# Generalizing Dependency Features for Opinion Mining


Mahesh Joshi[1] and Carolyn Rosé[1,2]

[1] Language Technologies Institute
[2] Human-Computer Interaction Institute
Carnegie Mellon University



ACL-IJCNLP 2009 Short Papers Track

# One Slide Summary

- **Goal**: utilize structure in language to improve “opinion vs. not-opinion” classification in product reviews
- **Key idea**: “Back off” the head word in a dependency relation to it’s part-of-speech tag, use these “back-off” features
  - amod(camera,great) => amod(NN,great)
- **Result**: It works!
  - Improvement in accuracy from 0.652 to 0.679
    - yes, it’s significant!

# One Slide Summary

- **Goal**: utilize structure in language to improve “opinion vs. not-opinion” classification in product reviews

- **Key idea**: “Back off” the head word in a dependency relation to it’s part-of-speech tag, use these “back-off” features
  - amod(camera,great) => amod(NN,great)

- **Result**: It works!
  - Improvement in accuracy from 0.652 to 0.679
    - yes, it’s significant!



# Task Description

- Opinion mining in product reviews
- Given a sentence from a product review
  - Predict whether or not it is an *opinion sentence*
- *Opinion Sentence*: “If a sentence contains one or more **product features** and one or more **opinion words**, then the sentence is called an opinion sentence.” [Hu and Liu; SIGKDD 2004]

# Examples

- Opinion sentences
  - *It is very **light** weight and has **good** signal strength.*
  - *Player works and looks **great** – if you can get the dvd's to play.*

- Non-opinion sentences
  - *Had it for a week.*
  - *If this doesnt bring back the picture, try pressing this button without playing a dvd.*

# Research Question

How can we better utilize structure in language for opinion classification?

# Dependency Relations

# Dependency Relations

*it is a fantastic camera*
*and well worth the price.*

# Dependency Relations

*it is a fantastic camera and well worth the price.*

(Stanford parser output)

==>

nsubj(camera-5,it-1)
cop(camera-5,is-2)
det(camera-5,a-3)
amod(camera-5,fantastic-4)
advmod(worth-8,well-7)
det(price-10,the-9)
amod(price-10,worth-8)
conj_and(camera-5,price-10
)

# Dependency Relations

*it is a fantastic camera and well worth the price.*

(Stanford parser output)

Relation

nsubj(camera-5,it-1)
cop(camera-5,is-2)
det(camera-5,a-3)
amod(camera-5,fantastic-4)
advmod(worth-8,well-7)
det(price-10,the-9)
amod(price-10,worth-8)
conj_and(camera-5,price-10
)

# Dependency Relations

*it is a fantastic camera and well worth the price.*

(Stanford parser output)

==>

nsubj(camera-5,it-1)
cop(camera-5,is-2)
det(camera-5,a-3)
amod(camera-5,fantastic-4)
advmod(worth-8,well-7)
det(price-10,the-9)
amod(price-10,worth-8)
conj_and(camera-5,price-10
)

# Dependency Relations

*it is a fantastic camera and well worth the price.*

(Stanford parser output)

==>

Head Word

nsubj(camera–5,it–1)
cop(camera–5,is–2)
det(camera–5,a–3)
amod(camera–5,fantastic–4)
advmod(worth–8,well–7)
det(price–10,the–9)
amod(price–10,worth–8)
conj_and(camera–5,price–10
)

# Dependency Relations

*it is a fantastic camera and well worth the price.*

(Stanford parser output)

==>

nsubj(camera-5,it-1)
cop(camera-5,is-2)
det(camera-5,a-3)
amod(camera-5,fantastic-4)
advmod(worth-8,well-7)
det(price-10,the-9)
amod(price-10,worth-8)
conj_and(camera-5,price-10)

# Dependency Relations

*it is a fantastic camera and well worth the price.*

(Stanford parser output)

==>

nsubj(camera-5,it-1)
cop(camera-5,is-2)
det(camera-5,a-3)
amod(camera-5,fantastic-4)
advmod(worth-8,well-7)
det(price-10,the-9)
amod(price-10,worth-8)
conj_and(camera-5,price-10)

Modifier Word

# Dependency Relations

# Dependency Relations

- amod(camera-5,fantastic-4)
  - adjectival modifier relationship between camera (head, noun) and fantastic (modifier, adjective)

# Dependency Relations

- amod(camera-5,fantastic-4)
  - adjectival modifier relationship between camera (head, noun) and fantastic (modifier, adjective)

- advmod(worth-8,well-7)
  - adverbial modifier relationship between worth (head, adjective) and well (modifier, adverb)

# Previous Work

# Previous Work

- Success in using dependency relations: 
  - [Gamon; COLING 2004], [Matsumoto et al.; PAKDD 2005]
    - Different task: predicting customer satisfaction ratings, and polarity (positive / negative) for movie reviews respectively
  - [Wilson et al.; AAAI 2004]
    - Different task: predicting *strength* of subjective language
  - Use full set of dependency relations

# Previous Work

- Success in using dependency relations: 
  - [Gamon; COLING 2004], [Matsumoto et al.; PAKDD 2005]
    - Different task: predicting customer satisfaction ratings, and polarity (positive / negative) for movie reviews respectively
  - [Wilson et al.; AAAI 2004]
    - Different task: predicting *strength* of subjective language
  - Use full set of dependency relations
- We propose multiple generalization approaches
  - one of our approaches was used by Gamon as well as Wilson et al.

# Previous Work

# Previous Work

- Dependency relations not found to be useful: 
  - [Dave et al.; WWW 2003], [Ng et al.; ACL 2006]
    - Different task: polarity prediction in product reviews and movie reviews respectively
  - Both use a subset of dependency relations
    - manually chosen grammatical relations

# Previous Work

- Dependency relations not found to be useful: 
  - [Dave et al.; WWW 2003], [Ng et al.; ACL 2006]
    - Different task: polarity prediction in product reviews and movie reviews respectively
  - Both use a subset of dependency relations
    - manually chosen grammatical relations

- We use the full set of dependency relations

# Recent Work - not in paper

# Recent Work - not in paper

- Transformation of dependency relations
  - [Greene and Resnik; NAACL 2009]
  - Given dependency relations of form: relation(head_word,modifier_word)
    - Create features of form: head_word-relation & relation-modifier_word

# Recent Work - not in paper

- Transformation of dependency relations
  - [Greene and Resnik; NAACL 2009]
  - Given dependency relations of form: relation(head_word,modifier_word)
    - Create features of form: head_word-relation & relation-modifier_word

# Recent Work - not in paper

- Transformation of dependency relations
  - [Greene and Resnik; NAACL 2009]
  - Given dependency relations of form: relation(head_word,modifier_word)
    - Create features of form: head_word-relation & relation-modifier_word
- Dependency relations chosen if they contain “domain relevant” verbs and/or nouns
- Most closely related to our approach

# Lesson from Past Work

# Lesson from Past Work

- Using full set of dependency relations works better

# Lesson from Past Work

- Using full set of dependency relations works better
- However, there might be overfitting in scarce data scenarios
  - common to NLP where annotated data is expensive!
- Need features that can generalize well

# Motivating Our Approach

- Consider two opinion sentences:

*This is a great camera!*

*Despite its few negligible flaws, this great mp3 player won my vote.*

# Motivating Our Approach

- Both have the dependency relation **amod** with different pair of words participating:

amod(camera,great)

amod(player,great)

- We can see the structural similarity of these features

- A machine learning algorithm can't!

# Motivating Our Approach

- Both have the dependency relation **amod** with different pair of words participating:

amod(camera,great)

amod(player,great)

- We can see the structural similarity of these features

- A machine learning algorithm can't!

# So Lets "Back Off!"

- "Back off" the head word to its part-of-speech tag

  amod(camera, great) => amod(NN, great)

  amod(player, great) => amod(NN, great)

- Now the algorithm can see that these are similar features (we made them identical)

# Advantages of Backing Off

# Advantages of Backing Off

- Stronger evidence of association of a generalized feature with the *opinion* category

# Advantages of Backing Off

- Stronger evidence of association of a generalized feature with the *opinion* category

- New test sentence: "*this is a great phone*"
  - amod(phone,great): may not be useful because we might have never seen it!
  - amod(NN,great): still valid

# Backing Off / Generalizing

# Backing Off / Generalizing

- Composite Back-off Features
  - Head word => its part-of-speech tag
    - amod(camera,great) => amod(NN,great)

# Backing Off / Generalizing

- Composite Back-off Features
  - Head word => its part-of-speech tag
    - amod(camera,great) => amod(NN,great)

# Backing Off / Generalizing

- Composite Back-off Features
  - Head word => its part-of-speech tag
    - amod(camera,great) => amod(NN,great)
  - Modifier word => its part-of-speech tag
    - amod(camera,great) => amod(camera,JJ)

# Backing Off / Generalizing

- Composite Back-off Features
  - Head word => its part-of-speech tag
    - amod(camera,great) => amod(NN,great)
  - Modifier word => its part-of-speech tag
    - amod(camera,great) => amod(camera,JJ)

# Backing Off / Generalizing

- Composite Back-off Features
  - Head word => its part-of-speech tag
    - amod(camera,great) => amod(NN,great)
  - Modifier word => its part-of-speech tag
    - amod(camera,great) => amod(camera,JJ)
- Full Back-off Features
  - Head word => its part-of-speech tag; Also modifier word => its part-of-speech tag
    - amod(camera,great) => amod(NN,JJ)
    - used by [Gamon; COLING 2004] and [Wilson et al.; AAAI 2004]

# Backing Off / Generalizing

- Composite Back-off Features
  - Head word => its part-of-speech tag
    - amod(camera,great) => amod(NN,great)
  - Modifier word => its part-of-speech tag
    - amod(camera,great) => amod(camera,JJ)
- Full Back-off Features
  - Head word => its part-of-speech tag; Also modifier word => its part-of-speech tag
    - amod(camera,great) => amod(NN,JJ)
    - used by [Gamon; COLING 2004] and [Wilson et al.; AAAI 2004]

# Backing Off / Generalizing

- Composite Back-off Features
  - Head word => its part-of-speech tag
    - amod(camera,great) => amod(NN,great)
  - Modifier word => its part-of-speech tag
    - amod(camera,great) => amod(camera,JJ)
- Full Back-off Features
  - Head word => its part-of-speech tag; Also modifier word => its part-of-speech tag
    - amod(camera,great) => amod(NN,JJ)
    - used by [Gamon; COLING 2004] and [Wilson et al.; AAAI 2004]

# Experiments

# Experiments

- 2,200 sentences (Randomly sampled from Amazon.com and Cnet.com reviews for 11 products, 200 per product)
  - Subset of dataset released by [Hu and Liu; SIGKDD 2004]

# Experiments

- 2,200 sentences (Randomly sampled from Amazon.com and Cnet.com reviews for 11 products, 200 per product)
  - Subset of dataset released by [Hu and Liu; SIGKDD 2004]

- Support Vector Machine classifier, linear kernel

# Experiments

- 2,200 sentences (Randomly sampled from Amazon.com and Cnet.com reviews for 11 products, 200 per product)
  - Subset of dataset released by [Hu and Liu; SIGKDD 2004]
- Support Vector Machine classifier, linear kernel
- Chi-squared feature selection
  - used significant features at $\alpha = 0.05$

# Evaluation

- 11-fold cross-validation, sentences for each product used in test fold once
- Reporting average accuracy, Cohen’s kappa

# Baselines

# Baselines

- Several standard feature sets
  - ngrams (n = 1, 2, 3)
  - Part-of-Speech ngrams (n = 2, 3)
  - Dependency relations (no back-off)

# Baselines

- Several standard feature sets
  - ngrams (n = 1, 2, 3)
  - Part-of-Speech ngrams (n = 2, 3)
  - Dependency relations (no back-off)

- Back-off ngrams (n = 2, 3) similar to [McDonald et al.; ACL 2007]
  - back off words in an ngram to POS tags
  - create features using all possible ($2^n$-1) back-off combinations

# Results

| Feature Set | Accuracy | Kappa |
|---|---|---|
| Unigrams | 0.652 (±0.048) | 0.295 (±0.049) |
| Uni.+Bigrams | 0.657 (±0.066) | 0.304 (±0.089) |
| Uni.+Trigrams | 0.655 (±0.062) | 0.306 (±0.077) |
| Uni.+Back-off Bigrams | 0.650 (±0.056) | 0.299 (±0.079) |
| Uni.+Back-off Trigrams | 0.647 (±0.051) | 0.287 (±0.075) |
| Uni.+POS Bigrams | 0.676 (±0.057) | **0.349 (±0.083)** |
| Uni.+POS Trigrams | 0.661 (±0.050) | 0.317 (±0.064) |
| Uni.+Dep. Lex | 0.639 (±0.055) | 0.268 (±0.079) |
| Uni.+Dep. Head-Back-off | **0.679 (±0.063)** | **0.351 (±0.097)** |
| Uni.+Dep. Mod-Back-off | 0.657 (±0.056) | 0.308 (±0.063) |
| Uni.+Dep. Head-Mod-Back-off | 0.670 (±0.046) | 0.336 (±0.065) |

# Results

| Feature Set | Accuracy | Kappa |
|---|---|---|
| Unigrams | 0.652 (±0.048) | 0.295 (±0.049) |
| Uni.+Bigrams | 0.657 (±0.066) | 0.304 (±0.089) |
| Uni.+Trigrams | 0.655 (±0.062) | 0.306 (±0.077) |
| Uni.+Back-off Bigrams | 0.650 (±0.056) | 0.299 (±0.079) |
| Uni.+Back-off Trigrams | 0.647 (±0.051) | 0.287 (±0.075) |
| Uni.+POS Bigrams | 0.676 (±0.057) | **0.349 (±0.083)** |
| Uni.+POS Trigrams | 0.661 (±0.050) | 0.317 (±0.064) |
| Uni.+Dep. Lex | 0.639 (±0.055) | 0.268 (±0.079) |
| Uni.+Dep. Head-Back-off | **0.679 (±0.063)** | **0.351 (±0.097)** |
| Uni.+Dep. Mod-Back-off | 0.657 (±0.056) | 0.308 (±0.063) |
| Uni.+Dep. Head-Mod-Back-off | 0.670 (±0.046) | 0.336 (±0.065) |

# Results

| Feature Set | Accuracy | Kappa |
|---|---|---|
| Unigrams | 0.652 (±0.048) | 0.295 (±0.049) |
| Uni.+Bigrams | 0.657 (±0.066) | 0.304 (±0.089) |
| Uni.+Trigrams | 0.655 (±0.062) | 0.306 (±0.077) |
| Uni.+Back-off Bigrams | 0.650 (±0.056) | 0.299 (±0.079) |
| Uni.+Back-off Trigrams | 0.647 (±0.051) | 0.287 (±0.075) |
| Uni.+POS Bigrams | 0.676 (±0.057) | **0.349 (±0.083)** |
| Uni.+POS Trigrams | 0.661 (±0.050) | 0.317 (±0.064) |
| Uni.+Dep. Lex | 0.639 (±0.055) | 0.268 (±0.079) |
| Uni.+Dep. Head-Back-off | **0.679 (±0.063)** | **0.351 (±0.097)** |
| Uni.+Dep. Mod-Back-off | 0.657 (±0.056) | 0.308 (±0.063) |
| Uni.+Dep. Head-Mod-Back-off | 0.670 (±0.046) | 0.336 (±0.065) |

# Results

| Feature Set | Accuracy | Kappa |
|---|---|---|
| Unigrams | 0.652 (±0.048) | 0.295 (±0.049) |
| Uni.+Bigrams | 0.657 (±0.066) | 0.304 (±0.089) |
| Uni.+Trigrams | 0.655 (±0.062) | 0.306 (±0.077) |
| Uni.+Back-off Bigrams | 0.650 (±0.056) | 0.299 (±0.079) |
| Uni.+Back-off Trigrams | 0.647 (±0.051) | 0.287 (±0.075) |
| Uni.+POS Bigrams | 0.676 (±0.057) | **0.349 (±0.083)** |
| Uni.+POS Trigrams | 0.661 (±0.050) | 0.317 (±0.064) |
| Uni.+Dep. Lex | 0.639 (±0.055) | 0.268 (±0.079) |
| Uni.+Dep. Head-Back-off | **0.679 (±0.063)** | **0.351 (±0.097)** |
| Uni.+Dep. Mod-Back-off | 0.657 (±0.056) | 0.308 (±0.063) |
| Uni.+Dep. Head-Mod-Back-off | 0.670 (±0.046) | 0.336 (±0.065) |

# Results

| Feature Set | Accuracy | Kappa |
|---|---|---|
| Unigrams | 0.652 (±0.048) | 0.295 (±0.049) |
| Uni.+Bigrams | 0.657 (±0.066) | 0.304 (±0.089) |
| Uni.+Trigrams | 0.655 (±0.062) | 0.306 (±0.077) |
| Uni.+Back-off Bigrams | 0.650 (±0.056) | 0.299 (±0.079) |
| Uni.+Back-off Trigrams | 0.647 (±0.051) | 0.287 (±0.075) |
| Uni.+POS Bigrams | 0.676 (±0.057) | **0.349 (±0.083)** |
| Uni.+POS Trigrams | 0.661 (±0.050) | 0.317 (±0.064) |
| Uni.+Dep. Lex | 0.639 (±0.055) | 0.268 (±0.079) |
| Uni.+Dep. Head-Back-off | **0.679 (±0.063)** | **0.351 (±0.097)** |
| Uni.+Dep. Mod-Back-off | 0.657 (±0.056) | 0.308 (±0.063) |
| Uni.+Dep. Head-Mod-Back-off | 0.670 (±0.046) | 0.336 (±0.065) |

# Results

| Feature Set | Accuracy | Kappa |
|---|---|---|
| Unigrams | 0.652 (±0.048) | 0.295 (±0.049) |
| Uni.+Bigrams | 0.657 (±0.066) | 0.304 (±0.089) |
| Uni.+Trigrams | 0.655 (±0.062) | 0.306 (±0.077) |
| Uni.+Back-off Bigrams | 0.650 (±0.056) | 0.299 (±0.079) |
| Uni.+Back-off Trigrams | 0.647 (±0.051) | 0.287 (±0.075) |
| Uni.+POS Bigrams | 0.676 (±0.057) | **0.349 (±0.083)** |
| Uni.+POS Trigrams | 0.661 (±0.050) | 0.317 (±0.064) |
| Uni.+Dep. Lex | 0.639 (±0.055) | 0.268 (±0.079) |
| Uni.+Dep. Head-Back-off | **0.679 (±0.063)** | **0.351 (±0.097)** |
| Uni.+Dep. Mod-Back-off | 0.657 (±0.056) | 0.308 (±0.063) |
| Uni.+Dep. Head-Mod-Back-off | 0.670 (±0.046) | 0.336 (±0.065) |

# Discussion

- “Head-Back-off” features
  - significantly better than unigrams-only baseline
  - represent a better way to use dependency relations

- Generalizing them further (full back-off features) worsens performance

# Head-Back-off Successes

- *It is perhaps too small.*
  - cop(JJ,is)

- *The lens retracts and has its own metal cover so you don't need to fuss with a lens cap.*
  - det(NN,the), poss(NN,its), neg(VB,n’t)

- *The panorama setting is unbelievable!*
  - cop(JJ,is), det(NN,the)

# Head-Back-off Successes

- *Even with the waterproof housing it is small.*
  - cop(JJ,is), nsubj(JJ,it), det(NN,the)
- *The auto color balance is often fooled by the clouds.*
  - det(NN,the)
- *First, I have to say that I have NEVER had the slightest problem with this camera or the software.*
  - det(NN,this)

# Error Analysis

- Sentences that are opinions, but not about the product's features
  - *Really like them, they work well and the macro function of the 2500 really helps my Ebay biz.*
    - "Really like them" is about two other cameras the reviewer owns, not the camera being reviewed.
  - *I have had this little gem for four months now.*
    - About the product as a whole (not any particular explicitly mentioned feature)

# Error Analysis

- Few misclassifications due to Head-Back-off features “misfiring”
  - *Some people, in their reviews, complain about its small size, and how it doesn’t compare with larger cameras.*
    - Misclassified as “opinion sentence”
    - poss(NN,its), neg(VB,n’t)
  - *The closest competitor is the SONY DSC-P1 (3.3mp).*
    - Misclassified as “opinion sentence”
    - cop(NN,is), det(NN,the)

# Conclusions

- “Head-Back-off” features are a sweet spot
  - not too specific, not too general
- Future work:
  - Explore relation to supervised domain adaptation

Questions?