Exploiting Syntax in Sentiment Polarity Classification

Wolfgang Seeker
joint work with
Adam Bermingham, Jennifer Foster, Deirdre Hogan

Dublin City University

February 4, 2009

Subjectivity

Subjectivity

Subjective language refers to all aspects of natural language used to express **opinions**, **evaluations** or **speculations**.

Aspects of subjectivity in natural language (Wiebe et al. [2004]):

- lexical (*complain*, *pathetic*, *excitingly*, *hero*)
- phrasal (*stand in awe*, *what a NP*)
- morphosyntactic (*fronting*, *parallelism*, *aspect changes*)
- symbolic (*:-)*, -.-)

Examples (Wiebe 2004)

Opinionated

- We stand in awe of the Woodstock's generation's ability to be unceasingly fascinated by the subject of itself.
- At several different layers, it's a fascinating tale.
- There is nothing original or creative and little enjoyable in the film. (Movie Review Corpus)

Neutral

- Bell Industries Inc. increased its quarterly to 10 cents from 7 cents a share.

Sentiment Analysis I

In e.g. TREC 2008, three different tasks are defined:

- Find relevant blog posts
- **Find opinionated blog posts**
- **Find negative & positive blog posts**

Opinion Finding

Opinion finding techniques try to separate texts describing facts from those that express opinion.

Sentiment Analysis I

In e.g. TREC 2008, three different tasks are defined:

- Find relevant blog posts
- **Find opinionated blog posts**
- **Find negative & positive blog posts**

Polarity Classification

Polarity classification tries to classify texts according to the polarity of the opinion expressed in them (negative/positive).

Sentiment Analysis II

Sentiment Analysis in NLP:

- information extraction
- text, email, review classification/categorisation
- text summarisation
- (multiperspective) question answering
- flame recognition
- ...

Polarity Classification

Polarity Classification has been applied to different fields:

- Blogs (Bermingham et al. [2008],Ounis et al. [2008])
- Customer Feedback (Gamon [2004])
- Movie Reviews (Pang et al. [2002])
- Product Reviews (Turney [2002])
- News

Bag of Words

Bag of Words - Baseline

The bag of words approach uses word frequency/occurrence as features to learn a model. It's often used as a baseline.

Example

POS: I really like this movie because i like Sean Connery . He plays a convincing King Richard .
NEG: The film is THE HITCHER and as a friend of mine would say , it sucks pond water .

$\Downarrow$

1 <i:2 really:1 like:2 this:1 movie:1 because:1 sean:1 connery:1 he:1 plays:1 a:1 convincing:1 king:1 richard:1 .:2 the:0 film:0 is:0 hitcher:0 and:0 as:0 friend:0 of:0 mine:0 would:0 say:0 ,:0 it:0 sucks:0 pond:0 water:0>

Bag of Words

Bag of Words - Baseline

The bag of words approach uses word frequency/occurrence as features to learn a model. It's often used as a baseline.

Example

POS: I really like this movie because i like Sean Connery . He plays a convincing King Richard .
NEG: The film is THE HITCHER and as a friend of mine would say , it sucks pond water .

$\Downarrow$

1 <i:1 really:1 like:1 this:1 movie:1 because:1 sean:1 connery:1 he:1 plays:1 a:1 convincing:1 king:1 richard:1 .:1 the:0 film:0 is:0 hitcher:0 and:0 as:0 friend:0 of:0 mine:0 would:0 say:0 ,:0 it:0 sucks:0 pond:0 water:0>

Bag of Words

Bag of Words - Baseline

The bag of words approach uses word frequency/occurrence as features to learn a model. It's often used as a baseline.

Example

POS: I really like this movie because i like Sean Connery . He plays a convincing King Richard .
NEG: The film is THE HITCHER and as a friend of mine would say , it sucks pond water .

$\Downarrow$

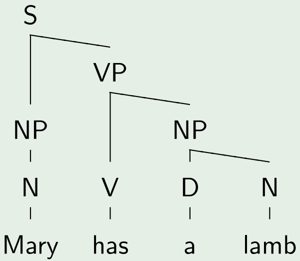
-1 <i:0 really:0 like:0 this:0 movie:0 because:0 sean:0 connery:0 he:0 plays:0 a:1 convincing:0 king:0 richard:0 .:1 the:1 film:1 is:1 hitcher:1 and:1 as:1 friend:1 of:1 mine:1 would:1 say:1 ,:1 it:1 sucks:1 pond:1 water:1>

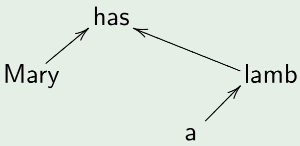
What's a Parse Feature?

Parse Features

Parse/Syntactic features are relations between words according to a grammar and are supposed to reflect semantic relations.

Phrase Structure Tree & Dependency Tree





Where Parse Features Might Help Us

Example

There is nothing original or creative and little enjoyable in the film.

Where Parse Features Might Help Us

Example

There is nothing original or creative and little enjoyable in the film.

A bag of words approach will give us these features:
<there:1 is:1 nothing:1 original:1 or:1 creative:1 and:1 little:1 enjoyable:1 in:1 the:1 film:1>

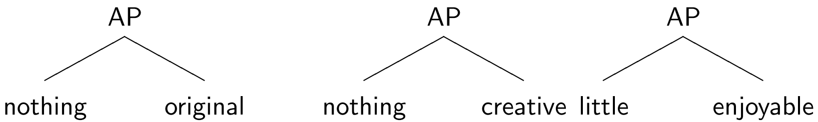
Where Parse Features Might Help Us

Example

There is nothing original or creative and little enjoyable in the film.

A bag of words approach will give us these features:
<there:1 is:1 nothing:1 original:1 or:1 creative:1 and:1 little:1 enjoyable:1 in:1 the:1 film:1>

A phrase structure tree might give us those instead (among others):



(You would need an 4gram model to capture *nothing creative*)

Some Previous Work Using Parse Features

- Matsumoto et al. [2005] used fragments of dependency parse trees to classify movie reviews
- Lerman et al. [2008] used fragments of dependency parse trees based containing keywords in order to predict the impact of daily news on the sentiment towards candidates in the U.S. presidential elections in 2004.
- ...

Where even Parse Features Won't Help

It can be a hard task ...

- THE TOXIC AVENGER is a funny film for anyone who can laugh for an hour and a half at the same joke premise with little assistance from the rest of an amateurish script.
- If you are reading this because it is your darling fragrance, please wear it at home exclusively, and tape the windows shut.
  (review by Luca Turin and Tania Sanchez of the Givenchy perfume Amarige, in Perfumes: The Guide, Viking 2008.) (in Pang and Lee [2008])

What We are Interested in

Coming from the LORG-Project, where we're developing a parsing toolkit for practical use in real-world applications, we are interested mainly in two questions:

- What kind of parser output are best for polarity classification?

What We are Interested in

Coming from the LORG-Project, where we're developing a parsing toolkit for practical use in real-world applications, we are interested mainly in two questions:

- What kind of parser output are best for polarity classification?
- What is the best way to represent parser output as a feature vector?

Data Set - Movie Reviews

Pang et al. [2002],Pang and Lee [2004] used internet movie reviews for polarity classification:

- it will be opinionated (less need for opinion filtering mechanisms)
- overall ratings are included (class labels for free)
- real-world language use
- closed domain
- freely available

This data set has often been used in the literature and would enable us to compare our results to others.

Our Movie Review Corpus

To avoid overfitting to the testset (Pang&Lee review corpus), we created our own review corpus as a development set:

- 7000 reviews from Internet Movie Data Base (*http://us.imdb.com/Reviews*)
- 3500 positive and 3500 negative documents
- class labels based on review ratings, which were removed automatically afterwards
- used a modified version of a script by Joachim Wagner for preprocessing (sentence splitting etc.)
- TreeTagger (Schmid [1994]) was used to tag input for dependency parsers
- parsed with various parsers generating four different types of syntactic structures
- not as clean as Pang&Lee corpus (But maybe more realistic?)

Data

Parse data:

- Phrase structure trees: Berkeley Parser, Stanford Parser
- Dependency trees: Malt Parser, MST Parser, KSDep Parser
- Dependency triples: DCU Annotation Algorithm, Stanford Parser
- f-Structures: DCU Annotation Algorithm

Data

Parse data:

- Phrase structure trees: Berkeley Parser, Stanford Parser
- Dependency trees: Malt Parser, MST Parser, KSDep Parser
- Dependency triples: DCU Annotation Algorithm, Stanford Parser
- f-Structures: DCU Annotation Algorithm

subj(mary∼1, has∼0)
num(mary∼1, sg)
obj(lamb∼3, has∼0)
num(lamb∼3, sg)
def(lamb∼3, -)
...

$$\begin{bmatrix} PRED & 'have < SUBJ, OBJ >' \\ SUBJ & \begin{bmatrix} PRED & 'mary' \\ NUM & sg \end{bmatrix} \\ OBJ & \begin{bmatrix} PRED & 'lamb' \\ NUM & sg \\ DEF & - \end{bmatrix} \\ TENSE & pres \end{bmatrix}$$

Learning Algorithm

Support Vector Machines

In a high-dimensional vector space, find that hyperplane that separates the training data best by maximising the distance to the nearest instances of both classes.

- binary classifier
- based on a vector product to measure the similarity between two instances
- one of the best performing classifiers today

Open source implementation:
SVMLight by Thorsten Joachims (Joachims [1999])

Feature Representation I

I. Precompute your features:

- Lerman et al. [2008] extract relations from dependency trees that contain certain keywords
- Matsumoto et al. [2005] use FREQT to precompute all occurring subtrees in a set of dependency trees and use those which occur more often than a certain threshold (20).

This means, enumerate all possible features (subtrees) and then put their frequency counts into a vector!

- exponential time complexity
- only "useful" features can be selected

Feature Representation II

II. Use Tree Kernels:

Tree Kernel

Tree kernels are algorithms that measure the similarity of two given trees by counting their common substructures. Tree kernels might differ in the kind of substructures they consider.

- Replace the vector product in SVMs by a kernel algorithm
- Implicit evaluation of the feature space without enumerating every feature explicitly
- Polynomial time complexity
- Algorithms for phrase structure trees and dependency trees
- SVMLightTK (Moschitti [2006]) introduces two tree kernels to SVMLight

Tree Kernels in SVMLightTK

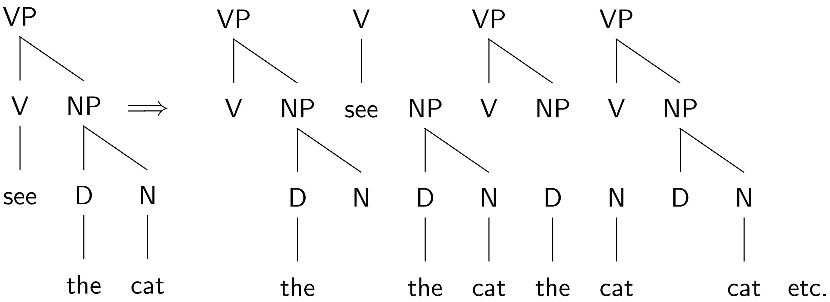
- SubSetTree Kernel by Collins and Duffy [2002], Moschitti [2006]
- SubTree Kernel by Vishwanathan and Smola [2002], Moschitti [2006]

Tree Kernel Algorithm - informal

- For every node pair between two trees the productions are checked
- If the productions are different $\longrightarrow$ no common subtree
- If the productions are equal and the nodes are preterminals $\longrightarrow$ common subtree
- If the productions are equal and the nodes are not preterminals $\longrightarrow$ check all daughters

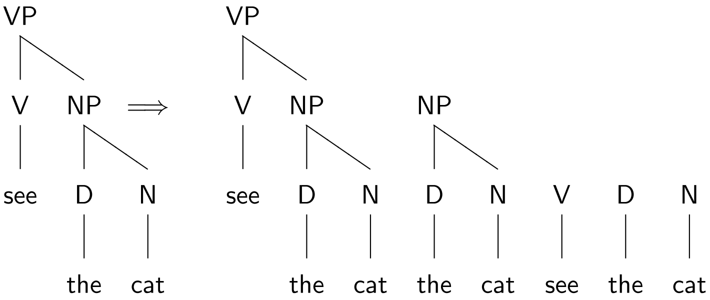
Implicit Feature Space

SubSetTree Kernel by Collins and Duffy [2002]

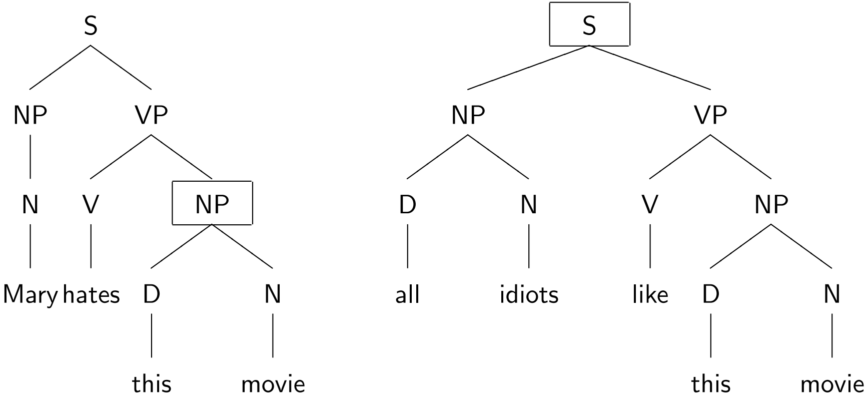


Implicit Feature Space

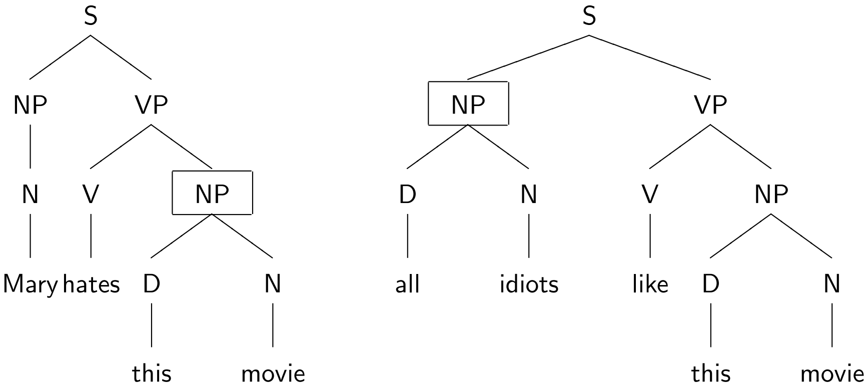
SubTree Kernel by Vishwanathan and Smola [2002]



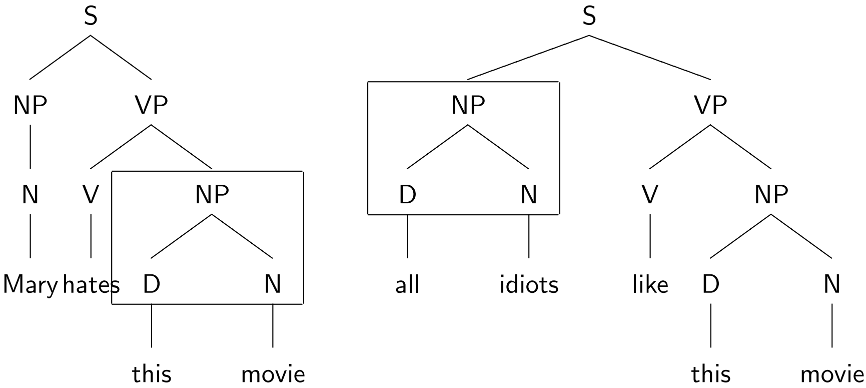
Subtree Kernel - Example



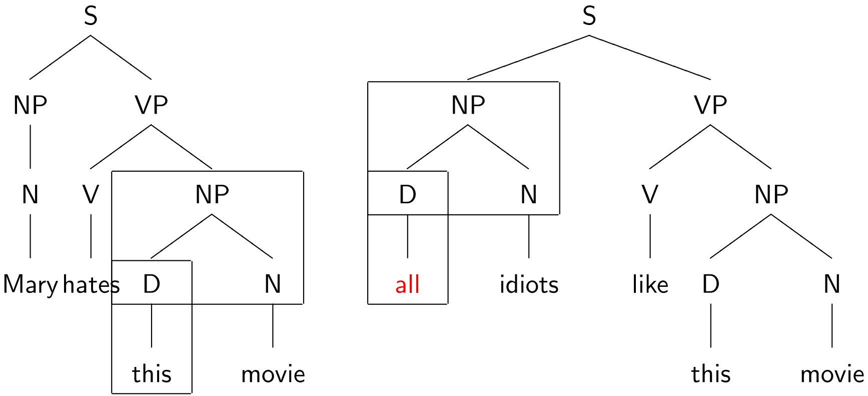
Subtree Kernel - Example



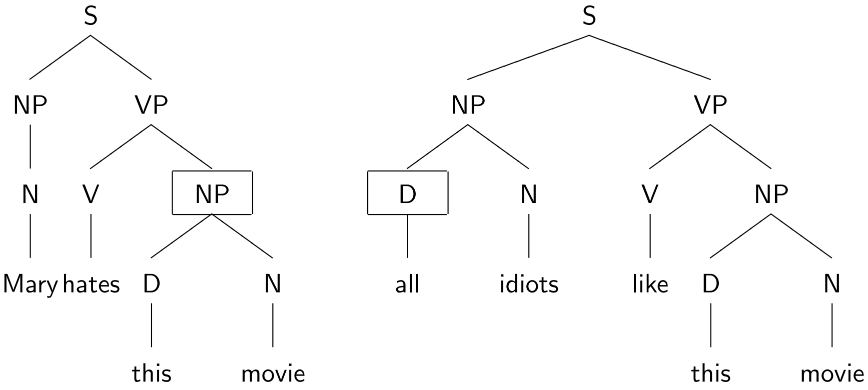
Subtree Kernel - Example



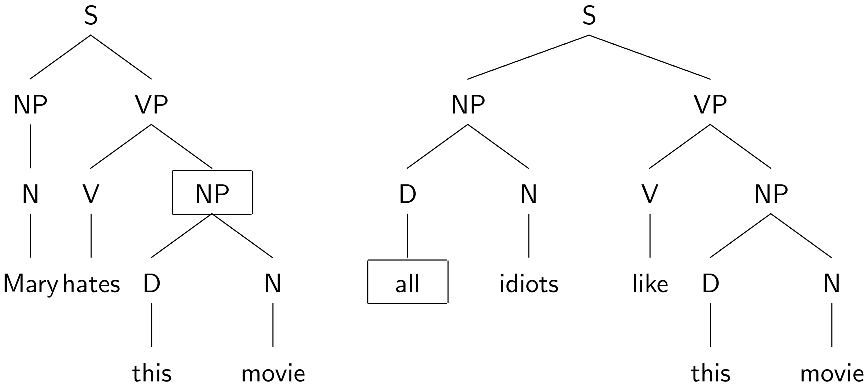
Subtree Kernel - Example



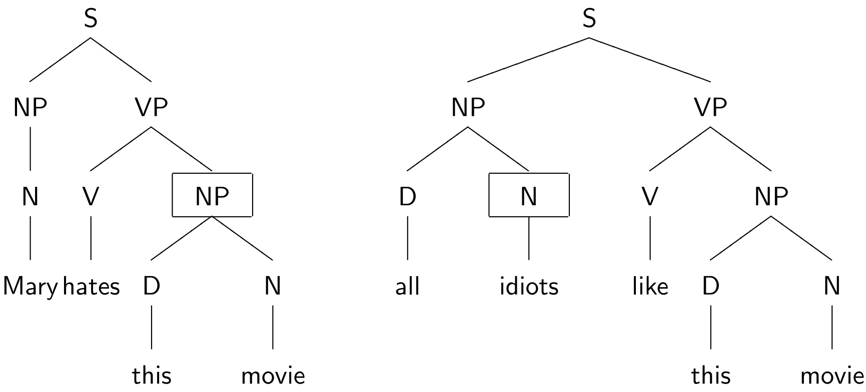
Subtree Kernel - Example



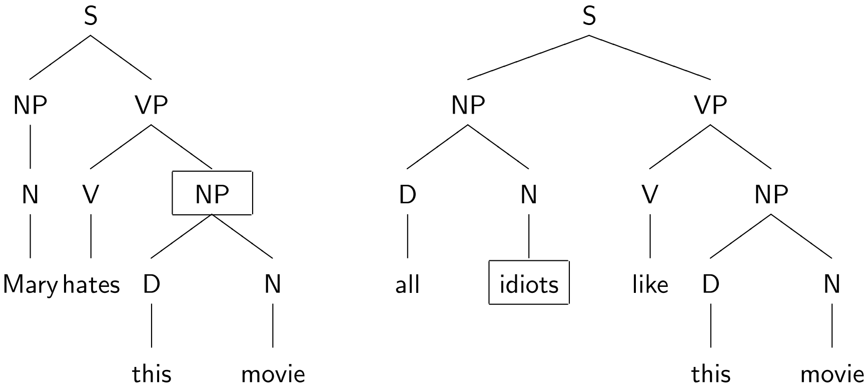
Subtree Kernel - Example



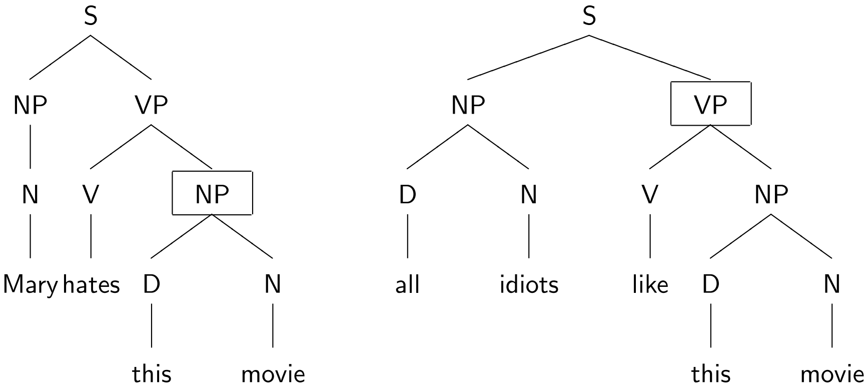
Subtree Kernel - Example



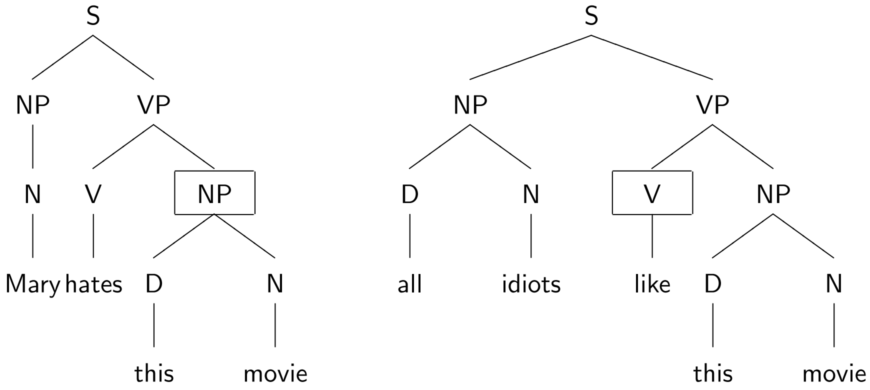
Subtree Kernel - Example



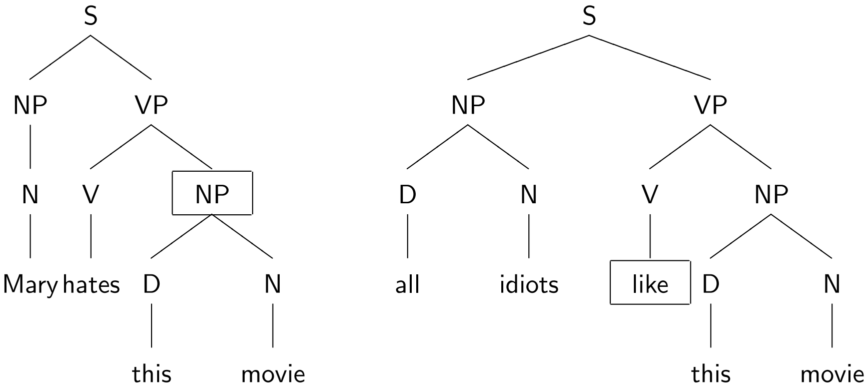
Subtree Kernel - Example



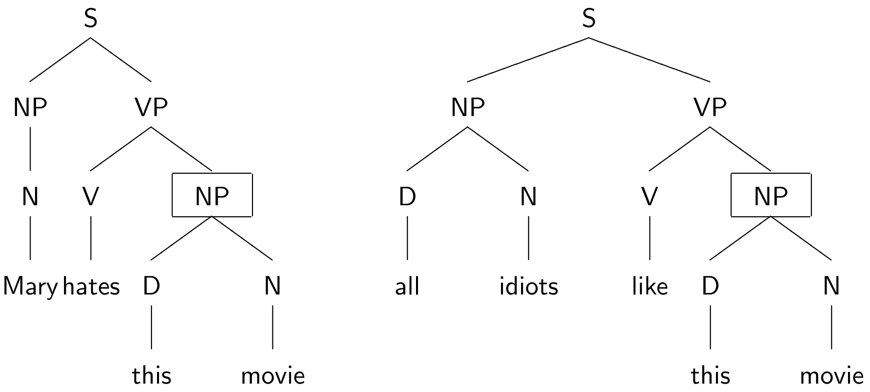
Subtree Kernel - Example



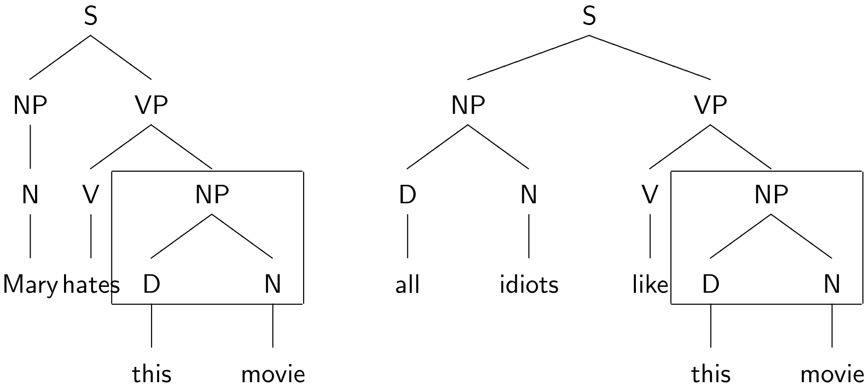
Subtree Kernel - Example



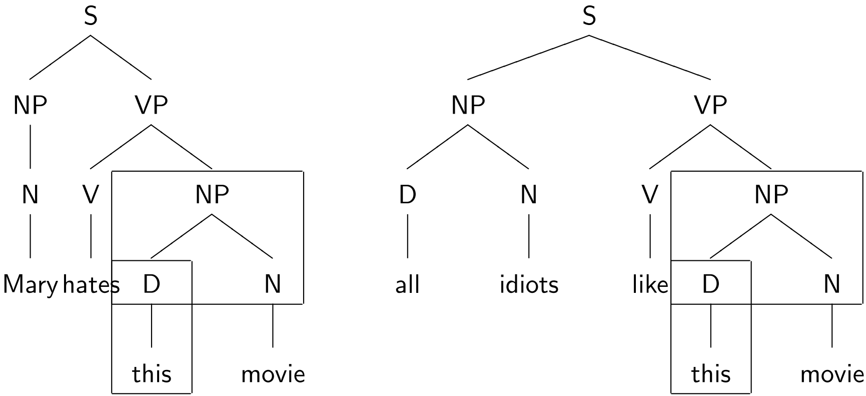
Subtree Kernel - Example



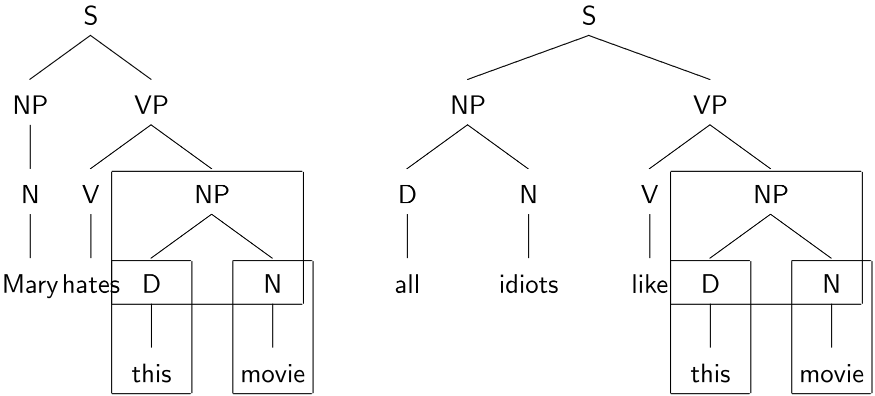
Subtree Kernel - Example



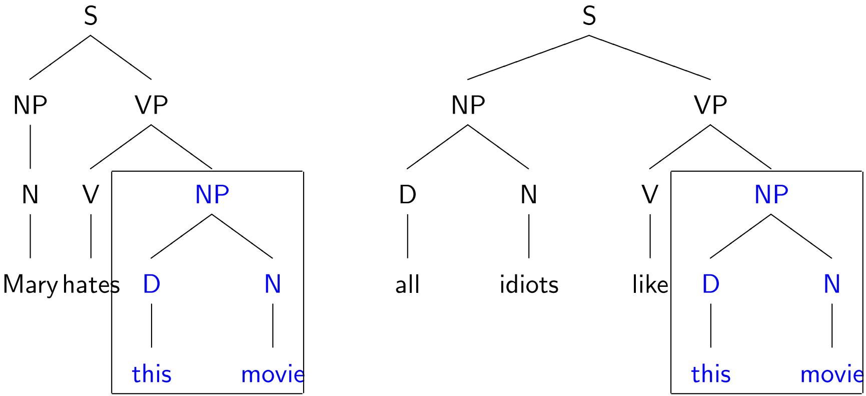
Subtree Kernel - Example



Subtree Kernel - Example



Subtree Kernel - Example



Tree Kernels in NLP

Tree kernels have been applied to a number of different NLP tasks:

- Semantic role labelling (Pighin et al. [2008])
- Relation extraction (Culotta and Sorensen [2004]), protein-pair interaction extraction (Miyao et al. [2008])
- Question classification (Pan et al. [2008])

Note that all of this is sentence level classification!

Challenges

Issues with tree kernels:

- document level vs. sentence level (no sentence level labels)
- no feature selection $\rightarrow$ huge feature space
- maybe data sparseness
- overfitting (?)

Tree kernels will prove useful if we find a good way of reducing the feature space.

Challenges

Issues with tree kernels:

- document level vs. sentence level (no sentence level labels)
- no feature selection → huge feature space
- maybe data sparseness
- overfitting (?)

Tree kernels will prove useful if we find a good way of reducing the feature space.
The usual suspects:

- No gold trees → lots of noise
  like preprocessing errors, tagging errors, parser errors, real-world orthography
- Penn Treebank Tagset is not fine grained enough
  (DT can be *no* and *the*)

Ongoing Work

- Lemmatising the trees seems to help a little bit (feature reduction)
- Annotating trees with SentiWordNet scores might help (but probably more features)
  - Find a clever way of getting the right score (multiple senses)
- Pruning the trees (how to decide what to keep?)
- Filter out sentences by
  - Use SentiWordNet to filter out objective sentences
  - Exclude sentences based on their position (plot description, actor lists)
  - Use domain-specific keyword lists to find relevant sentences

References I

Adam Bermingham, Alan F. Smeaton, Jennifer Foster, and Deirdre Hogan. Dcu at the trec 2008 blog track. In *TREC 2008 - Text REtrieval Conference*, 2008. URL http://doras.dcu.ie/2196/.

M. Collins and N. Duffy. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems*, 2002.

Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423, Morristown, NJ, USA, 2004. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1218955.1219009. URL http://www.cs.umass.edu/~culotta/pubs/culotta04dependency.pdf.

Michael Gamon. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and role of linguistic analysis. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 611–617, 2004. URL http://research.microsoft.com/nlp/publications/coling2004_sentiment.pdf.

References II

T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1999.

Kevin Lerman, Ari Gilder, Mark Dredze, and Fernando Pereira. Reading the markets: Forecasting public opinion of political candidates by news analysis. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*, Manchester, United Kingdom, 2008.

Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. Sentiment classification using word sub-sequences and dependency sub-trees. In Tu Bao Ho, David Cheung, and Huan Li, editors, *Proceeding of PAKDD'05, the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, volume 3518 of *Lecture Notes in Computer Science*, pages 301–310, Hanoi, VN, 2005. Springer-Verlag. doi: http://dx.doi.org/10.1007/11430919_37.

References III

Yusuke Miyao, Rune Saetre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. Task-oriented evaluation of syntactic parsers and their representations. In *Proceedings of the 46th Annual Meeting of the ACL*, pages 46–54, Columbus, Ohio, June 2008.

Alessandro Moschitti. Making tree kernels practical for natural language learning. In *EACL*, 2006. URL `http://acl.ldc.upenn.edu/E/E06/E06-1015.pdf`.

Iadh Ounis, Craig Macdonald, and Ian Soboroff. Overview of the trec-2008 blog trac. In *The Seventeenth Text REtrieval Conference (TREC 2008) Proceedings*. NIST, 2008.

References IV

Yan Pan, Yong Tang, Luxin Lin, and Yemin Luo. Question classification with semantic tree kernel. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 837–838, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: http://doi.acm.org/10.1145/1390334.1390530. URL http://portal.acm.org/citation.cfm?id=1390530.

B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1-2):1–135, 2008. URL http://www.cs.cornell.edu/home/llee/omsa/omsa-published.pdf.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL-04, 42nd Meeting of the Association for Computational Linguistics*, pages 271–278, Barcelona, ES, 2004. Association for Computational Linguistics. URL http://www.cs.cornell.edu/home/llee/papers/cutsent.pdf.

References V

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP-02, the Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Philadelphia, US, 2002. Association for Computational Linguistics. URL `http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf`.

Daniele Pighin, Alessandro Moschitti, and Roberto Basili. Tree kernels for semantic role labeling. *Computational Linguistics Journal*, 2008.

Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, September 1994. URL `http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.pdf`.

Peter D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *ACL*, pages 417–424, 2002. URL `http://www.aclweb.org/anthology/P02-1053.pdf`.

References VI

S. V. N. Vishwanathan and A. J. Smola. Fast kernels on strings and trees. 2002. URL `citeseer.ist.psu.edu/675716.html`.

Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning subjective language. *Computational Linguistics*, 30(3):277–308, September 2004. URL `http://acl.ldc.upenn.edu/J/J04/J04-3002.pdf`.

Sources of Subjectivity

Nested Sources (Wiebe 2004)

The Foreign Ministry said Thursday that it was "*surprised, to put it mildly*" by the U.S. State Department's *criticism* of Russia's human rights record and *objected* in particular to the "*odious*" section on Chechnya.

- *surprised, to put it mildly*:
  (author, Foreign Ministry, Foreign Ministry)
- *criticism*:
  (author, Foreign Ministry, Foreign Ministry, U.S. State Dep.)
- *objected*: (author, Foreign Ministry)
- *odious*: (author, Foreign Ministry)

It's not that easy to decide, whether a sentence is opinionated or not.