Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, July 2006

Feature Subsumption for Opinion Analysis

Ellen Riloff and Siddharth Patwardhan
School of Computing
University of Utah
Salt Lake City, UT 84112
{riloff,sidd}@cs.utah.edu

Janyce Wiebe
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
wiebe@cs.pitt.edu

Abstract

Lexical features are key to many approaches to sentiment analysis and opinion detection. A variety of representations have been used, including single words, multi-word Ngrams, phrases, and lexico-syntactic patterns. In this paper, we use a *subsumption hierarchy* to formally define different types of lexical features and their relationship to one another, both in terms of representational coverage and performance. We use the subsumption hierarchy in two ways: (1) as an analytic tool to automatically identify complex features that outperform simpler features, and (2) to reduce a feature set by removing unnecessary features. We show that reducing the feature set improves performance on three opinion classification tasks, especially when combined with traditional feature selection.

1 Introduction

Sentiment analysis and opinion recognition are active research areas that have many potential applications, including review mining, product reputation analysis, multi-document summarization, and multi-perspective question answering. Lexical features are key to many approaches, and a variety of representations have been used, including single words, multi-word Ngrams, phrases, and lexico-syntactic patterns. It is common for different features to overlap representationally. For example, the unigram "happy" will match all of the texts that the bigram "very happy" matches. Since both features represent a positive sentiment and the bigram matches fewer contexts than the unigram, it is probably sufficient just to have the unigram. However, there are many cases where a feature captures a subtlety or non-compositional meaning that a simpler feature does not. For example, "basket case" is a highly opinionated phrase, but the words "basket" and "case" individually are not. An open question in opinion analysis is how often more complex feature representations are needed, and which types of features are most valuable. Our first goal is to devise a method to automatically identify features that are representationally subsumed by a simpler feature but that are better opinion indicators. These subjective expressions could then be added to a subjectivity lexicon (Esuli and Sebastiani, 2005), and used to gain understanding about which types of complex features capture meaningful expressions that are important for opinion recognition.

Many opinion classifiers are created by adopting a "kitchen sink" approach that throws together a variety of features. But in many cases adding new types of features does not improve performance. For example, Pang et al. (2002) found that unigrams outperformed bigrams, and unigrams outperformed the combination of unigrams plus bigrams. Our second goal is to automatically identify features that are unnecessary because similar features provide equal or better coverage and discriminatory value. Our hypothesis is that a reduced feature set, which selectively combines unigrams with only the most valuable complex features, will perform better than a larger feature set that includes the entire "kitchen sink" of features.

In this paper, we explore the use of a *subsumption hierarchy* to formally define the subsumption relationships between different types of textual features. We use the subsumption hierarchy in two ways. First, we use subsumption as an an-

alytic tool to compare features of different complexities and automatically identify complex features that substantially outperform their simpler counterparts. Second, we use the subsumption hierarchy to reduce a feature set based on representational overlap and on performance. We conduct experiments with three opinion data sets and show that the reduced feature sets can improve classification performance.

2 The Subsumption Hierarchy

2.1 Text Representations

We analyze two feature representations that have been used for opinion analysis: Ngrams and Extraction Patterns. *Information extraction (IE) patterns* are lexico-syntactic patterns that represent expressions which identify role relationships. For example, the pattern *"<subj> ActVP(recommended)"* extracts the subject of active-voice instances of the verb "recommended" as the recommender. The pattern *"<subj> PassVP(recommended)"* extracts the subject of passive-voice instances of "recommended" as the object being recommended.

(Riloff and Wiebe, 2003) explored the idea of using extraction patterns to represent more complex subjective expressions that have non-compositional meanings. For example, the expression *"drive (someone) up the wall"* expresses the feeling of being annoyed, but the meanings of the words "drive", "up", and "wall" have no emotional connotations individually. Furthermore, this expression is not a fixed word sequence that can be adequately modeled by Ngrams. Any noun phrase can appear between the words "drive' and "up", so a flexible representation is needed to capture the general pattern *"drives <NP> up the wall"*.

This example represents a general phenomenon: many expressions allow intervening noun phrases and/or modifying terms. For example:

"stepped on <mods> toes"
Ex: *stepped on the boss' toes*

"dealt <np> <mods> blow"
Ex: *dealt the company a decisive blow*

"brought <np> to <mods> knees"
Ex: *brought the man to his knees*

(Riloff and Wiebe, 2003) also showed that syntactic variations of the same verb phrase can behave very differently. For example, they found that passive-voice constructions of the verb "ask" had a 100% correlation with opinion sentences, but active-voice constructions had only a 63% correlation with opinions.

Pattern Type	Example Pattern
<subj> PassVP	<subj> is satisfied
<subj> ActVP	<subj> complained
<subj> ActVP Dobj	<subj> dealt blow
<subj> ActInfVP	<subj> appear to be
<subj> PassInfVP	<subj> is meant to be
<subj> AuxVP Dobj	<subj> has position
<subj> AuxVP Adj	<subj> is happy
ActVP <dobj>	endorsed <dobj>
InfVP <dobj>	to condemn <dobj>
ActInfVP <dobj>	get to know <dobj>
PassInfVP <dobj>	is meant to be <dobj>
Subj AuxVP <dobj>	fact is <dobj>
NP Prep <np>	opinion on <np>
ActVP Prep <np>	agrees with <np>
PassVP Prep <np>	is worried about <np>
InfVP Prep <np>	to resort to <np>
<possessive> NP	<noun>'s speech

Figure 1: Extraction Pattern Types

Our goal is to use the subsumption hierarchy to identify Ngram and extraction pattern features that are more strongly associated with opinions than simpler features. We used three types of features in our research: unigrams, bigrams, and IE patterns. The Ngram features were generated using the *Ngram Statistics Package* (NSP) (Banerjee and Pedersen, 2003).[1] The extraction patterns (EPs) were automatically generated using the Sundance/AutoSlog software package (Riloff and Phillips, 2004). AutoSlog relies on the Sundance shallow parser and can be applied exhaustively to a text corpus to generate IE patterns that can extract every noun phrase in the corpus. AutoSlog has been used to learn IE patterns for the domains of terrorism, joint ventures, and microelectronics (Riloff, 1996), as well as for opinion analysis (Riloff and Wiebe, 2003). Figure 1 shows the 17 types of extraction patterns that AutoSlog generates. PassVP refers to passive-voice verb phrases (VPs), ActVP refers to active-voice VPs, InfVP refers to infinitive VPs, and AuxVP refers

[1] NSP is freely available for use under the GPL from http://search.cpan.org/dist/Text-NSP. We discarded Ngrams that consisted entirely of stopwords. We used a list of 281 stopwords.

to VPs where the main verb is a form of "to be" or "to have". Subjects (subj), direct objects (dobj), PP objects (np), and possessives can be extracted by the patterns.[2]

2.2 The Subsumption Hierarchy

We created a *subsumption hierarchy* that defines the representational scope of different types of features. We will say that feature A *representationally subsumes* feature B if the set of text spans that match feature A is a superset of the set of text spans that match feature B. For example, the unigram "happy" subsumes the bigram "very happy" because the set of text spans that match "happy" includes the text spans that match "very happy".

First, we define a hierarchy of valid subsumption relationships, shown in Figure 2. The 2Gram node, for example, is a child of the 1Gram node because a 1Gram can subsume a 2Gram. Ngrams may subsume extraction patterns as well. Every extraction pattern has at least one corresponding 1Gram that will subsume it.[3]. For example, the 1Gram "recommended" subsumes the pattern "<subj> ActVP(recommended)" because the pattern only matches active-voice instances of "recommended". An extraction pattern may also subsume another extraction pattern. For example, "<subj> ActVP(recommended)" subsumes "<subj> ActVP(recommended) Dobj(movie)".

To compare specific features we need to formally define the representation of each type of feature in the hierarchy. For example, the hierarchy dictates that a 2Gram can subsume the pattern "ActInfVP <dobj>", but this should hold only if the words in the bigram correspond to adjacent words in the pattern. For example, the 2Gram "to fish" subsumes the pattern "ActInfVP(like to fish) <dobj>". But the 2Gram "like fish" should not subsume it. Similarly, consider the pattern "InfVP(plan) <dobj>", which represents the infinitive "to plan". This pattern subsumes the pattern "ActInfVP(want to plan) <dobj>", but it should not subsume the pattern "ActInfVP(plan to start)".

To ensure that different features truly subsume each other representationally, we formally define each type of feature based on words, sequential dependencies, and syntactic dependencies. A *sequential dependency* between words w_i and w_{i+1} means that w_i and w_{i+1} must be adjacent, and that w_i must precede w_{i+1}. Figure 3 shows the formal definition of a bigram (2Gram) node. The bigram is defined as two words with a sequential dependency indicating that they must be adjacent.

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

Figure 3: 2Gram Definition

A *syntactic dependency* between words w_i and w_{i+1} means that w_i has a specific syntactic relationship to w_{i+1}, and w_i must precede w_{i+1}. For example, consider the extraction pattern "NP Prep <np>", in which the object of the preposition attaches to the NP. Figure 4 shows the definition of this extraction pattern in the hierarchy. The pattern itself contains three components: the NP, the attaching preposition, and the object of the preposition (which is the NP that the pattern extracts). The definition also includes two syntactic dependencies: the first dependency is between the NP and the preposition (meaning that the preposition syntactically attaches to the NP), while the second dependency is between the preposition and the extraction (meaning that the extracted NP is the syntactic object of the preposition).

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

Figure 4: "NP Prep <np>" Pattern Definition

Consequently, the bigram "affair with" will not subsume the extraction pattern "affair with <np>" because the bigram requires the noun and preposition to be adjacent but the pattern does not. For example, the extraction pattern matches the text *"an affair in his mind with Countess Olenska"* but the bigram does not. Conversely, the extraction pattern does not subsume the bigram either because the pattern requires syntactic attachment but the bigram does not. For example, the bigram matches

[2]However, the items extracted by the patterns are not actually used by our opinion classifiers; only the patterns themselves are matched against the text.

[3]Because every type of extraction pattern shown in Figure 1 contains at least one word (not including the extracted phrases, which are not used as part of our feature representation).

Figure 2: The Subsumption Hierarchy

the sentence *"He ended the affair with a sense of relief"*, but the extraction pattern does not.

Figure 5 shows the definition of another extraction pattern, "InfVP <dobj>", which includes both syntactic and sequential dependencies. This pattern would match the text *"to protest high taxes"*. The pattern definition has three components: the infinitive "to", a verb, and the direct object of the verb (which is the NP that the pattern extracts). The definition also shows two syntactic dependencies. The first dependency indicates that the verb syntactically attaches to the infinitive "to". The second dependency indicates that the extracted NP syntactically attaches to the verb (i.e., it is the direct object of that particular verb).

The pattern definition also includes a sequential dependency, which specifies that "to" must be adjacent to the verb. Strictly speaking, our parser does not require them to be adjacent. For example, the parser allows intervening adverbs to split infinitives (e.g., *"to strongly protest high taxes"*), and this does happen occasionally. But split infinitives are relatively rare, so in the vast majority of cases the infinitive "to" will be adjacent to the verb. Consequently, we decided that a bigram (e.g., *"to protest"*) should representationally subsume this extraction pattern because the syntactic flexibility afforded by the pattern is negligible. The sequential dependency link represents this judgment call that the infinitive "to" and the verb are adjacent in most cases.

For all of the node definitions, we used our best judgment to make decisions of this kind. We tried to represent major distinctions between features, without getting caught up in minor differences that were likely to be negligible in practice.

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)

Figure 5: "InfVP <dobj>" Pattern Definition

To use the subsumption hierarchy, we assign each feature to its appropriate node in the hierarchy based on its type. Then we perform a top-down breadth-first traversal. Each feature is compared with the features at its ancestor nodes. If a feature's words and dependencies are a superset of an ancestor's words and dependencies, then it is subsumed by the (more general) ancestor and discarded.[4] When the subsumption process is finished, a feature remains in the hierarchy only if

[4]The words that they have in common must also be in the same relative order.

there are no features above it that subsume it.

2.3 Performance-based Subsumption

Representational subsumption is concerned with whether one feature is more general than another. But the purpose of using the subsumption hierarchy is to identify more complex features that outperform simpler ones. Applying the subsumption hierarchy to features without regard to performance would simply eliminate all features that have a more general counterpart in the feature set. For example, all bigrams would be discarded if their component unigrams were also present in the hierarchy.

To estimate the quality of a feature, we use Information Gain (IG) because that has been shown to work well as a metric for feature selection (Forman, 2003). We will say that feature A *behaviorally subsumes* feature B if two criteria are met: (1) A representationally subsumes B, and (2) IG(A) $\geq$ IG(B) - δ, where δ is a parameter representing an acceptable margin of performance difference. For example, if δ=0 then condition (2) means that feature A is just as valuable as feature B because its information gain is the same or higher. If δ>0 then feature A is allowed to be a little worse than feature B, but within an acceptable margin. For example, δ=.0001 means that A's information gain may be up to .0001 lower than B's information gain, and that is considered to be an acceptable performance difference (i.e., A is good enough that we are comfortable discarding B in favor of the more general feature A).

Note that based on the subsumption hierarchy shown in Figure 2, all 1Grams will always survive the subsumption process because they cannot be subsumed by any other types of features. Our goal is to identify complex features that are worth adding to a set of unigram features.

3 Data Sets

We used three opinion-related data sets for our analyses and experiments: the *OP data set* created by (Wiebe et al., 2004), the *Polarity data set*[5] created by (Pang and Lee, 2004), and the *MPQA data set* created by (Wiebe et al., 2005).[6] The OP and Polarity data sets involve document-level opinion classification, while the MPQA data set involves sentence-level classification.

The OP data consists of 2,452 documents from the Penn Treebank (Marcus et al., 1993). Metadata tags assigned by the Wall Street Journal define the opinion/non-opinion classes: the class of any document labeled *Editorial, Letter to the Editor*, *Arts & Leisure Review*, or *Viewpoint* by the Wall Street Journal is *opinion*, and the class of documents in all other categories (such as *Business* and *News*) is *non-opinion*. This data set is highly skewed, with only 9% of the documents belonging to the opinion class. Consequently, a trivial (but useless) opinion classifier that labels all documents as non-opinion articles would achieve 91% accuracy.

The Polarity data consists of 700 positive and 700 negative reviews from the Internet Movie Database (IMDb) archive. The positive and negative classes were derived from author ratings expressed in stars or numerical values. The MPQA data consists of English language versions of articles from the world press. It contains 9,732 sentences that have been manually annotated for subjective expressions. The opinion/non-opinion classes are derived from the lower-level annotations: a sentence is an opinion if it contains a subjective expression of medium or higher intensity; otherwise, it is a non-opinion sentence. 55% of the sentences belong to the opinion class.

4 Using the Subsumption Hierarchy for Analysis

In this section, we illustrate how the subsumption hierarchy can be used as an analytic tool to automatically identify features that substantially outperform simpler counterparts. These features represent specialized usages and expressions that would be good candidates for addition to a subjectivity lexicon. Figure 6 shows pairs of features, where the first is more general and the second is more specific. These feature pairs were identified by the subsumption hierarchy as being representationally similar but behaviorally different (so the more specific feature was retained). The *IGain* column shows the information gain values produced from the training set of one cross-validation fold. The *Class* column shows the class that the more specific feature is correlated with (the more general feature is usually not strongly correlated with either class).

The top table in Figure 6 contains examples for the opinion/non-opinion classification task from

[5] Version v2.0, which is available at: http://www.cs.cornell.edu/people/pabo/movie-review-data/

[6] Available at http://www.cs.pitt.edu/mpqa/databaserelease/

Opinion/Non-Opinion Classification

ID	Feature	IGain	Class	Example
A_1	line	.0016	-	...issue consists of notes backed by credit *line* receivables
A_2	the line	.0075	opin	...lays it on *the line*; ...steps across *the line*
B_1	nation	.0046	-	...has 750,000 cable-tv subscribers around the *nation*
B_2	a nation	.0080	opin	It's not that we are spawning *a nation* of ascetics ...
C_1	begin	.0006	-	Campeau buyers will *begin* writing orders...
C_2	begin with	.0036	opin	To *begin with*, we should note that in contrast...
D_1	benefits	.0040	-	...earlier period included $235,000 in tax *benefits*.
D_{EP}	NP Prep(benefits to)	.0090	opin	...boon to the rich with no proven *benefits to* the economy
E_1	due	.0001	-	...an estimated $ 1.23 billion in debt *due* next spring
E_{EP}	ActVP Prep(due to)	.0038	opin	It's all *due to* the intense scrutiny...

Positive/Negative Sentiment Classification

ID	Feature	IGain	Class	Example
F_1	short	.0014	-	to make a long story *short*...
F_2	nothing short	.0039	pos	*nothing short* of spectacular
G_1	ugly	.0008	-	...an *ugly* monster on a cruise liner
G_2	and ugly	.0054	neg	it's a disappointment to see something this dumb *and ugly*
H_1	disaster	.0010	-	...rated pg-13 for *disaster* related elements
H_{EP}	AuxVP Dobj(be disaster)	.0048	neg	...this *is* such a confused *disaster* of a film
I_1	work	.0002	-	the next day during the drive to *work*...
I_{EP}	ActVP(work)	.0062	pos	the film *will work* just as well...
J_1	manages	.0003	-	he still *manages* to find time for his wife
J_{EP}	ActInfVP(manages to keep)	.0054	pos	this film *manages to keep* up a rapid pace

Figure 6: Sample features that behave differently, as revealed by the subsumption hierarchy. (1 ⇒ unigram; 2 ⇒ bigram; EP ⇒ extraction pattern)

the OP data. The more specific features are more strongly correlated with opinion articles. Surprisingly, simply adding a determiner can dramatically change behavior. Consider A_2. There are many subjective idioms involving *"the line"* (two are shown in the table; others include *"toe the line"* and *"draw the line"*), while objective language about credit lines, phone lines, etc. uses the determiner less often. Similarly, consider B_2. Adding *"a"* to *"nation"* often corresponds to an abstract reference used when making an argument (e.g., *"a nation of ascetics"*), whereas other instances of "nation" are used more literally (e.g., *"the 6th largest in the nation"*). 21% of feature B_1's instances appear in opinion articles, while 70% of feature B_2's instances are in opinion articles.

"Begin with" (C_2) captures an adverbial phrase used in argumentation (*"To begin with..."*) but does not match objective usages such as *"will begin"* an action. The word *"benefits"* alone (D_1) matches phrases like *"tax benefits"* and *"employee benefits"* that are not opinion expressions, while D_{EP} typically matches positive senses of the word *"benefits"*. Interestingly, the bigram *"benefits to"* is not highly correlated with opinions because it matches infinitive phrases such as *"tax benefits to provide"* and *"health benefits to cut"*. In this case, the extraction pattern "NP Prep(benefits to)" is more discriminating than the bigram for opinion classification. The extraction pattern E_{EP} is also highly correlated with opinions, while the unigram *"due"* and the bigram *"due to"* are not.

The bottom table in Figure 6 shows feature pairs identified for their behavioral differences on the Polarity data set, where the task is to distinguish positive reviews from negative reviews. F_2 and G_2 are bigrams that behave differently from their component unigrams. The expression *"nothing short (of)"* is typically used to express positive sentiments, while *"nothing"* and *"short"* by themselves are not. The word *"ugly"* is often used as a descriptive modifier that is not expressing a sentiment per se, while *"and ugly"* appears in predicate adjective constructions that are expressing a negative sentiment. The extraction pattern H_{EP} is more discriminatory than H_1 because it distinguishes negative sentiments (*"the film is a disaster!"*) from plot descriptions (*"the disaster movie..."*). I_{EP} shows that active-voice usages of "work" are strong positive indicators, while the unigram *"work"* appears in a variety of both positive and negative contexts. Finally, J_{EP} shows that the expression *"manages to keep"* is a strong positive indicator, while *"manages"* by itelf is much less discriminating.

These examples illustrate that the subsumption hierarchy can be a powerful tool to better understand the behaviors of different kinds of features, and to identify specific features that may be desirable for inclusion in specialized lexical resources.

5 Using the Subsumption Hierarchy to Reduce Feature Sets

When creating opinion classifiers, people often throw in a variety of features and trust the machine learning algorithm to figure out how to make the best use of them. However, we hypothesized that classifiers may perform better if we can proactively eliminate features that are not necesary because they are subsumed by other features. In this section, we present a series of experiments to explore this hypothesis. First, we present the results for an SVM classifier trained using different sets of unigram, bigram, and extraction pattern features, both before and after subsumption. Next, we evaluate a standard feature selection approach as an alternative to subsumption and then show that combining subsumption with standard feature selection produces the best results of all.

5.1 Classification Experiments

To see whether feature subsumption can improve classification performance, we trained an SVM classifier for each of the three opinion data sets. We used the SVM^{light} (Joachims, 1998) package with a linear kernel. For the Polarity and OP data we discarded all features that have frequency < 5, and for the MPQA data we discarded features that have frequency < 2 because this data set is substantially smaller. All of our experimental results are averages over 3-fold cross-validation.

First, we created 4 baseline classifiers: a *1Gram* classifier that uses only the unigram features; a *1+2Gram* classifier that uses unigram and bigram features; a *1+EP* classifier that uses unigram and extraction pattern features, and a *1+2+EP* classifier that uses all three types of features. Next, we created analogous 1+2Gram, 1+EP, and 1+2+EP classifiers but applied the subsumption hierarchy first to eliminate unnecessary features before training the classifier. We experimented with three delta values for the subsumption process: δ=.0005, .001, and .002.

Figures 7, 8, and 9 show the results. The subsumption process produced small but consistent improvements on all 3 data sets. For example, Figure 8 shows the results on the OP data, where all of the accuracy values produced after subsumption (the rightmost 3 columns) are higher than the accuracy values produced without subsumption (the Base[line] column). For all three data sets, the best overall accuracy (shown in boldface) was always achieved after subsumption.

Features	**Base**	**δ=.0005**	**δ=.001**	**δ=.002**
1Gram	79.8			
1+2Gram	81.2	81.0	81.3	81.0
1+EP	81.7	81.4	81.4	82.0
1+2+EP	81.7	82.3	82.3	**82.7**

Figure 7: Accuracies on Polarity Data

Features	**Base**	**δ=.0005**	**δ=.001**	**δ=.002**
1Gram	97.5	-	-	-
1+2Gram	98.0	**98.7**	98.6	**98.7**
1+EP	97.2	97.8	97.9	97.9
1+2+EP	97.8	98.6	**98.7**	**98.7**

Figure 8: Accuracies on OP Data

Features	**Base**	**δ=.0005**	**δ=.001**	**δ=.002**
1Gram	74.8			
1+2Gram	74.3	**74.9**	74.6	74.8
1+EP	74.4	74.6	74.6	74.6
1+2+EP	74.4	**74.9**	74.7	74.6

Figure 9: Accuracies on MPQA Data

We also observed that subsumption had a dramatic effect on the F-measure scores on the OP data, which are shown in Figure 10. The OP data set is fundamentally different from the other data sets because it is so highly skewed, with 91% of the documents belonging to the non-opinion class. Without subsumption, the classifier was conservative about assigning documents to the opinion class, achieving F-measure scores in the 82-88 range. After subsumption, the overall accuracy improved but the F-measure scores increased more dramatically. These numbers show that the subsumption process produced not only a more accurate classifier, but a more useful classifier that identifies more documents as being opinion articles.

For the MPQA data, we get a very small improvement of 0.1% (74.8% $\rightarrow$ 74.9%) using subsumption. But note that without subsumption the performance actually decreased when bigrams and

Features	Base	δ=.0005	δ=.001	δ=.002
1Gram	84.5			
1+2Gram	88.0	**92.5**	92.0	92.3
1+EP	82.4	86.9	87.4	87.4
1+2+EP	86.7	91.8	**92.5**	92.3

Figure 10: F-measures on OP Data

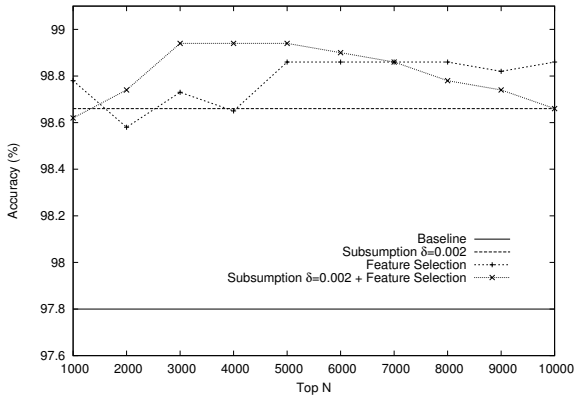

Figure 11: Feature Selection on OP Data

extraction patterns were added! The subsumption process counteracted the negative effect of adding the more complex features.

5.2 Feature Selection Experiments

We conducted a second series of experiments to determine whether a traditional feature selection approach would produce the same, or better, improvements as subsumption. For each feature, we computed its information gain (IG) and then selected the N features with the highest scores.[7] We experimented with values of N ranging from 1,000 to 10,000 in increments of 1,000.

We hypothesized that applying subsumption before traditional feature selection might also help to identify a more diverse set of high-performing features. In a parallel set of experiments, we explored this hypothesis by first applying subsumption to reduce the size of the feature set, and then selecting the best N features using information gain.

Figures 11, 12, and 13 show the results of these experiments for the 1+2+EP classifiers. Each graph shows four lines. One line corresponds to the baseline classifier with no subsumption, and another line corresponds to the baseline classifier with subsumption using the best δ value for that data set. Each of these two lines corresponds to

[7] In the case of ties, we included all features with the same score as the Nth-best as well.

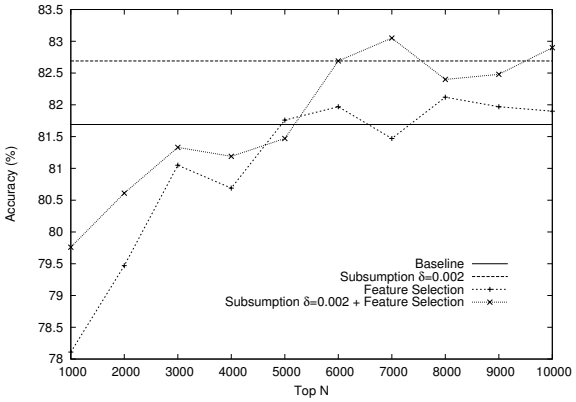

Figure 12: Feature Selection on Polarity Data

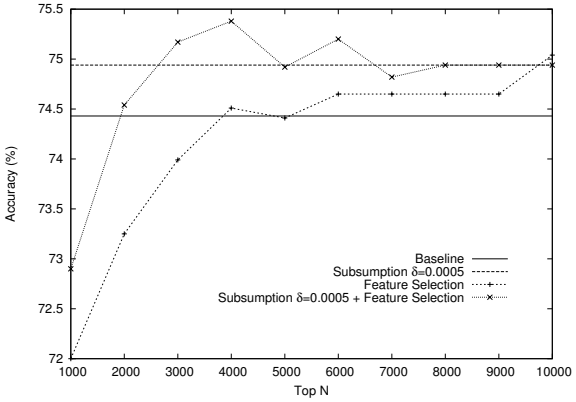

Figure 13: Feature Selection on MPQA Data

just a single data point (accuracy value), but we drew that value as a line across the graph for the sake of comparison. The other two lines on the graph correspond to (a) feature selection for different values of N (shown on the x-axis), and (b) subsumption followed by feature selection for different values of N.

On all 3 data sets, traditional feature selection performs worse than the baseline in some cases, and it virtually never outperforms the best classifier trained after subsumption (but without feature selection). Furthermore, the combination of subsumption plus feature selection generally performs best of all, and nearly always outperforms feature selection alone. For all 3 data sets, our best accuracy results were achieved by performing subsumption prior to feature selection. The best accuracy results are 99.0% on the OP data, 83.1% on the Polarity data, and 75.4% on the MPQA data. For the OP data, the improvement over baseline for both accuracy and F-measure are statistically significant at the $p < 0.05$ level (paired t-test). For the MPQA data, the improvement over baseline is

statistically significant at the $p < 0.10$ level.

6 Related Work

Many features and classification algorithms have been explored in sentiment analysis and opinion recognition. Lexical cues of differing complexities have been used, including single words and Ngrams (e.g., (Mullen and Collier, 2004; Pang et al., 2002; Turney, 2002; Yu and Hatzivassiloglou, 2003; Wiebe et al., 2004)), as well as phrases and lexico-syntactic patterns (e.g, (Kim and Hovy, 2004; Hu and Liu, 2004; Popescu and Etzioni, 2005; Riloff and Wiebe, 2003; Whitelaw et al., 2005)). While many of these studies investigate combinations of features and feature selection, this is the first work that uses the notion of subsumption to compare Ngrams and lexico-syntactic patterns to identify complex features that outperform simpler counterparts and to reduce a combined feature set to improve opinion classification.

7 Conclusions

This paper uses a *subsumption hierarchy* of feature representations as (1) an analytic tool to compare features of different complexities, and (2) an automatic tool to remove unnecessary features to improve opinion classification performance. Experiments with three opinion data sets showed that subsumption can improve classification accuracy, especially when combined with feature selection.

Acknowledgments

This research was supported by NSF Grants IIS-0208798 and IIS-0208985, the ARDA AQUAINT Program, and the Institute for Scientific Computing Research and the Center for Applied Scientific Computing within Lawrence Livermore National Laboratory.

References

S. Banerjee and T. Pedersen. 2003. The Design, Implementation, and Use of the Ngram Statistics Package. In *Proc. Fourth Int'l Conference on Intelligent Text Processing and Computational Linguistics*.

A. Esuli and F. Sebastiani. 2005. Determining the semantic orientation of terms through gloss analysis. In *Proc. CIKM-05*.

G. Forman. 2003. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *J. Mach. Learn. Res.*, 3:1289–1305.

M. Hu and B. Liu. 2004. Mining and summarizing customer reviews. In *Proc. KDD-04*.

T. Joachims. 1998. Making Large-Scale Support Vector Machine Learning Practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

S-M. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Proc. COLING-04*.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

T. Mullen and N. Collier. 2004. Sentiment Analysis Using Support Vector Machines with Diverse Information Sources. In *Proc. EMNLP-04*.

B. Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. ACL-04*.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proc. EMNLP-02*.

A-M. Popescu and O. Etzioni. 2005. Extracting product features and opinions from reviews. In *Proc. HLT-EMNLP-05*.

E. Riloff and W. Phillips. 2004. An Introduction to the Sundance and AutoSlog Systems. Technical Report UUCS-04-015, School of Computing, University of Utah.

E. Riloff and J. Wiebe. 2003. Learning Extraction Patterns for Subjective Expressions. In *Proc. EMNLP-03*.

E. Riloff. 1996. An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *Artificial Intelligence*, 85:101–134.

P. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proc. ACL-02*.

C. Whitelaw, N. Garg, and S. Argamon. 2005. Using appraisal groups for sentiment analysis. In *Proc. CIKM-05*.

J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin. 2004. Learning subjective language. *Computational Linguistics*, 30(3):277–308.

J. Wiebe, T. Wilson, and C. Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2/3).

H. Yu and V. Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proc. EMNLP-03*.