The viability of web-derived polarity lexicons

Leonid Velikovich **Sasha Blair-Goldensohn** **Kerry Hannan** **Ryan McDonald**
Google Inc., New York, NY
{leonidv|sasha|khannan|ryanmcd}@google.com

Abstract

We examine the viability of building large polarity lexicons semi-automatically from the web. We begin by describing a graph propagation framework inspired by previous work on constructing polarity lexicons from lexical graphs (Kim and Hovy, 2004; Hu and Liu, 2004; Esuli and Sabastiani, 2009; Blair-Goldensohn et al., 2008; Rao and Ravichandran, 2009). We then apply this technique to build an English lexicon that is significantly larger than those previously studied. Crucially, this web-derived lexicon does not require WordNet, part-of-speech taggers, or other language-dependent resources typical of sentiment analysis systems. As a result, the lexicon is not limited to specific word classes – e.g., adjectives that occur in WordNet – and in fact contains slang, misspellings, multiword expressions, etc. We evaluate a lexicon derived from English documents, both qualitatively and quantitatively, and show that it provides superior performance to previously studied lexicons, including one derived from WordNet.

1 Introduction

Polarity lexicons are large lists of phrases that encode the polarity of each phrase within it – either positive or negative – often with some score representing the magnitude of the polarity (Hatzivassiloglou and McKeown, 1997; Wiebe, 2000; Turney, 2002). Though classifiers built with machine learning algorithms have become commonplace in the sentiment analysis literature, e.g., Pang et al. (2002), the core of many academic and commercial sentiment analysis systems remains the polarity lexicon, which can be constructed manually (Das and Chen, 2007), through heuristics (Kim and Hovy, 2004; Esuli and Sabastiani, 2009) or using machine learning (Turney, 2002; Rao and Ravichandran, 2009). Often lexicons are combined with machine learning for improved results (Wilson et al., 2005). The pervasiveness and sustained use of lexicons can be ascribed to a number of reasons, including their interpretability in large-scale systems as well as the granularity of their analysis.

In this work we investigate the viability of polarity lexicons that are derived solely from unlabeled web documents. We propose a method based on graph propagation algorithms inspired by previous work on constructing polarity lexicons from lexical graphs (Kim and Hovy, 2004; Hu and Liu, 2004; Esuli and Sabastiani, 2009; Blair-Goldensohn et al., 2008; Rao and Ravichandran, 2009). Whereas past efforts have used linguistic resources – e.g., WordNet – to construct the lexical graph over which propagation runs, our lexicons are constructed using a graph built from co-occurrence statistics from the entire web. Thus, the method we investigate can be seen as a combination of methods for propagating sentiment across lexical graphs and methods for building sentiment lexicons based on distributional characteristics of phrases in raw data (Turney, 2002). The advantage of breaking the dependence on WordNet (or related resources like thesauri (Mohammad et al., 2009)) is that it allows the lexicons to include non-standard entries, most notably spelling mistakes and variations, slang, and multiword expressions.

The primary goal of our study is to understand the characteristics and practical usefulness of such a lexicon. Towards this end, we provide both a qualitative and quantitative analysis for a web-derived English

lexicon relative to two previously published lexicons – the lexicon used in Wilson et al. (2005) and the lexicon used in Blair-Goldensohn et al. (2008). Our experiments show that a web-derived lexicon is not only significantly larger, but has improved accuracy on a sentence polarity classification task, which is an important problem in many sentiment analysis applications, including sentiment aggregation and summarization (Hu and Liu, 2004; Carenini et al., 2006; Lerman et al., 2009). These results hold true both when the lexicons are used in conjunction with string matching to classify sentences, and when they are included within a contextual classifier framework (Wilson et al., 2005).

Extracting polarity lexicons from the web has been investigated previously by Kaji and Kitsuregawa (2007), who study the problem exclusively for Japanese. In that work a set of positive/negative sentences are first extracted from the web using cues from a syntactic parser as well as the document structure. Adjectives phrases are then extracted from these sentences based on different statistics of their occurrence in the positive or negative set. Our work, on the other hand, does not rely on syntactic parsers or restrict the set of candidate lexicon entries to specific syntactic classes, i.e., adjective phrases. As a result, the lexicon built in our study is on a different scale than that examined in Kaji and Kitsuregawa (2007). Though this hypothesis is not tested here, it also makes our techniques more amenable to adaptation for other languages.

2 Constructing the Lexicon

In this section we describe a method to construct polarity lexicons using graph propagation over a phrase similarity graph constructed from the web.

2.1 Graph Propagation Algorithm

We construct our lexicon using graph propagation techniques, which have previously been investigated in the construction of polarity lexicons (Kim and Hovy, 2004; Hu and Liu, 2004; Esuli and Sabastiani, 2009; Blair-Goldensohn et al., 2008; Rao and Ravichandran, 2009). We assume as input an undirected edge weighted graph $G = (V, E)$, where $w_{ij} \in [0, 1]$ is the weight of edge $(v_i, v_j) \in E$. The node set V is the set of candidate phrases for inclusion in a sentiment lexicon. In practice, G should encode semantic similarities between two nodes, e.g., for sentiment analysis one would hope that $w_{ij} > w_{ik}$ if v_i=*good*, v_j=*great* and v_k=*bad*. We also assume as input two sets of *seed phrases*, denoted P for the positive seed set and N for the negative seed set. The common property among all graph propagation algorithms is that they attempt to propagate information from the seed sets to the rest of the graph through its edges. This can be done using machine learning, graph algorithms or more heuristic means.

The specific algorithm used in this study is given in Figure 1, which is distinct from common graph propagation algorithms, e.g., label propagation (see Section 2.3). The output is a polarity vector $\mathbf{pol} \in \mathbb{R}^{|V|}$ such that $\mathbf{pol}_i$ is the polarity score for the i^{th} candidate phrase (or the i^{th} node in G). In particular, we desire $\mathbf{pol}$ to have the following semantics:

$$\mathbf{pol}_i = \begin{cases} > 0 & i^{th} \text{ phrase has positive polarity} \\ < 0 & i^{th} \text{ phrase has negative polarity} \\ = 0 & i^{th} \text{ phrase has no sentiment} \end{cases}$$

Intuitively, the algorithm works by computing both a positive and a negative polarity magnitude for each node in the graph, call them $\mathbf{pol}_i^+$ and $\mathbf{pol}_i^-$. These values are equal to the sum over the max weighted path from every seed word (either positive or negative) to node v_i. Phrases that are connected to multiple positive seed words through short yet highly weighted paths will receive high positive values. The final polarity of a phrase is then set to $\mathbf{pol}_i = \mathbf{pol}_i^+ - \beta\mathbf{pol}_i^-$, where β a constant meant to account for the difference in overall mass of positive and negative flow in the graph. Thus, after the algorithm is run, if a phrase has a higher positive than negative polarity score, then its final polarity will be positive, and negative otherwise.

There are some implementation details worth pointing out. First, the algorithm in Figure 1 is written in an iterative framework, where on each iteration, paths of increasing lengths are considered. The input variable T controls the max path length considered by the algorithm. This can be set to be a small value in practice, since the multiplicative path weights result in long paths rarely contributing to polarity scores. Second, the parameter γ is a threshold that defines the minimum polarity magnitude a

Input: $G = (V, E)$, $w_{ij} \in [0, 1]$,
P, N, $\gamma \in \mathbb{R}$, $T \in \mathbb{N}$
Output: $\mathbf{pol} \in \mathbb{R}^{|V|}$
Initialize: $\mathbf{pol}_i, \mathbf{pol}_i^+, \mathbf{pol}_i^- = 0$, for all i
$\mathbf{pol}_i^+ = 1.0$ for all $v_i \in P$ and
$\mathbf{pol}_i^- = 1.0$ for all $v_i \in N$

1. set $\alpha_{ii} = 1$, and $\alpha_{ij} = 0$ for all $i \neq j$
2. for $v_i \in P$
3. $F = \{v_i\}$
4. for $t : 1 \ldots T$
5. for $(v_k, v_j) \in E$ such that $v_k \in F$
6. $\alpha_{ij} = \max\{\alpha_{ij},\ \alpha_{ik} \cdot w_{kj}\}$
$F = F \cup \{v_j\}$
7. for $v_j \in V$
8. $\mathbf{pol}_j^+ = \sum_{v_i \in P} \alpha_{ij}$
9. Repeat steps 1-8 using N to compute $\mathbf{pol}^-$
10. $\beta = \sum_i \mathbf{pol}_i^+ / \sum_i \mathbf{pol}_i^-$
11. $\mathbf{pol}_i = \mathbf{pol}_i^+ - \beta \mathbf{pol}_i^-$, for all i
12. if $|\mathbf{pol}_i| < \gamma$ then $\mathbf{pol}_i = 0.0$, for all i

Figure 1: Graph Propagation Algorithm.

phrase must have to be included in the lexicon. Both T and γ were tuned on held-out data.

To construct the final lexicon, the remaining nodes – those with polarity scores above γ – are extracted and assigned their corresponding polarity.

2.2 Building a Phrase Graph from the Web

Graph propagation algorithms rely on the existence of graphs that encode meaningful relationships between candidate nodes. Past studies on building polarity lexicons have used linguistic resources like WordNet to define the graph through synonym and antonym relations (Kim and Hovy, 2004; Esuli and Sabastiani, 2009; Blair-Goldensohn et al., 2008; Rao and Ravichandran, 2009). The goal of this study is to examine the size and quality of polarity lexicons when the graph is induced automatically from documents on the web.

Constructing a graph from web-computed lexical co-occurrence statistics is a difficult challenge in and of itself and the research and implementation hurdles that arise are beyond the scope of this work (Alfonseca et al., 2009; Pantel et al., 2009). For this study, we used an English graph where the node set V was based on all n-grams up to length 10 extracted from 4 billion web pages. This list was filtered to 20 million candidate phrases using a number of heuristics including frequency and mutual information of word boundaries. A context vector for each candidate phrase was then constructed based on a window of size six aggregated over all mentions of the phrase in the 4 billion documents. The edge set E was constructed by first, for each potential edge (v_i, v_j), computing the cosine similarity value between context vectors. All edges (v_i, v_j) were then discarded if they were not one of the 25 highest weighted edges adjacent to either node v_i or v_j. This serves to both reduce the size of the graph and to eliminate many spurious edges for frequently occurring phrases, while still keeping the graph relatively connected. The weight of the remaining edges was set to the corresponding cosine similarity value.

Since this graph encodes co-occurrences over a large, but local context window, it can be noisy for our purposes. In particular, we might see a number of edges between positive and negative sentiment words as well as sentiment words and non-sentiment words, e.g., sentiment adjectives and all other adjectives that are distributionally similar. Larger windows theoretically alleviate this problem as they encode semantic as opposed to syntactic similarities. We note, however, that the graph propagation algorithm described above calculates the sentiment of each phrase as the aggregate of all the best paths to seed words. Thus, even if some local edges are erroneous in the graph, one hopes that, globally, positive phrases will be influenced more by paths from positive seed words as opposed to negative seed words. Section 3, and indeed this paper, aims to measure whether this is true or not.

2.3 Why Not Label Propagation?

Previous studies on constructing polarity lexicons from lexical graphs, e.g., Rao and Ravichandran (2009), have used the label propagation algorithm, which takes the form in Figure 2 (Zhu and Ghahramani, 2002). Label propagation is an iterative algorithm where each node takes on the weighted average of its neighbour's values from the previous iteration. The result is that nodes with many paths to seeds get high polarities due to the influence from their neighbours. The label propagation algorithm is known to have many desirable properties including convergence, a well defined objective function

Input: $G = (V, E)$, $w_{ij} \in [0, 1]$, P, N
Output: $\mathbf{pol} \in \mathbb{R}^{|V|}$
Initialize: $\mathbf{pol}_i = 1.0$ for all $v_i \in P$ and
$\mathbf{pol}_i = -1.0$ for all $v_i \in N$ and
$\mathbf{pol}_i = 0.0 \; \forall v_i \notin P \cup N$

1. for : t .. T
2. $\mathbf{pol}_i = \frac{\sum_{(v_i,v_j)\in E} w_{ij} \times \mathbf{pol}_j}{\sum_{(v_i,v_j)} w_{ij}}, \forall v_i \in V$
3. reset $\mathbf{pol}_i = 1.0 \; \forall v_i \in P$
   reset $\mathbf{pol}_i = -1.0 \; \forall v_i \in N$

Figure 2: The label propagation algorithm (Zhu and Ghahramani, 2002).

(minimize squared error between values of adjacent nodes), and an equivalence to computing random walks through graphs.

The primary difference between standard label propagation and the graph propagation algorithm given in Section 2.1, is that a node with multiple paths to a seed will be influenced by all these paths in the label propagation algorithm, whereas only the single path from a seed will influence the polarity of a node in our proposed propagation algorithm – namely the path with highest weight. The intuition behind label propagation seems justified. That is, if a node has multiple paths to a seed, it should be reflected in a higher score. This is certainly true when the graph is of high quality and all paths trustworthy. However, in a graph constructed from web co-occurrence statistics, this is rarely the case.

Our graph consisted of many dense subgraphs, each representing some semantic entity class, such as actors, authors, tech companies, etc. Problems arose when polarity flowed into these dense subgraphs with the label propagation algorithm. Ultimately, this flow would amplify since the dense subgraph provided exponentially many paths from each node to the source of the flow, which caused a reinforcement effect. As a result, the lexicon would consist of large groups of actor names, companies, etc. This also led to convergence issues since the polarity is divided proportional to the size of the dense subgraph. Additionally, negative phrases in the graph appeared to be in more densely connected regions, which resulted in the final lexicons being highly skewed towards negative entries due to the influence of multiple paths to seed words.

For best path propagation, these problems were less acute as each node in the dense subgraph would only get the polarity a single time from each seed, which is decayed by the fact that edge weights are smaller than 1. Furthermore, the fact that edge weights are less than 1 results in most long paths having weights near zero, which in turn results in fast convergence.

3 Lexicon Evaluation

We ran the best path graph propagation algorithm over a graph constructed from the web using manually constructed positive and negative seed sets of 187 and 192 words in size, respectively. These words were generated by a set of five humans and many are morphological variants of the same root, e.g., excel/excels/excelled. The algorithm produced a lexicon that contained 178,104 entries. Depending on the threshold γ (see Figure 1), this lexicon could be larger or smaller. As stated earlier, our selection of γ and all hyperparameters was based on manual inspection of the resulting lexicons and performance on held-out data.

In the rest of this section we investigate the properties of this lexicon to understand both its general characteristics as well as its possible utility in sentiment applications. To this end we compare three different lexicons:

1. **Wilson et al.**: Described in Wilson et al. (2005). Lexicon constructed by combining the lexicon built in Riloff and Wiebe (2003) with other sources[1]. Entries are are coarsely rated – strong/weak positive/negative – which we weighted as 1.0, 0.5, -0.5, and -1.0 for our experiments.

2. **WordNet LP**: Described in Blair-Goldensohn et al. (2008). Constructed using label propagation over a graph derived from WordNet synonym and antonym links. Note that label propagation is not prone to the kinds of errors discussed in Section 2.3 since the lexical graph is derived from a high quality source.

3. **Web GP**: The web-derived lexicon described in Section 2.1 and Section 2.2.

[1] See http://www.cs.pitt.edu/mpqa/

3.1 Qualitative Evaluation

Table 1 breaks down the lexicon by the number of positive and negative entries of each lexicon, which clearly shows that the lexicon derived from the web is more than an order of magnitude larger than previously constructed lexicons.[2] This in and of itself is not much of an achievement if the additional phrases are of poor quality. However, in Section 3.2 we present an empirical evaluation that suggests that these terms provide both additional *and* useful information. Table 1 also shows the recall of the each lexicon relative to the other. Whereas the Wilson et al. (2005) and WordNet lexicon have a recall of only 3% relative to the web lexicon, the web lexicon has a recall of 48% and 70% relative to the two other lexicons, indicating that it contains a significant amount of information from the other lexicons. However, this overlap is still small, suggesting that a combination of all the lexicons could provide the best performance. In Section 3.2 we investigate this empirically through a meta classification system.

Table 2 shows the distribution of phrases in the web-derived lexicon relative to the number of tokens in each phrase. Here a token is simply defined by whitespace and punctuation, with punctuation counting as a token, e.g., "half-baked" is counted as 3 tokens. For the most part, we see what one might expect, as the number of tokens increases, the number of corresponding phrases in the lexicon also decreases. Longer phrases are less frequent and thus will have both fewer and lower weighted edges to adjacent nodes in the graph. There is a single phrase of length 9, which is "motion to dismiss for failure to state a claim". In fact, the lexicon contains quite a number of legal and medical phrases. This should not be surprising, since in a graph induced from the web, a phrase like "cancer" (or any disease) should be distributionally similar to phrases like "illness", "sick", and "death", which themselves will be similar to standard sentiment phrases like "bad" and "terrible". These terms are predominantly negative in the lexicon representing the broad notion that legal and medical events are undesirable.

[2]This also includes the web-derived lexicon of (Kaji and Kitsuregawa, 2007), which has 10K entries. A recent study by Mohammad et al. (2009) generated lexicons from thesauri with 76K entries.

Phrase length	1	2	3
# of phrases	37,449	108,631	27,822

Phrase length	4	5	6	7	8	9
# of phrases	3,489	598	71	29	4	1

Table 2: Number of phrases by phrase length in lexicon built from the web.

Perhaps the most interesting characteristic of the lexicon is that the most frequent phrase length is 2 and not 1. The primary reason for this is an abundance of adjective phrases consisting of an adverb and an adjective, such as "more brittle" and "less brittle". Almost every adjective of length 1 is frequently combined in such a way on the web, so it not surprising that we see many of these phrases in the lexicon. Ideally we would see an order on such phrases, e.g., "more brittle" has a larger negative polarity than "brittle", which in turn has a larger negative polarity than "less brittle". However, this is rarely the case and usually the adjective has the highest polarity magnitude. Again, this is easily explained. These phrases are necessarily more common and will thus have more edges with larger weights in the graph and thus a greater chance of accumulating a high sentiment score. The prominence of such phrases suggests that a more principled treatment of them should be investigated in the future.

Finally, Table 3 presents a selection of phrases from both the positive and negative lexicons categorized into revealing verticals. For both positive and negative phrases we present typical examples of phrases – usually adjectives – that one would expect to be in a sentiment lexicon. These are phrases not included in the seed sets. We also present multiword phrases for both positive and negative cases, which displays concretely the advantage of building lexicons from the web as opposed to using restricted linguistic resources such as WordNet. Finally, we show two special cases. The first is spelling variations (and mistakes) for positive phrases, which were far more prominent than for negative phrases. Many of these correspond to social media text where one expresses an increased level of sentiment by repeating characters. The second is vulgarity in negative phrases, which was far more prominent than for positive phrases. Some of these are clearly appropri-

	All Phrases	Pos. Phrases	Neg. Phrases	Recall wrt other lexicons		
				Wilson et al.	WordNet LP	Web GP
Wilson et al.	7,628	2,718	4,910	100%	37%	2%
WordNet LP	12,310	5,705	6,605	21%	100%	3%
Web GP	178,104	90,337	87,767	70%	48%	100%

Table 1: Lexicon statistics. Wilson et al. is the lexicon used in Wilson et al. (2005), WordNet LP is the lexicon constructed by Blair-Goldensohn et al. (2008) that uses label propagation algorithms over a graph constructed through WordNet, and Web GP is the web-derived lexicon from this study.

POSITIVE PHRASES			**NEGATIVE PHRASES**		
Typical	**Multiword expressions**	**Spelling variations**	**Typical**	**Multiword expressions**	**Vulgarity**
cute	once in a life time	loveable	dirty	run of the mill	fucking stupid
fabulous	state - of - the - art	nicee	repulsive	out of touch	fucked up
cuddly	fail - safe operation	niice	crappy	over the hill	complete bullshit
plucky	just what the doctor ordered	cooool	sucky	flash in the pan	shitty
ravishing	out of this world	coooool	subpar	bumps in the road	half assed
spunky	top of the line	koool	horrendous	foaming at the mouth	jackass
enchanting	melt in your mouth	kewl	miserable	dime a dozen	piece of shit
precious	snug as a bug	cozy	lousy	pie - in - the - sky	son of a bitch
charming	out of the box	cosy	abysmal	sick to my stomach	sonofabitch
stupendous	more good than bad	sikk	wretched	pain in my ass	sonuvabitch

Table 3: Example positive and negative phrases from web lexicon.

ate, e.g., "shitty", but some are clearly insults and outbursts that are most likely included due to their co-occurrence with angry texts. There were also a number of derogatory terms and racial slurs in the lexicon, again most of which received negative sentiment due to their typical disparaging usage.

3.2 Quantitative Evaluation

To determine the practical usefulness of a polarity lexicon derived from the web, we measured the performance of the lexicon on a sentence classification/ranking task. The input is a set of sentences and the output is a classification of the sentences as being either positive, negative or neutral in sentiment. Additionally, the system outputs two rankings, the first a ranking of the sentence by positive polarity and the second a ranking of the sentence by negative polarity. Classifying sentences by their sentiment is a subtask of sentiment aggregation systems (Hu and Liu, 2004; Gamon et al., 2005). Ranking sentences by their polarity is a critical sub-task in extractive sentiment summarization (Carenini et al., 2006; Lerman et al., 2009).

To classify sentences as being positive, negative or neutral, we used an augmented vote-flip algorithm (Choi and Cardie, 2009), which is given in Figure 3. This intuition behind this algorithm is simple. The number of matched positive and negative phrases from the lexicon are counted and whichever has the most votes wins. The algorithm flips the decision if the number of negations is odd. Though this algorithm appears crude, it benefits from not relying on threshold values for neutral classification, which is difficult due to the fact that the polarity scores in the three lexicons are not on the same scale.

To rank sentences we defined the purity of a sentence X as the normalized sum of the sentiment scores for each phrase x in the sentence:

$$\mathsf{purity}(\mathsf{X}) = \frac{\sum_{\mathsf{x}\in\mathsf{X}} \mathbf{pol}_{\mathsf{x}}}{\delta + \sum_{\mathsf{x}\in\mathsf{X}} |\mathbf{pol}_{\mathsf{x}}|}$$

This is a normalized score in the range $[-1, 1]$. Intuitively, sentences with many terms of the same polarity will have purity scores at the extreme points of the range. Before calculating purity, a simple negation heuristic was implemented that reversed the sentiment scores of terms that were within the scope of negations. The term δ helps to favor sentences with multiple phrase matches. Purity is a common metric used for ranking sentences for inclusion in sentiment summaries (Lerman et al., 2009). Purity and negative purity were used to rank sentences as being positive and negative sentiment, respectively.

The data used in our initial English-only experi-

	Lexicon Classifier						Contextual Classifier					
	Positive			Negative			Positive			Negative		
	P	R	AP	P	R	AP	P	R	AP	P	R	AP
Wilson et al.	56.4	61.8	60.8	58.1	39.0	59.7	74.5	70.3	76.2	80.7	70.1	81.2
WordNet LP	50.9	61.7	62.0	54.9	36.4	59.7	72.0	72.5	75.7	78.0	69.8	79.3
Web GP	57.7	65.1†	69.6†	60.3	42.9	68.5†	74.1	75.0†	79.9†	80.5	72.6†	82.9†
Meta Classifier	-	-	-	-	-	-	76.6‡	74.7	81.2‡	81.8‡	72.2	84.1‡

Table 4: Positive and negative precision (P), recall (R), and average precision (AP) for three lexicons using either lexical matching or contextual classification strategies. † Web GP is statistically significantly better than Wilson et al. and WordNet LP ($p < 0.05$). ‡ Meta Classifier is statistically significantly better than all other systems ($p < 0.05$).

Input: Scored lexicon $\mathbf{pol}$, negation list NG, input sentence X
Output: sentiment $\in \{$POS, NEG, NEU$\}$

1. set $p, n, ng = 0$
2. for $x \in X$
3. if $\mathbf{pol}_x > 0$ then p++
4. else if $\mathbf{pol}_x < 0$ then n++
5. else if $x \in$ NG then ng++
6. flip = $(ng \,\%\, 2 == 1)$ //ng is odd
7. if $(p > n$ & ¬flip$)$ ‖ $(n > p$ & flip$)$
return POS
8. else if $(p > n$ & flip$)$ ‖ $(n > p$ & ¬flip$)$
return NEG
19. return NEU

Figure 3: Vote-flip algorithm (Choi and Cardie, 2009).

ments were a set of 554 consumer reviews described in (McDonald et al., 2007). Each review was sentence split and annotated by a human as being positive, negative or neutral in sentiment. This resulted in 3,916 sentences, with 1,525, 1,542 and 849 positive, negative and neutral sentences, respectively.

The first six columns of Table 4 shows: 1) the positive/negative precision-recall of each lexicon-based system where sentence classes were determined using the vote-flip algorithm, and 2) the average precision for each lexicon-based system where purity (or negative purity) was used to rank sentences. Both the Wilson et al. and WordNet LP lexicons perform at a similar level, with the former slightly better, especially in terms of precision. The web-derived lexicon, Web GP, outperforms the other two lexicons across the board, in particular when looking at average precision, where the gains are near 10% absolute. If we plot the precision-recall graphs using purity to classify sentences – as opposed to the vote-flip algorithm, which only provides an unweighted classification – we can see that at almost all recall levels the web-derived lexicon has superior precision to the other lexicons (Figure 4). Thus, even though the web-derived lexicon is constructed from a lexical graph that contains noise, the graph propagation algorithms appear to be fairly robust to this noise and are capable of producing large and accurate polarity lexicons.

The second six columns of Table 4 shows the performance of each lexicon as the core of a contextual classifier (Wilson et al., 2005). A contextual classifier is a machine learned classifier that predicts the polarity of a sentence using features of that sentence and its context. For our experiments, this was a maximum entropy classifier trained and evaluated using 10-fold cross-validation on the evaluation data. The features included in the classifier were the purity score, the number of positive and negative lexicon matches, and the number of negations in the sentence, as well as concatenations of these features within the sentence and with the same features derived from the sentences in a window of size 1.

For each sentence, the contextual classifier predicted either a positive, negative or neutral classification based on the label with highest probability. Additionally, all sentences were placed in the positive and negative sentence rankings by the probability the classifier assigned to the positive and negative classes, respectively. Mirroring the results of Wilson et al. (2005), we see that contextual classifiers improve results substantially over lexical matching. More interestingly, we see that the a contextual classifier over the web-derived lexicons maintains the performance edge over the other lexicons, though the gap is smaller. Figure 5 plots the precision-recall curves for the positive and negative sentence rank-

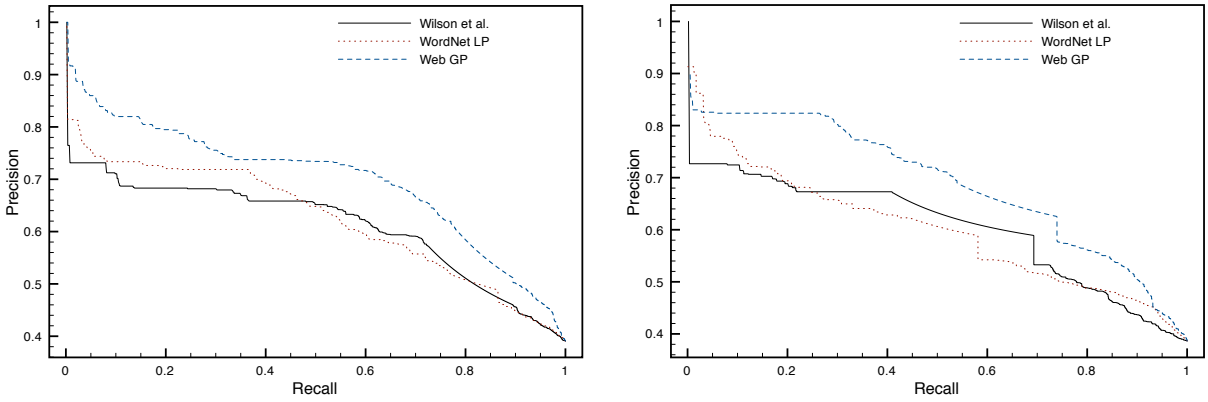

Figure 4: Lexicon classifier precision/recall curves for positive (left) and negative (right) classes.

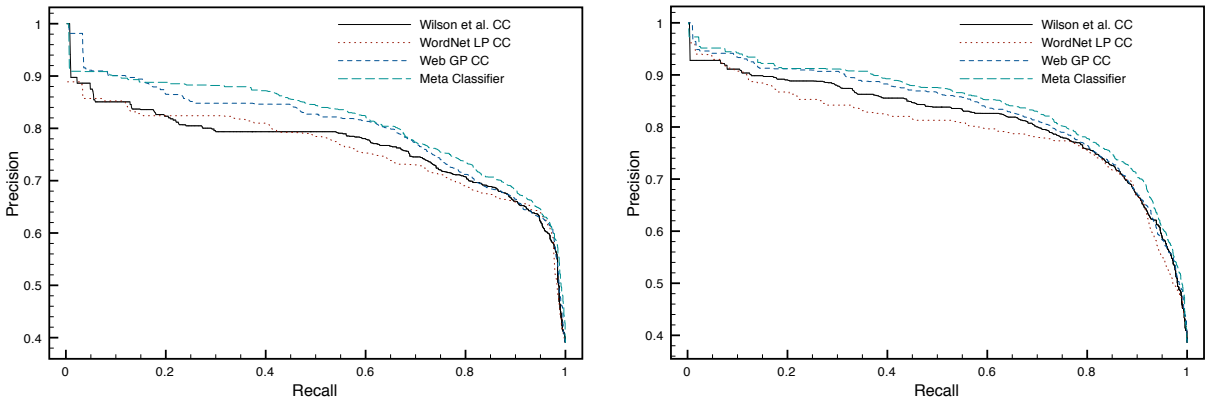

Figure 5: Contextual classifier precision/recall curves for positive (left) and negative (right) classes

ings, again showing that at almost every level of recall, the web-derived lexicon has higher precision.

For a final English experiment we built a meta-classification system that is identical to the contextual classifiers, except it is trained using features derived from all lexicons. Results are shown in the last row of Table 4 and precision-recall curves are shown in Figure 5. Not surprisingly, this system has the best performance in terms of average precision as it has access to the largest amount of information, though its performance is only slightly better than the contextual classifier for the web-derived lexicon.

4 Conclusions

In this paper we examined the viability of sentiment lexicons learned semi-automatically from the web, as opposed to those that rely on manual annotation and/or resources such as WordNet. Our qualitative experiments indicate that the web derived lexicon can include a wide range of phrases that have not been available to previous systems, most notably spelling variations, slang, vulgarity, and multi-word expressions. Quantitatively, we observed that the web derived lexicon had superior performance to previously published lexicons for English classification. Ultimately, a meta classifier that incorporates features from all lexicons provides the best performance. In the future we plan to investigate the construction of web-derived lexicons for languages other than English, which is an active area of research (Mihalcea et al., 2007; Jijkoun and Hofmann, 2009; Rao and Ravichandran, 2009). The advantage of the web-derived lexicons studied here is that they do not rely on language specific resources besides unlabeled data and seed lists. A primary question is whether such lexicons improve performance over a translate-to-English strategy (Banea et al., 2008).

Acknowledgements: The authors thank Andrew Hogue, Raj Krishnan and Deepak Ravichandran for insightful discussions about this work.

References

E. Alfonseca, K. Hall, and S. Hartmann. 2009. Large-scale computation of distributional similarities for queries. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

C. Banea, R. Mihalcea, J. Wiebe, and S. Hassan. 2008. Multilingual subjectivity analysis using machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G.A. Reis, and J. Reynar. 2008. Building a sentiment summarizer for local service reviews. In *NLP in the Information Explosion Era*.

G. Carenini, R. Ng, and A. Pauls. 2006. Multi-document summarization of evaluative text. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Y. Choi and C. Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

S.R. Das and M.Y. Chen. 2007. Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388.

A Esuli and F. Sabastiani. 2009. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the Language Resource and Evaluation Conference (LREC)*.

M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. 2005. Pulse: Mining customer opinions from free text. In *Proceedings of the 6th International Symposium on Intelligent Data Analysis (IDA)*.

V. Hatzivassiloglou and K.R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

M. Hu and B. Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*.

V.B. Jijkoun and K. Hofmann. 2009. Generating a non-english subjectivity lexicon: Relations that matter. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

N. Kaji and M. Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

S.M. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. 2009. Sentiment summarization: Evaluating and learning user preferences. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

R. Mihalcea, C. Banea, and J. Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

S. Mohammad, B. Dorr, and C. Dunne. 2009. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

P. Pantel, E. Crestan, A. Borkovsky, A. Popescu, and V. Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

D. Rao and D. Ravichandran. 2009. Semi-Supervised Polarity Lexicon Induction. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

E. Riloff and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

P. Turney. 2002. Thumbs up or thumbs down? Sentiment orientation applied to unsupervised classification of reviews. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

J. Wiebe. 2000. Learning subjective adjectives from corpora. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

X. Zhu and Z. Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical report, CMU CALD tech report CMU-CALD-02.