Random Fields(CRF)

CRF:

- A type of Discriminative classifier, and as such, they model the decision boundary between the different classes.
- We'll first need to decide on a set of feature functions ***fi***

Feature Functions in a CRF:

In a CRF, each **feature function** is a function that takes in as input:

- a sentence s
- the position i of a word in the sentence
- the label l_i of the current word
- the label l_{i-1} of the previous word

and outputs a real-valued number (though the numbers are often just either 0 or 1).

Logic 3: CRF

Features to Probabilities:

- Assign each feature function ***fj*** a weight ***λj.***
- Given a sentence **s**, we can now score a labeling ***l*** of ***s*** by adding up the weighted features over all words in the sentence:

$$score(l|s) = \sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s, i, l_i, l_{i-1})$$

Finally, we can transform these scores into probabilities $p(l|s)$ between 0 and 1 by exponentiating and normalizing:

$$p(l|s) = \frac{exp[score(l|s)]}{\sum_{l'} exp[score(l'|s)]} = \frac{exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s,i,l_i,l_{i-1})]}{\sum_{l'} exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s,i,l'_i,l'_{i-1})]}$$

Logic 3: CRF

Learning Weights: Gradient descent

$\lambda_i = \lambda_i + \alpha[\sum_{j=1}^{m} f_i(s,j,l_j,l_{j-1}) - \sum_{l'} p(l'|s) \sum_{j=1}^{m} f_i(s,j,l'_j,l'_{j-1})]$ where α is some learning rate.

- First term in the gradient is the contribution of feature *fi* under the true label
- Second term in the gradient is the expected contribution of feature *fi* under the current model.

Features

Features list:

- Word
- POS tag
- Previous word
- Previous word POS tag
- Previous previous word
- Previous previous word POS tag
- Next word
- Next word POS tag
- Next next word
- Next next word POS tag
- Is first
- Is last
- Is capitalized
- Is all caps
- Is all lower
- prefix-1
- prefix-2
- prefix-3
- suffix-1
- suffix-2
- suffix-3
- Has hyphen
- Is numeric
- Capitals inside

Evaluation

Logic	Class	Precision	Recall	F1-score
Logic-1	O	0.94	0.89	0.91
Logic-1	B	0.23	0.43	0.30
Logic-1	I	0.34	0.19	0.25
Logic-2	O	0.96	0.99	0.97
Logic-2	B	0.72	0.64	0.68
Logic-2	I	0.79	0.43	0.56
Logic-3	O	0.96	0.99	0.97
Logic-3	B	0.79	0.61	0.69
Logic-3	I	0.84	0.50	0.62

- **Logic-3 F1-score(macro avg.): 0.76**

Class I: Now recall and precision is good.

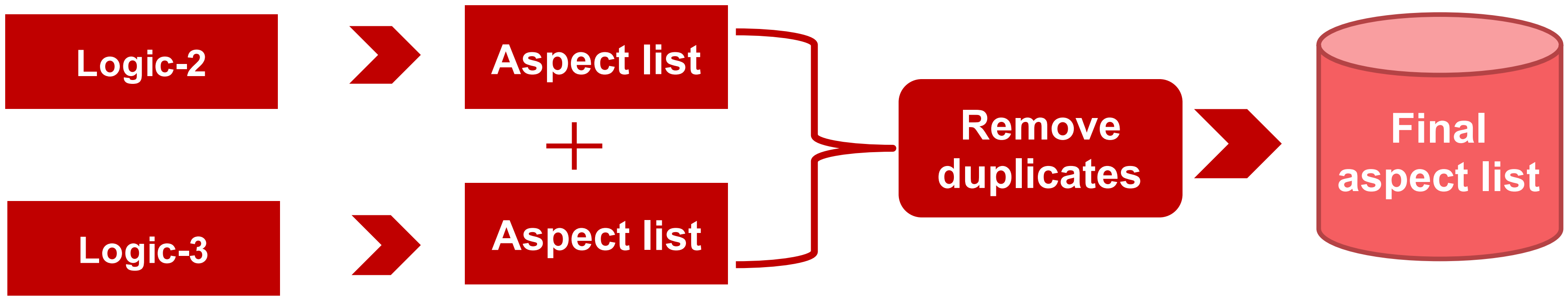
Class B: recall is good and precision also better then logic-2

Results: SemEval 2014 Task-4

No.	Team	F1-score
1	HIS_RD.	74.55
2	DLIREC	73.78
3	DLIREC	70.4
4	NRC-Can.	68.56
5	UNITOR	67.95
6	XRCE	67.24
7	SAP_RI	66.6
8	IITP	66.55
9	UNITOR	66.08
10	SeemGo	65.99
11	ECNU	65.88
12	SNAP	62.4

No.	Team	F1-score
13	Logic-2	62.01
14	Logic-3(crf)	61.51
15	DMIS	60.59
16	UWB	60.39
17	JU_CSE.	59.37
18	lsis_lif	56.97
19	USF	52.58
20	Blinov	52.07
21	UFAL	48.98
22	UBham	47.49
23	UBham	47.26
24	SINAI	45.28

Logic 4: Ensemble logic-2 and logic-3



Results: SemEval 2014 Task-4

Surprise

No.	Team	F1-score
1	Logic-4	75.64
2	HIS_RD.	74.55
3	DLIREC	73.78
4	DLIREC	70.4
5	NRC-Can.	68.56
6	UNITOR	67.95
7	XRCE	67.24
8	SAP_RI	66.6
9	IITP	66.55
10	UNITOR	66.08
11	SeemGo	65.99
12	ECNU	65.88
13	SNAP	62.4

No.	Team	F1-score
14	Logic-2	62.01
15	Logic-3(crf)	61.51
16	DMIS	60.59
17	UWB	60.39
18	JU_CSE.	59.37
19	lsis_lif	56.97
20	USF	52.58
21	Blinov	52.07
22	UFAL	48.98
23	UBham	47.49
24	UBham	47.26
25	SINAI	45.28

Outline

- **Project description**
- **Data: Statistics**
- **Related work**
- **Logics and results**
    - **Logic – 1:** Unsupervised approach(Rules based)
    - **Logic – 2:** Supervised approach(Extension of logic-1)
    - **Logic – 3:** CRF
    - **Logic – 4:** Ensemble
- **Future work**

Features work: Rules based approach

1. **Syntactical relations:**
   - Can be improved further on by looking up more combined patterns and I have a gut feeling there are more combined patterns out there.
2. **Aspect term with multiple token:**
   - Yes, this part also we can improve by adding more POS tag pattern.
3. **Increase domain product aspects :**
   - WordNet.

Features work: CRF

1. More features:

Word Feature:
- Word token
- Word lemma
- Word part of speech
- Previous word token, lemma, part of speech
- Next word token, lemma, part of speech
- Negation word appears in previous 4 words
- Is superlative degree
- Is comparative degree

Dictionary Feature
- WordNet Synonym
- WordNet Antonym
- SentiWordNet Prior Polarity

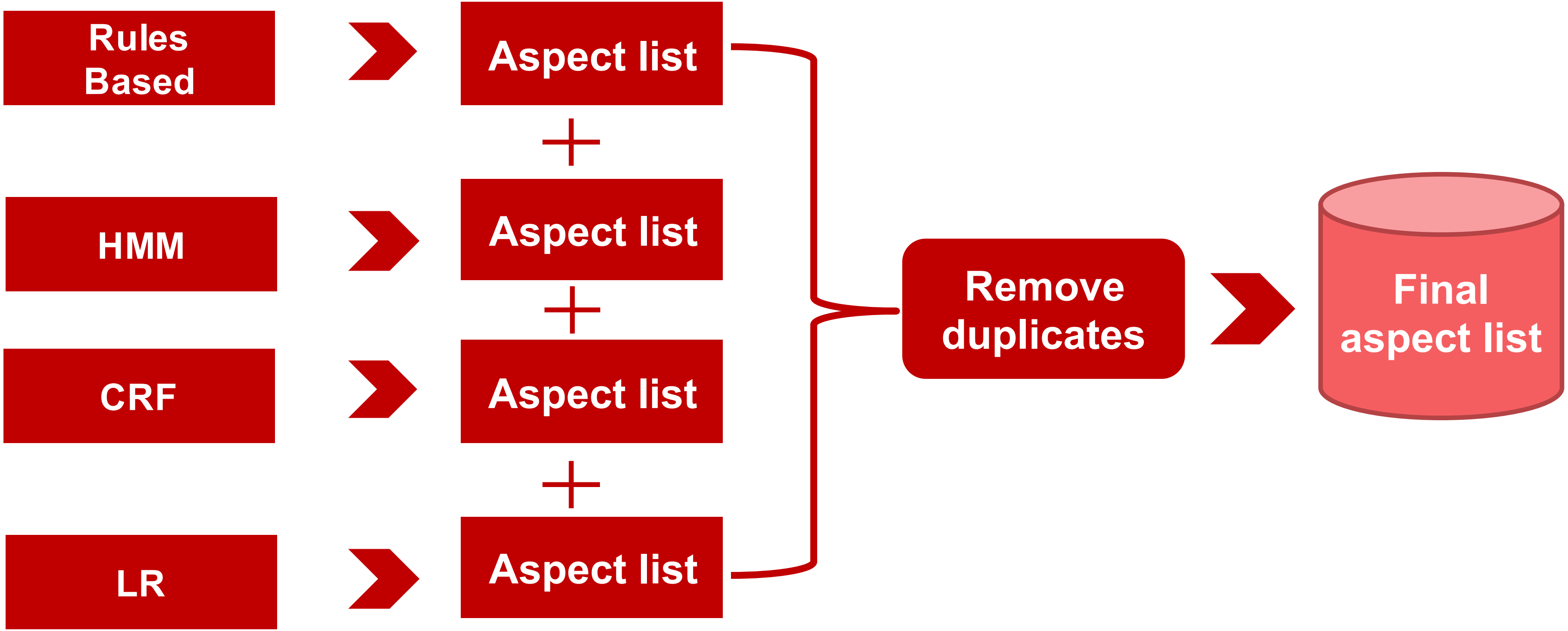
Sentence Feature
- Num of positive words in SentiWordNet
- Num of negative words in SentiWordNet
- Num of Negation word

Syntactic Features:
- Parent word
- Parent SentiWordnet Prior Polarity
- In subject
- In copular
- In object

Edge Feature
- Conjunction word
- Syntactic relationship

- CRF (Li et al. 2010)

Features Workflow



- Bagging approach for supervised classifiers

Thank You