Feature Subsumption for Opinion Analysis

Opinions in Text

- Active area of research – sentiment analysis and opinion recognition
- Potential applications include:
  - Review mining
  - Product Reputation Analysis
  - Multi-perspective QA

Opinion Recognition

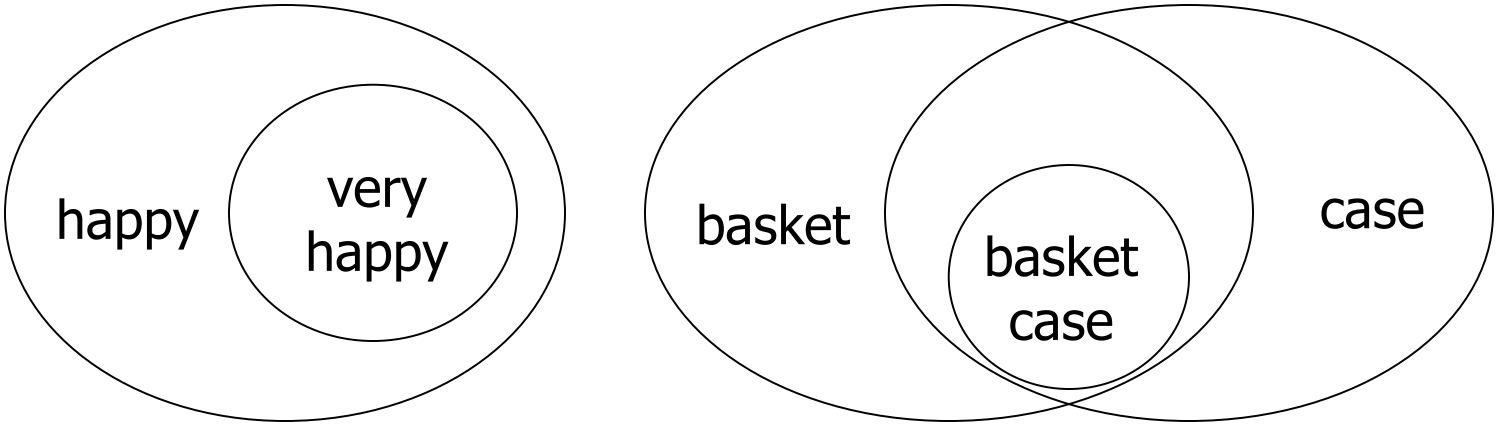
- Typically done using a Machine Learning system using lexical features:
  - Ngrams
  - Phrases
  - Lexico-syntactic patterns
- For example:
  - “rules”, “loses it”, “drives <np> crazy”

Text Representations

- 1Grams: *"disaster"*
- 2Grams: *"laughing stock"*
- IE Patterns:
  - *<subj> ActVP(recommends)*
  - *NP(benefits) PP(to)*

Some Observations

- Features vary in complexity
- There exists redundancy or representational overlap in features
  - For example:



Questions…

- Can we
  - analyze the features of varying complexities?
  - automatically identify complex features that outperform their simpler counterparts?
  - improve classification performance by removing redundancy/feature overlap?

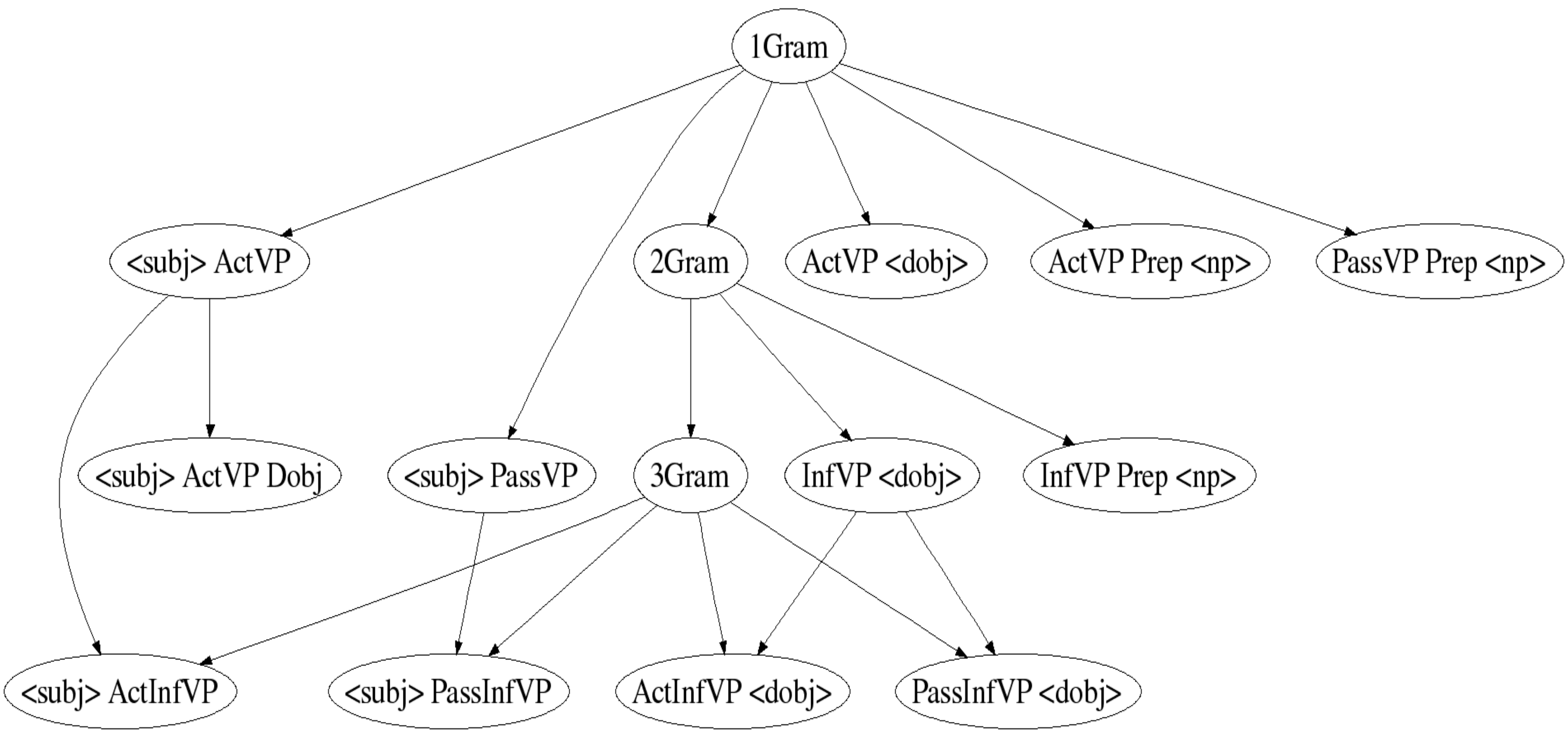
The Subsumption Hierarchy

- Defines representational scope of different types of features
- Defines rules for subsumption of features

Goals

- Use the hierarchy as an analysis tool to identify complex constructions that are better opinion indicators than their simpler components
- Automatically remove overlapping features with the aim of improving classification performance

The Hierarchy



Feature Type Definition

- Each node in the hierarchy defines the representation of each type of feature based on
  - Words
  - Sequential dependencies
  - Syntactic dependencies

Node Examples

Name = **2Gram**
Constituent[0] = WORD1
Constituent[1] = WORD2
Dependency = Sequential(0, 1)

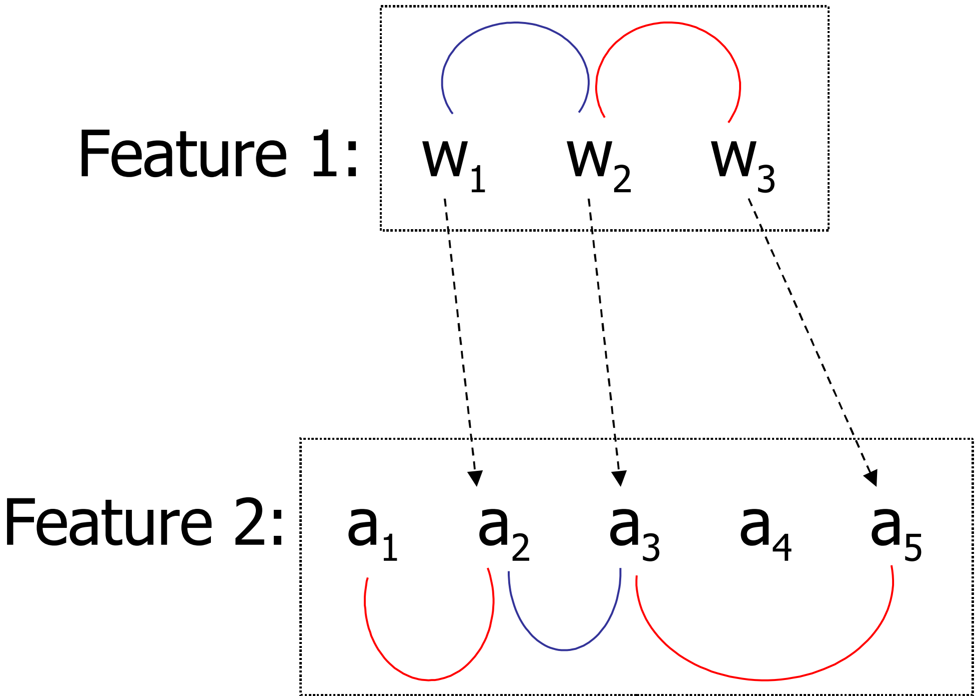
Name = **NP Prep <np>**
Constituent[0] = NP
Constituent[1] = PREP
Constituent[2] = NP_EXTRACTION
Dependency = Syntactic(0, 1)
Dependency = Syntactic(1, 2)

Name = **InfVP <dobj>**
Constituent[0] = INFINITIVE_TO
Constituent[1] = VERB
Constituent[2] = DOBJ_EXTRACTION
Dependency = Syntactic(0, 1)
Dependency = Syntactic(1, 2)
Dependency = Sequential(0, 1)

Representational Subsumption

- Feature *A* subsumes feature *B* iff
  - *A* is an ancestor of *B*
  - Words in *B* are a superset of words in *A*
  - Matching words in *B* are in the same relative order as those in *A*
  - Dependencies in *A* are mirrored between corresponding words in *B*

Representational Subsumption



The Subsumption Process

- Each feature is assigned to its appropriate node
- Top-down breadth-first traversal of hierarchy
  - Iterate over each feature at current node and pair with each feature at ancestors
  - Representational Subsumption applied to feature pairs

Behavioral Subsumption

- Without regard to performance, we would eliminate all features having a more general counterpart
- We define performance-based subsumption
- Feature *A* behaviorally subsumes *B* iff
  - *A* representationally subsumes *B*
  - $IG(A) \geq IG(B) - \delta$

Data Sets

- OP data set (Wiebe et al., 2004): document-level opinion classification. 2,452 documents (9% opinion).
- Polarity data set (Pang and Lee, 2004): document-level polarity classification. 1,400 documents (50% positive).
- MPQA data set (Wiebe et al., 2005): sentence-level opinion classification. 9,732 sentences (55% opinion).

Subsumption for Analysis

Feature	**IGain**	**Example**
line the line	.0016 .0075	… notes backed by credit *line* receivables … lays it on *the line*; … steps across *the line*
benefits NP(benefits) Prep(to)	.0040 .0090	… included $235,000 in tax *benefits* ... with no proven *benefits to* the economy
due ActVp(due) Prep(to)	.0001 .0038	…estimated $1.23 bn in debt *due* next week … it's all *due to* the intense scrutiny.

Subsumption for Analysis

Feature	IGain	Example
short nothing short	.0014 .0039	… to make a long story *short* … *nothing short* of spectacular
disaster AuxVp(be) Dobj(disaster)	.0010 .0048	… rated PG-13 for *disaster* related elements ... this *is* such a confused *disaster* of a film
work ActVp(work)	.0001 .0038	… the next day during drive to *work* … the film *will work* just as well

Reducing Feature Sets

- SVM trained on reduced feature sets
- For OP and Polarity data discarded features with $f < 5$
- For MPQA data discarded features with $f < 2$
- Results over 3-fold cross validation
- Experimented with varying δ

Accuracies: OP

Features	**Base**	**δ = 0.0005**	**δ = 0.001**	**δ = 0.002**
1Gram	97.5			
1+2Gram	98.0	**98.7**	98.6	**98.7**
1Gram+EP	97.2	97.8	97.9	97.9
1+2Gram+EP	97.8	98.6	**98.7**	**98.7**

Accuracies: Polarity

Features	Base	δ = 0.0005	δ = 0.001	δ = 0.002
1Gram	79.8			
1+2Gram	81.2	81.0	81.3	81.0
1Gram+EP	81.7	81.4	81.4	82.0
1+2Gram+EP	81.7	82.3	82.3	**82.7**

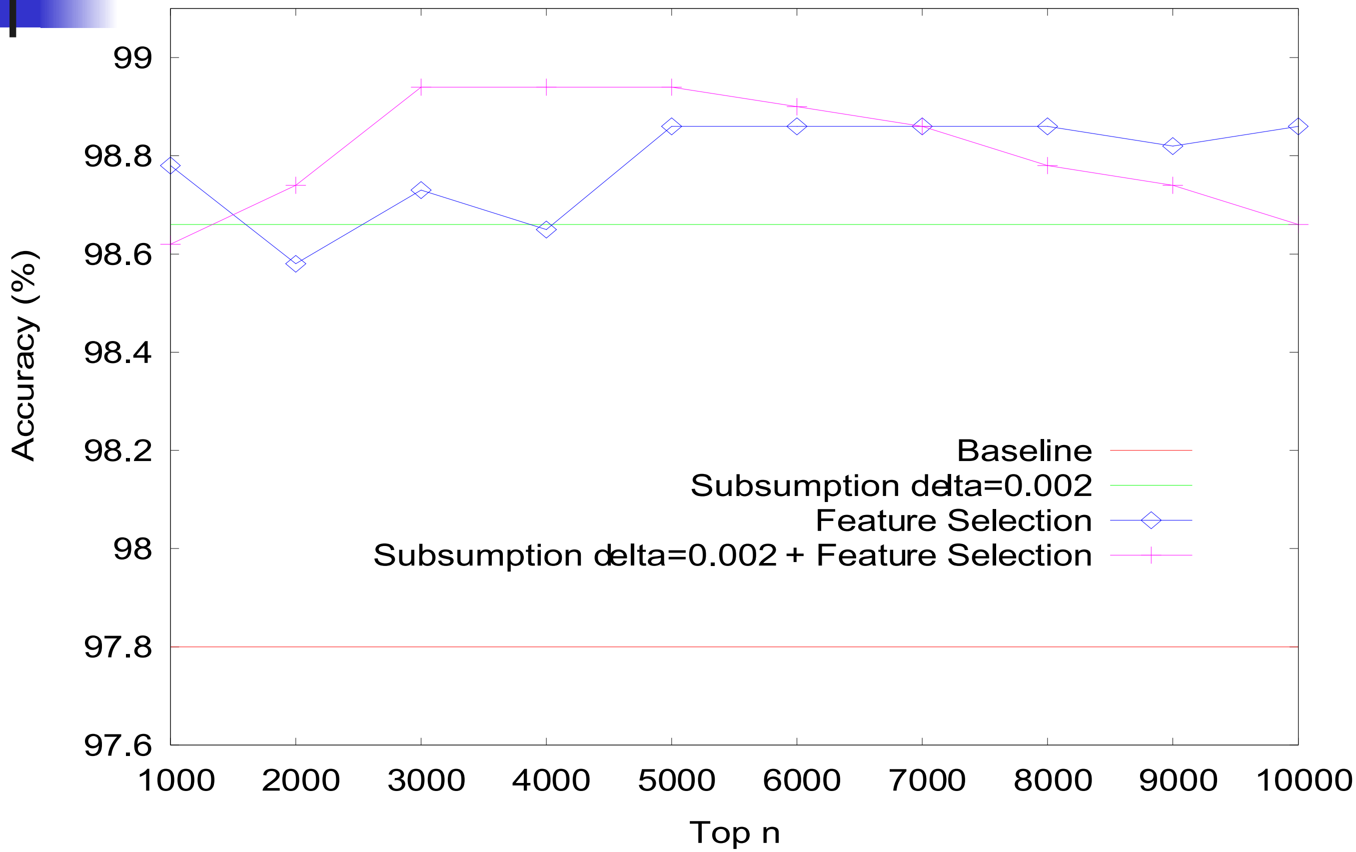
Accuracies: MPQA

Features	Base	δ = 0.0005	δ = 0.001	δ = 0.002
1Gram	74.8			
1+2Gram	74.3	**74.9**	74.6	74.8
1Gram+EP	74.4	74.6	74.6	74.6
1+2Gram+EP	74.4	**74.9**	74.7	74.6

Traditional Feature Selection

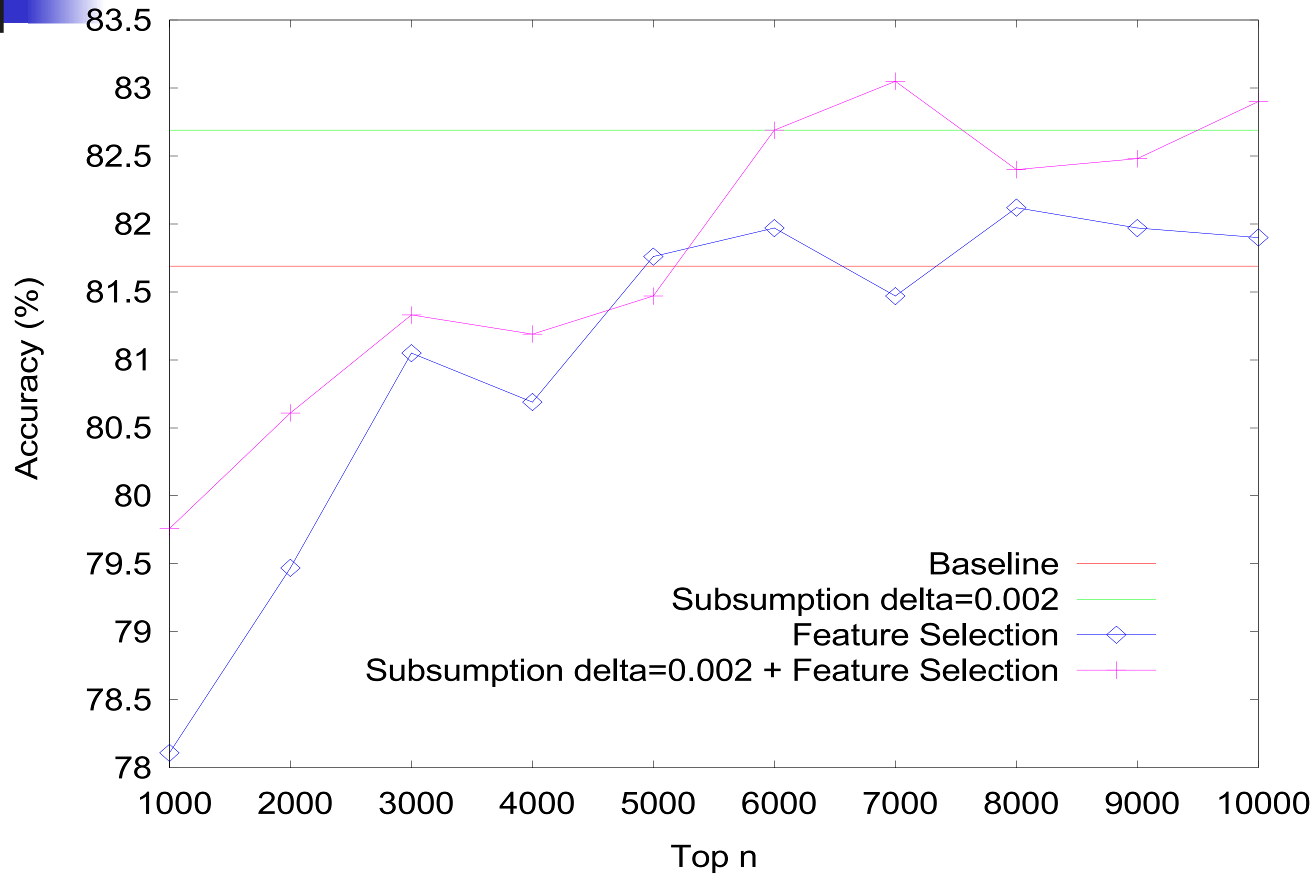
- Compared subsumption with traditional feature selection
- Selected top *N* features sorted by IGain
- Also applied feature selection to “subsumed” feature set
- Experimented with 1+2+EPs

Feature Selection: OP



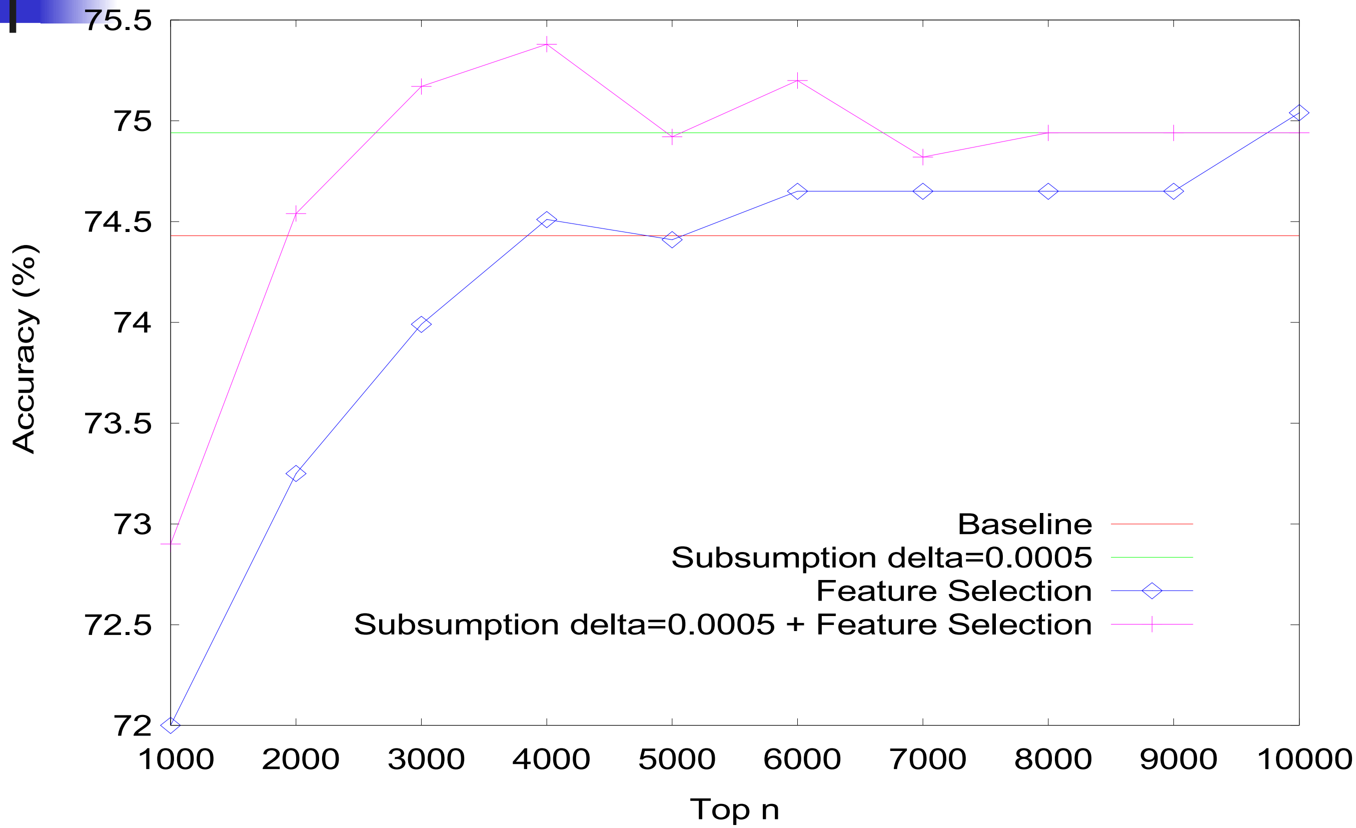
Feature Selection on OP Data

Feature Selection: Polarity



Feature Selection on Polarity Data

Feature Selection: MPQA



Feature Selection on MPQA Sentences

Conclusions

- Described a *subsumption hierarchy* of feature representations:
  - as an analytic tool
  - as an automatic tool to remove unnecessary features
  - improved opinion classification performance