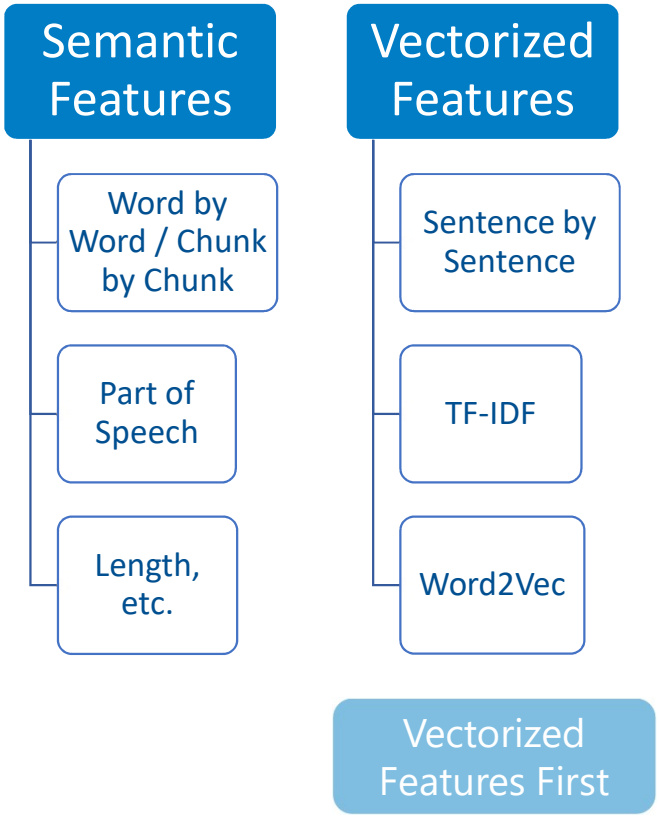
Aspect Term Extraction

Wei Liu

Walking through in mind

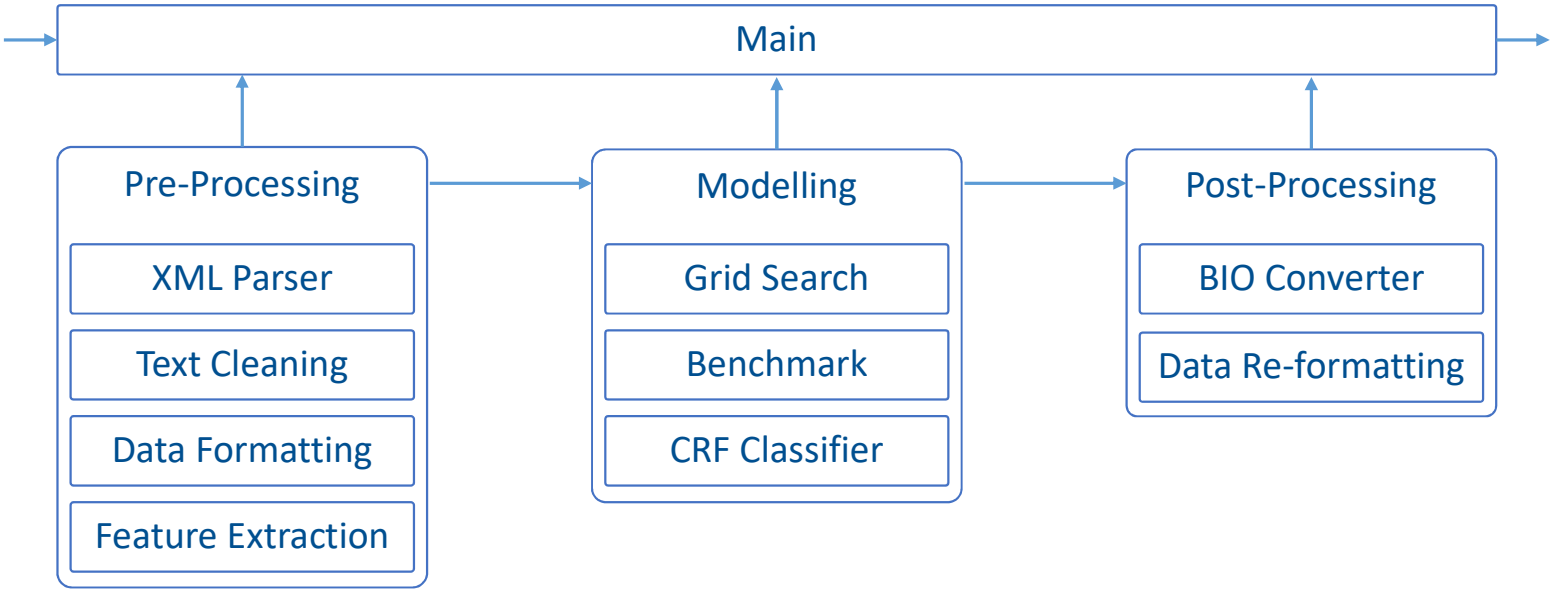
- Text Classification Problem
- Review contain zero to multiple terms
- Term contain one to multiple words
- What features to use?
- What labels to use?



Contents:

- Module Structure
- Data Pre-Processing:
  - XML Parsing
  - Split Review
  - Cleaning
  - BIO-encoding
  - From BIO to Label
  - Vectorize data
- Algorithm Evaluation:
  - Benchmarks
  - Metrics for Vectorized features
  - Some results
  - Results after Post-Processing
- Semantic Features and CRF:
  - Conditional Random Field (CRF)
  - Semantic Features for CRF
  - Result
  - Experiment more features for CRF
  - What CRF Learned

Module Structure



Data Pre-Processing: XML Parsing

```
<sentence id="2339">
   <text>I charge it at night and skip taking the cord with me because of the good battery life.</text>
   <aspectTerms>
      <aspectTerm term="cord" polarity="neutral" from="41" to="45"/>
      <aspectTerm term="battery life" polarity="positive" from="74" to="86"/>
   </aspectTerms>
</sentence>
```

Review	Terms
I charge it at night and skip taking the cord with me because of the good battery life	Cord, battery life

Data Pre-Processing: Split Review

Review	Terms
it is of high quality, has a killer GUI, is extremely stable, is highly expandable, is bundled with lots of very good applications, is easy to use, and is absolutely gorgeous.	quality, GUI, applications, use

Review	Terms
it is of high quality	quality
has a killer GUI	GUI
…	…

Data Pre-Processing: Cleaning

- General Cleaning:
  - Has/had -> have
  - Is/are/was/were -> be
  - ...../,,,,, -> ./,
  - E-mail/E – mail -> email
  - (optional) Stop words
  - (optional) Stems
  - And more…

Data Pre-Processing: BIO-encoding

I	charge	it	at	night	and	skip	taking	the	cord
O	O	O	O	O	O	O	O	BB	B
with	me	because	the	battery	life	is	good		
EB	O	O	BB	B	I	EI	O		

- Extended BIO-encoding:
  - BB: token before B
  - EB: token after B
  - EI: token after I

Data Pre-Processing: BIO to Label

- Convolution with 5 tokens:

nan	nan	because	the	battery
nan	because	the	battery	life
because	the	battery	life	is
the	battery	life	is	good
battery	life	is	good	nan
life	is	good	nan	nan

Data Pre-Processing: BIO to Label (cont'd)

- Map convolutions to label:
  - Tag of central token -> Label

Review Conv					Tag
nan	nan	because	the	battery	O
nan	because	the	battery	life	BB
because	the	battery	life	is	B
the	battery	life	is	good	I
battery	life	is	good	nan	EI
life	is	good	nan	nan	O

Data Pre-Processing: Vectorize data

- Count Vectorizer:

nan	because	the	battery	life	is	good	Tag
2	1	1	1	0	0	0	O
1	1	1	1	1	0	0	BB
0	1	1	1	1	1	0	B
0	0	1	1	1	1	1	I
1	0	0	1	1	1	1	EI
2	0	0	0	1	1	1	O

Data Pre-Processing: Vectorize data (cont'd)

- TF-IDF Vectorizer:
  - With Euclidean (L2) normalization

nan	because	the	battery	life	is	good	Tag
0.693	0.477	0.409	0.324	0	0	0	O
0.454	0.529	0.454	0.392	0.392	0	0	BB
0	0.529	0.454	0.392	0.392	0.454	0	B
0	0	0.454	0.392	0.392	0.454	0.529	I
0.454	0	0	0.392	0.392	0.454	0.529	EI
0.693	0	0	0	0.353	0.409	0.477	O

Algorithm Evaluation

It’s time to generate some heat !

Algorithm Evaluation: Benchmarks

Algorithm	F1 Score
Linear SVC	0.845
Linear SVC - SGD	0.835
Bernoulli NB	0.835
Multinomial NB	0.811
Random Forest	0.808 (0.833*)
K-NN	0.790
X G Boost	0.787 (0.842*)

* Tree Algorithms show improvements via SMOTE over-sampling.

Algorithm Evaluation: Metrics

	Precision	Recall	F1	Support
B	0.46	0.32	0.38	650
I	0.55	0.40	0.46	409
BB	0.41	0.34	0.37	548
EB	0.30	0.44	0.36	260
EI	0.26	0.33	0.29	169
O	0.94	0.97	0.96	8381
Avg. / Total	0.84	0.85	0.85	10417

Algorithm Evaluation: Some Results

Review					Truth	Prediction
not	enjoy	the	new	windows	O	O
enjoy	the	new	windows	8	BB	BB
the	new	windows	8	and	B	B
new	windows	8	and	touchscreen	I	B
windows	8	and	touchscreen	functions	EI	O
8	and	touchscreen	functions	nan	B	EB
and	touchscreen	functions	nan	nan	I	I

Algorithm Evaluation: Post-Processing

- Post-Processing is possible:
  - Revert convolutions to sentence
  - Convert BB, EB, EI back to B,I
  - Validation and Majority Vote

Review					Truth	Prediction
slim	but	the	features	make	BB	BB
but	the	features	make	up	B	BB -> B
the	features	make	up	for	EB	EB

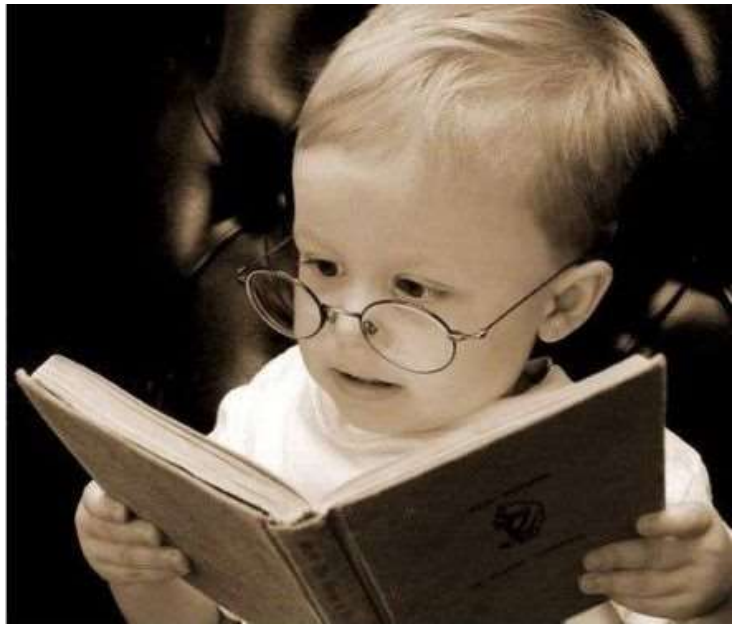
Algorithm Evaluation: Results after PP

	Precision	Recall	F1	Support
B	0.46 -> 0.40	0.32 -> 0.48	0.38 -> 0.43	650 -> 708
I	0.55 -> 0.38	0.40 -> 0.48	0.46 -> 0.43	409 -> 412
O	0.94 -> 0.96	0.97 -> 0.94	0.96 -> 0.95	8381 -> 9297
Avg. / Total	0.84 -> 0.95	0.85 -> 0.89	0.85 -> 0.89	10417

- Mini Conclusion:
  - Post-Processing steps need to be tweaked
  - Fast Fix: Only keep post-processing steps relating to tag B
  - 3 token convolution is better than 5 token ones
  - Remove stop words or nans give negative effect

Semantic Features and CRF

It’s time to read some literatures.



Semantic Features and CRF: CRF

- Conditional Random Field:
  - A very new thing for me.
  - One of the state-of-art in NLP before Deep Learning
  - Very Interesting API style (python-crfsuite)
  - Easy to learn and get hands on
  - Sequence modeling, taking context into account
  - linear chain CRF perform well in predict sequences of labels (super!)

Semantic Features and CRF: features

- Part-of-Speech Tags: NLTK.pos_tag
- Local Context: surrounding words
  - 3 token window / 5 token window
- Prefix and Suffix:
  - length up to 3 chars
- Root word: NLTK.stem
- Stop word:
  - True if token is stop word else False
- Length:
  - tried both number and Boolean (length >= 5).

Semantic Features and CRF: Results

	Precision	Recall	F1	Support
B	0.86	0.60	0.71	652
BB	0.80	0.56	0.66	547
EB	0.82	0.58	0.68	266
EI	0.79	0.50	0.61	171
I	0.88	0.53	0.66	414
O	0.91	0.99	0.95	8367
Avg. / Total	0.89	0.90	0.89	10417

Semantic Features and CRF: More Features

- Stop word:
  - True if token is stop word else False
- TF-IDF score:
  - Both un-normalized and normalized.
- Frequent Aspect Term:
  - True if token is top 100 most frequent aspect terms in training data.

Semantic Features and CRF: More Features

	Precision	Recall	F1	Support
B	0.86 -> 0.88	0.60 -> 0.67	0.71 -> 0.76	652
BB	0.80 -> 0.82	0.56 -> 0.63	0.66 -> 0.71	547
EB	0.82 -> 0.79	0.58 -> 0.62	0.68 -> 0.70	266
EI	0.79 -> 0.83	0.50 -> 0.64	0.61 -> 0.72	171
I	0.88 -> 0.90	0.53 -> 0.62	0.66 -> 0.74	414
O	0.91 -> 0.93	0.99 -> 0.99	0.95 -> 0.96	8367
Avg. / Total	0.89 -> 0.91	0.90 -> 0.92	0.89 -> 0.91	10417

What CRF Learned: transitions

Top Positives
BB -> B
I -> EI
B -> EB
O -> O
B -> I
I -> I
O -> BB
EI -> O
EB -> O
EI -> B

Top Negatives
O -> I
EI -> I
EB -> B
B -> O
I -> O
B -> BB
I -> EB
I -> BB
BB -> I
BB -> EB

What CRF Learned: state features

Top Positives	
I	+1:word.lower=resolution
I	+1:word.lower=hook
B	word[:3]=saf
B	word.stem=surf
EI	-1:word.lower=replacing
B	word[:2]=iw
I	+1:word.lower=feature
B	word[:2]=hd
I	word.stem=technician
EB	-1:word.stem=cord

Top Negatives	
B	word.isstop=True
BB	EOS
O	-1:word.lower=plug
B	word.stem=comput
EB	FRQ_TERM
O	+1:word.lower=burn
O	word[:3]=dri
O	word.lower=designed
O	+1:word.lower=although
I	BOS

Conclusion and Future work

- SVM with TF-IDF has difficult to capture positioning information
- Convolution method works, but not great with it's own (Deep Learning for the rescue).
- Extending BIO tagging method is a good idea
- CRF is awesome! (and Hotel in Switzerland is Expensive!!)
- Learn and use more CRF (for scenario that DL is not available)
- Play with more semantic features (like Chunk information)
- Use LSTM + Word2Vec + semantic features to test the ceiling
- Play with another test case

Thank you! Questions?

Wei Liu