Towards Debugging Sentiment Lexicons

Andrew Schneider
Computer and Information Sciences
Temple University
atschneider@temple.edu

Eduard Dragut
Computer and Information Sciences
Temple University
edragut@temple.edu

Abstract

Central to many sentiment analysis tasks are sentiment lexicons (SLs). SLs exhibit polarity inconsistencies. Previous work studied the problem of checking the consistency of an SL for the case when the entries have categorical labels (positive, negative or neutral) and showed that it is NP-hard. In this paper, we address the more general problem, in which polarity tags take the form of a continuous distribution in the interval [0, 1]. We show that this problem is polynomial. We develop a general framework for addressing the consistency problem using linear programming (LP) theory. LP tools allow us to uncover inconsistencies efficiently, paving the way to building SL debugging tools. We show that previous work corresponds to 0-1 integer programming, a particular case of LP. Our experimental studies show a strong correlation between polarity consistency in SLs and the accuracy of sentiment tagging in practice.

1 Introduction

Many sentiment analysis algorithms rely on *sentiment lexicons* (SLs), where word forms or word senses[1] are tagged as conveying positive, negative or neutral sentiments. SLs are constructed by one of three methods (Liu, 2012; Feldman, 2013): (1) **Manual** tagging by human annotators is generally reliable, but because it is labor-intensive, slow, and costly, this method has produced small-sized SLs comprising a few thousand words, e.g., Opinion Finder (OF) (Wilson et al., 2005), Appraisal Lexicon (AL) (Taboada and Grieve, 2004), General Inquirer (GI) (Stone et al., 1966), and Micro-WNOp (Cerini et al., 2007). (2) **Dictionary-based** acquisition relies on a set of seed words to expand its coverage to similar words. There are over thirty dictionary-based techniques (Andreevskaia and Bergler, 2006; Blum et al., 2004; Chen and Skiena, 2014; Choi and Wiebe, 2014; Esuli and Sebastiani, 2006; Feng et al., 2013; Hassan and Radev, 2010; Kamps et al., 2004; Mohammad et al., 2009; Takamura et al., 2005; Turney, 2002; Williams and Anand, 2009), most of them based on WordNet (Fellbaum, 1998), such as SentiWordNet (SWN)(Baccianella et al., 2010) and Q-WordNet (QWN) (Agerri and García-Serrano, 2010). (3) **Corpus-based** acquisition expands a set of seed words with the use of a large document corpus (Breck et al., 2007; Bross and Ehrig, 2013; Choi and Cardie, 2009; Ding et al., 2008; Du et al., 2010; Hatzivassiloglou and McKeown, 1997; Jijkoun et al., 2010; Kaji and Kitsuregawa, 2007; Klebanov et al., 2013; Lu et al., 2011; Peng and Park, 2011; Tang et al., 2014; Wu and Wen, 2010). Method (1) generally produces the most reliable annotations, however the considerable effort required to yield substantial lexicons makes it less useful in practice. The appeals of (2) and (3) lie in the formalism of their models and their capability of producing large-sized SLs. SLs are either word or sense/synset oriented. We refer to the former as Sentiment Word Lexicons (SWLs), e.g., GI, OF, and AL, and to the latter as Sentiment Sense Lexions (SSLs), e.g., SWN, QWN, and Micro-WNOp. Besides the method of compilation, SLs may also vary with regard to sentiment annotation.

Polarity disagreements are noted across SLs that do (SWN, Q-WordNet) and do not (AL, GI) reference WordNet. For instance, the adjectives `panicky` and `terrified`, have negative and positive polarities in OF, respectively. They each have only one synset which they share in WordNet: *"thrown into a state of intense fear or desperation"*. Assuming that there is an intrinsic re-

[1] We refer to a string of letters or sounds as a word form & to a pairing of a word form with a meaning as a word sense.

lationship between the sentiments of a word and its meanings, a single synset polarity assignment to this synset cannot agree with both *positive* and *negative* at the word level. If the information given in WordNet is accurate (the Oxford and Cambridge dictionaries give only this meaning for both words) then there must be an annotation inconsistency in OF, called a *polarity inconsistency*. While some inconsistencies are easy to detect, manual consistency checking of an entire SL is an impractical endeavor, primarily because of the sheer size (SWN has over 206,000 word-sense pairs). Additionally, WordNet's complex network structure renders manual checking virtually impossible; an instance of a polarity inconsistency may entail an entire sub-network of words and senses. In this paper we develop a rigorous formal method based on linear programming (LP)(Schrijver, 1986) for *polarity consistency checking* of SLs with accompanying methods to unearth mislabeled words and synsets when consistency is not satisfied.

We translate the *polarity consistency problem* (PCP) into a form of the LP problem, suitable as the input to a standard LP solver, and utilize the functionality available in modern LP software (e.g., identifying an irreducible infeasible subset) to pinpoint the sources of inconsistencies when they occur. In our experimentation we are able to quickly uncover numerous intra- and inter-lexicon inconsistencies in all of the input SLs tested and to suggest lexicon entries for a linguist to focus on in "debugging" the lexicon.

Background and Previous Work

Sentiment resources have taken two basic approaches to polarity annotation: discrete and fractional. In the discrete approach, polarity is defined to be one of the discrete values *positive*, *negative*, or *neutral*. A word or a synset takes exactly one of the three values. QWN, AL, GI, and OF follow the *discrete polarity annotation*. In the fractional approach, polarity is defined as a 3-tuple of non-negative real numbers that sum to 1, corresponding to the positive, negative, and neutral values respectively. SWN, Micro-WNOp, and Hassan and Radev (2010) employ a *fractional polarity annotation*. For example, the single synset of the adjective `admissible` in WordNet has the sentiment tags *positive* in QWN and $\langle .25, .625, .125 \rangle$ in SWN, so here SWN gives a primarily negative polarity with some positive and less neutral polarity. We denote by PCP-D and PCP-F the polarity

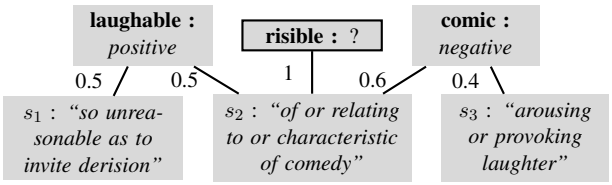

Figure 1: Discrete vs. fractional polarity consistency. Example taken from Dragut et al. (2012).

consistency problem for the discrete and fractional polarity annotations, respectively.

Dragut et al. (2012) introduces the PCP for domain independent SLs and gives a solution to a particular form of the PCP-D, but that method cannot solve PCP-F. For example, they show that the adjectives `laughable`, `comic`, and `risible` (Figure 1) constitute an inconsistency in the discrete case. AL gives positive polarity for `laughable` and OF gives negative for `comic`. If s_2 is not *positive* then `laughable` is not positive and if s_2 is not *negative* then `comic` is not negative, so there is no assignment of s_2 that satisfies the whole system. Hence there is an inconsistency. However, the following *fractional* polarity tags do satisfy the system: $s_1 : \langle 1, 0, 0 \rangle$, $s_2 : \langle .66, .34, 0 \rangle$, $s_3 : \langle 0, 1, 0 \rangle$, where the meaning of the second tag, for instance, is that s_2 is .66 positive, .34 negative, and 0 neutral. We thus see that the discrete polarity annotation is rigid and leads to more inconsistencies, whereas the fractional annotation captures more naturally the polarity spectrum of a word or synset. In this paper we give a solution to the PCP-F. The differences between our solution and that of Dragut et al. (2012) give some insight into the general differences between the fractional and discrete problems. First, the discrete case is intractable, i.e., computationally NP-complete (Dragut et al., 2012); we show in this paper (Section 3.2) that the fractional case is tractable (solvable in polynomial time). Second, the PCP-D is solved in Dragut et al. (2012) by translation to the Boolean satisfiability problem (SAT) (Schaefer, 1978); here we recast the PCP-F in terms of LP theory. Third, we show that the LP framework is a natural setting for the PCP as a whole, and that the PCP-D corresponds to the 0-1 integer LP problem (Section 3.2), a classic NP-complete problem (Karp, 2010).

Our experiments (Section 5.4) show that correcting even a small number of inconsistencies can greatly improve the accuracy of sentiment annotation tasks. We implement our algorithm as a versatile tool for debugging SLs, which helps locate the

sources of error in SLs. We apply our algorithm to both SWLs and SSLs and demonstrate the usefulness of our approach to improving SLs.

The main contributions of this paper are:

- solve the PCP-F;
- show that the PCP-F is tractable;
- show that the PCP is an instance of LP;
- develop a technique for identifying inconsistencies in SLs of various types;
- implement our algorithm as a prototype SL debugger;
- show that there is a strong correlation between polarity inconsistency in SLs and the performance of sentiment tagging tools developed on them.

2 Problem Definition

In this section we give a formal characterization of the polarity assignment of words and synsets in SLs using WordNet. We use $-$, $+$, 0 to denote negative, positive, and neutral polarities, respectively, throughout the paper.

2.1 Polarity Representation

We define the polarity of a synset or word r in WordNet to be a discrete probability distribution, called a **polarity distribution**: $P_+(r), P_-(r), P_0(r) \geq 0$ with $P_+(r) + P_-(r) + P_0(r) = 1$. $P_+(r)$, $P_-(r)$ and $P_0(r)$ represent the "likelihoods" that r is positive, negative or neutral, respectively. For instance, the WordNet synset *"worthy of reliance or trust"* of the adjective `reliable` is given the polarity distribution $P_+ = .375, P_- = .0$ and $P_0 = .625$ in SentiWordNet. We may drop r from the notation if the meaning is clear from context. The use of a polarity distribution to describe the polarity of a word or synset is shared with many previous works (Andreevskaia and Bergler, 2006; Baccianella et al., 2010; Kim and Hovy, 2006).

2.2 WordNet

A **word-synset network** $\mathcal{N}$ is a 4-tuple $(\mathcal{W}, \mathcal{S}, \mathcal{E}, f)$ where $\mathcal{W}$ is a finite set of words, $\mathcal{S}$ is a finite set of synsets, $\mathcal{E} \subseteq \mathcal{W} \times \mathcal{S}$ and f is a function assigning a positive integer to each element in $\mathcal{E}$. For any word w and synset s, s is a synset of w if $(w, s) \in \mathcal{E}$. For a pair $(w, s) \in \mathcal{E}$, $f(w, s)$ is called the **frequency of use** of w in the sense given by s. For a word w, we let $freq(w)$ denote the sum of all $f(w, s)$ such that $(w, s) \in \mathcal{E}$. We define the **relative frequency** of w with s by $rf(w, s) = \frac{f(w,s)}{freq(w)}$. If $f(w, s) = 0$, the frequency of each synset of w is increased by a small constant ϵ. We use $\epsilon = .1$ in our prototype.

2.3 Word Polarities

We contend that there exists a relation between the sentiment orientation of a word and the polarities of its related senses (synsets), and we make the assumption that this relation takes the form of a linear function. Thus, for $w \in \mathcal{W}$ and $p \in \{+, -, 0\}$, the polarity distribution of w is defined as:

$$P_p(w) = \sum_{s \in S_w} g(w, s) \cdot P_p(s), \qquad (1)$$

where $P_p(s)$ is the polarity value of synset s with polarity p and $g(w, s)$ is a rational number. For example, g can be the relative frequency of s with respect to w in WordNet: $g(w, s) = rf(w, s); \forall w \in \mathcal{W}, s \in \mathcal{S}$. Alternatively, for each word w we can draw $g(w, \cdot)$ from a Zipfian distribution, following the observation that the distribution of word senses roughly follows a Zipfian power-law (Kilgarriff, 2004; Sanderson, 1999). In this paper, we will assume $g(w, s) = rf(w, s)$.

For example, the three synsets of the adjective `reliable` with relative frequencies $\frac{9}{11}$, $\frac{1}{11}$, and $\frac{1}{11}$, respectively, are given the distributions $\langle .375, 0, .625 \rangle$, $\langle .5, 0, .5 \rangle$, and $\langle .625, 0, .375 \rangle$ in SentiWordNet. So for `reliable` we have $P_+ = \frac{9}{11}0.375 + \frac{1}{11}0.5 + \frac{1}{11}0.625 \approx 0.41$, $P_- = 0$, and $P_0 = \frac{9}{11}0.625 + \frac{1}{11}0.5 + \frac{1}{11}0.375 \approx 0.59$.

2.4 Modeling Sentiment Orientation in SLs

Words and synsets have *unique* polarities in some SLs, e.g., AL and OF. For instance, `reliable` has positive polarity in AL, GI, and OF. The question is: what does a discrete annotation of `reliable` tell us about its polarity distribution? One might take it to mean that the polarity distribution is simply $\langle 1, 0, 0 \rangle$. This contradicts the information in SWN, which gives some neutral polarity for all of the synsets of `reliable`. So a better polarity distribution would allow $P_0 > 0$. Furthermore, given that $\langle .41, 0, .59 \rangle$, $\langle .40, 0, .60 \rangle$, and $\langle .45, 0, .55 \rangle$ give virtually identical information to a sentiment analyst, it seems unreasonable to expect exactly one to be the correct polarity tag for `reliable` and the other two incorrect. Therefore, instead of claiming to pinpoint an exact polarity distribution for a word, we propose to set a boundary on its variation. This establishes a

range of values, instead of a single point, in which SLs can be said to agree.

Thus, for a word w, we can define

$$\text{polarity}(w) = \begin{cases} + & \text{if } P_+ > P_- \\ - & \text{if } P_- > P_+ \end{cases} \quad (2)$$

which we refer to as MAX_POL. This model is adopted either explicitly or implicitly by numerous works (Hassan and Radev, 2010; Kim and Hovy, 2004; Kim and Hovy, 2006; Qiu et al., 2009). Another model is the *majority sense model*, called MAJORITY, (Dragut et al., 2012), where

$$\text{polarity}(w) = \begin{cases} + & \text{if } P_+ > P_- + P_0 \\ - & \text{if } P_- > P_+ + P_0 \end{cases} \quad (3)$$

Another polarity model, MAX, is defined as

$$\text{polarity}(w) = \begin{cases} + & \text{if } P_+ > P_- \,\&\, P_+ > P_0 \\ - & \text{if } P_- > P_+ \,\&\, P_- > P_0 \end{cases} \quad (4)$$

For instance, `reliable` conveys positive polarity according to MAX_POL, since $P_+ > P_-$, but neutral according to MAJORITY. When the condition of being neither positive nor negative can be phrased as a conjunction of linear inequalities, as is the case with MAJORITY and MAX_POL, then we define neutral as not positive and not negative. These model definitions can be applied to synsets as well.

2.5 Polarity Consistency Definition

Instead of defining consistency for SLs dependent on a choice of model, we develop a generic definition applicable to a wide variety of models, including all of those discussed above. We require that the polarity of a word or synset in the network $\mathcal{N}$ be characterized by a *set of linear inequalities (constraints)* with rational coefficients. Formally, for each word $w \in \mathcal{W}$, the knowledge that w has a discrete polarity $p \in \{+, -, 0\}$ is characterized by a set of linear inequalities:

$$\psi(w, p) = \{a_{i,0}P_+ + a_{i,1}P_- + a_{i,2}P_0 \preceq b_i\}, \quad (5)$$

where $\preceq \in \{\leq, <\}$ and $a_{i,0}, a_{i,1}, a_{i,2}, b_i \in \mathbb{Q}$, $i = 0, 1, \ldots, m$. For instance, if the MAX model is used, for $w =$ `worship` whose polarity is positive in OF, we get the following set of inequalities: $\psi(w, +) = \{P_+ - P_- > 0, P_+ - P_0 > 0\} = \{(-1)P_+ + 1P_- + 0P_0 < 0, (-1)P_+ + 0P_- + 1P_0 < 0\}$.

Let $\mathcal{L}$ be an SL. We denote the system of inequalities introduced by all words and synsets in $\mathcal{L}$ with known polarities in the network $\mathcal{N}$ by $\Psi'(\mathcal{N}, \mathcal{L})$. The variables in $\Psi'(\mathcal{N}, \mathcal{L})$ are

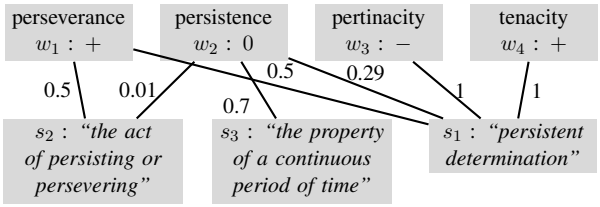

Figure 2: A network of 4 words and 3 synsets

$P_+(r), P_-(r)$ and $P_0(r)$, $r \in \mathcal{W} \cup \mathcal{S}$. Denote by $\Upsilon'(\mathcal{N}, \mathcal{L})$ the set of constraints implied by the polarity distributions for all $r \in \mathcal{L}$: $P_+(r) + P_-(r) + P_0(r) = 1$ and $P_{p \in \{+,-,0\}}(r) \geq 0, \forall r \in \mathcal{W} \cup \mathcal{S}$. Let $\Phi'(\mathcal{N}, \mathcal{L}) = \Psi'(\mathcal{N}, \mathcal{L}) \cup \Upsilon'(\mathcal{N}, \mathcal{L})$.

Example 1. *Let w_1, w_2, w_3, and w_4 be the nouns* `perseverance`, `persistence`, `pertinacity`, *and* `tenacity`, *respectively, which are in OF with polarities +, 0, −, and +, respectively (Figure 2). Assuming the MAJORITY model, $\psi(w_1, +) = \{P_+(w_1) > P_-(w_1) + P_0(w_1)\} = \{P_+(w_1) > 1 - P_+(w_1)\} = \{-P_+(w_1) < -\frac{1}{2}\}$, and $\psi(w_2, 0) = \{P_+(w_2) \leq P_-(w_2) + P_0(w_2), P_-(w_2) \leq P_+(w_2) + P_0(w_2)\} = \{P_+(w_2) \leq \frac{1}{2}, P_-(w_2) \leq \frac{1}{2}\}$. Similarly, $\psi(w_3, -) = \{-P_-(w_3) < -\frac{1}{2}\}$ and $\psi(w_4, -) = \{-P_+(w_4) < -\frac{1}{2}\}$.*

Definition 1. *A sentiment lexicon $\mathcal{L}$ is **consistent** if the system $\Phi'(\mathcal{N}, \mathcal{L})$ is feasible, i.e., has a solution.*

The PCP is then the problem of deciding if a given SL $\mathcal{L}$ is consistent. In general, PCP can be stated as follows: *Given an assignment of polarities to the words, does there exist an assignment of polarities to the synsets that agrees with that of the words*? If the polarity annotation is discrete, we have the PCP-D; if the polarity is fractional, we have PCP-F. Our focus is PCP-F in this paper.

The benefits of a generic problem model are at least two-fold. First, different linguists may have different views about the kinds of inequalities one should use to express the probability distribution of a word with a unique polarity in some SL. The new model can accommodate divergent views as long as they are expressed as linear constraints. Second, the results proven for the generic model will hold for any particular instance of the model.

3 Polarity Consistency: an LP Approach

A careful analysis of the proposed formulation of the problem of SL consistency checking reveals that this can be naturally translated into an LP problem. The goal of LP is the optimization of a linear objective function, subject to lin-

ear (in)equality constraints. LP problems are expressed in *standard* form as follows:

$$\begin{aligned} &\text{minimize } \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\text{subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ &\text{and } \mathbf{x} \geq \mathbf{0} \end{aligned} \quad (6)$$

$\mathbf{x}$ represents the vector of variables (to be determined), $\mathbf{c}$ and $\mathbf{b}$ are vectors of (known) coefficients, $\mathbf{A}$ is a (known) matrix of coefficients, and $(\cdot)^{\mathrm{T}}$ is the matrix transpose. An LP algorithm finds a point in the *feasible region* where $\mathbf{c}^{\mathrm{T}}\mathbf{x}$ has the smallest value, if such a point exists. The feasible region is the set of $\mathbf{x}$ that satisfy the constraints $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.

There are several non-trivial challenges that need to be addressed in transforming our problem (i.e., the system $\Phi'(\mathcal{N}, \mathcal{L})$) into an LP problem. For instance, we have both strict and weak inequalities in our model, whereas standard LP does not include strict inequalities. We describe the steps of this transformation next.

3.1 Translation to LP

In our problem, $\mathbf{x}$ is the concatenation of all the triplets $\langle P_+(r), P_-(r), P_0(r)\rangle$ for all $r \in \mathcal{W} \cup \mathcal{S}$.

Eliminate Word Related Variables. For each word $w \in \mathcal{L}$ we replace $P_+(w), P_-(w)$ and $P_0(w)$ with their corresponding expressions according to Equation 1; then the linear system $\Phi'(\mathcal{N}, \mathcal{L})$ has only the synset variables $P_+(s), P_-(s)$ and $P_0(s)$ for $s \in \mathcal{S}$.

Example (continued). *Using the relative frequencies of Figure 2 in Equation 1 we get:*
$\psi(w_1,+) = \{-.5P_+(s_1) - .5P_+(s_2) < -\frac{1}{2}\}$,
$\psi(w_2,0) = \{.29P_+(s_1) + .01P_+(s_2) + .7P_+(s_3) \leq \frac{1}{2},$
$.29P_-(s_1) + .01P_-(s_2) + .7P_-(s_3) \leq \frac{1}{2}\}$,
$\psi(w_3,-) = \{-P_-(s_1) < -\frac{1}{2}\}$, *and*
$\psi(w_4,+) = \{-P_+(s_1) < -\frac{1}{2}\}$.

Equality. The system $\Phi'(\mathcal{N}, \mathcal{L})$ contains constraints of the form $P_+(s) + P_-(s) + P_0(s) = 1$ for each $s \in \mathcal{S}$, but observe that there are no equality constraints in the standard LP form (Equation 6). The usual conversion procedure is to replace a given equality constraint: $\mathbf{a}^{\mathrm{T}}\mathbf{x} = b$, with: $\mathbf{a}^{\mathrm{T}}\mathbf{x} \leq b$ and $-\mathbf{a}^{\mathrm{T}}\mathbf{x} \leq -b$. However, this procedure increases the number of constraints in $\Phi'(\mathcal{N}, \mathcal{L})$ linearly. This can have a significant computation impact since $\Phi'(\mathcal{N}, \mathcal{L})$ may have thousands of constraints (see discussion in Section 5.3). Instead, we can show that the system F obtained by performing the following two-step transformation is equivalent to $\Phi'(\mathcal{N}, \mathcal{L})$, in the sense that F is feasible iff $\Phi'(\mathcal{N}, \mathcal{L})$ is feasible. For every $s \in \mathcal{S}$, (Step 1) we convert each $P_+(s) + P_-(s) + P_0(s) = 1$ to $P_+(s) + P_-(s) \leq 1$, and (Step 2) we replace every $P_0(s)$ in $\Phi'(\mathcal{N}, \mathcal{L})$ with $1 - P_+(s) - P_-(s)$.

Strict Inequalities. Strict inequalities are not allowed in LP and their presence in inequality systems in general poses difficulties to inequality system solvers (Goberna et al., 2003; Goberna and Rodriguez, 2006; Ghaoui et al., 1994). Fortunately results developed by the LP community allow us to overcome this obstacle and maintain the flexibility of our proposed model. We introduce a new variable $y \geq 0$, and for every strict constraint of the form $\mathbf{a}^{\mathrm{T}}\mathbf{x} < b$, we rewrite the inequality as $\mathbf{a}^{\mathrm{T}}\mathbf{x} + y \leq b$. Let $\Phi''(\mathcal{N}, \mathcal{L})$ be this new system of constraints. We modify the objective function (previously null) to maximize y (i.e., minimize $-y$). Denote by F' the LP that maximizes y subject to $\Phi''(\mathcal{N}, \mathcal{L})$. We can show that $\Phi'(\mathcal{N}, \mathcal{L})$ is feasible iff F' is feasible and $y \neq 0$. A sketch of the proof is as follows: if $y > 0$ then $\mathbf{a}^{\mathrm{T}}\mathbf{x} + y \leq b$ implies $\mathbf{a}^{\mathrm{T}}\mathbf{x} < b$. Conversely, if $\mathbf{a}^{\mathrm{T}}\mathbf{x} < b$ then $\exists y > 0$ such that $\mathbf{a}^{\mathrm{T}}\mathbf{x} + y \leq b$, and maximizing for y will yield a $y > 0$ iff one is feasible. This step is omitted if we have no strict constraints in $\Phi'(\mathcal{N}, \mathcal{L})$.

Example (continued). *The formulations of $\psi(w_1, +)$, $\psi(w_3, -)$, and $\psi(w_4, +)$ involve strict inequalities, so they are rewritten in $\Phi''(\mathcal{N}, \mathcal{L})$, e.g., $\psi''(w_4, +) = \{-P_+(s_1) + y \leq -\frac{1}{2}\}$.*

We denote by $\Phi(\mathcal{N}, \mathcal{L})$ the standard form of $\Phi'(\mathcal{N}, \mathcal{L})$ obtained by applying the above steps. This is the input to an LP solver.

Theorem 1. *Sentiment lexicon $\mathcal{L}$ is polarity consistent iff $\Phi(\mathcal{N}, \mathcal{L})$ is feasible.*

3.2 Time Complexity

For the network $\mathcal{N}$ and an SL $\mathcal{L}$, the above translation algorithm converts the PCP into an LP problem on the order of $O(|\mathcal{E}|)$, a polynomial time conversion. The general class of linear programming problems includes subclasses that are NP-hard, such as the *integer linear programming* (ILP) problems, as well as polynomial solvable subclasses. We observe that our problem is represented by a system of *rational* linear inequalities. This class of LP problems is solvable in polynomial time (Khachiyan, 1980; Gács and Lovász, 1981). This (informally) proves that the PCP-F is solvable in polynomial time. PCP is NP-complete in the discrete case (Dragut et al., 2012). This is not surprising since in our LP formulation of the

PCP, the discrete case corresponds to the 0-1 integer programming (BIP) subclass. (Recall that in the discrete case each synset has a unique polarity.) BIP is the special case of integer programming where variables are required to be 0 or 1. BIP is a classic NP-hard problem (Garey and Johnson, 1990). We summarize these statements in the following theorem.

Theorem 2. *The PCP-F problem is P and the PCP-D is NP-complete.*

We proved a more general and more comprehensive result than Dragut et al. (2012). The PCP solved by Dragut et al. (2012) is a particular case of PCP-D: it can be obtained by instantiating our framework with the MAJORITY model (Equation 3) and requiring each synset to take a unique polarity. We believe that the ability to encompass both fractional and discrete cases within one framework, that of LP, is an important contribution, because it helps to give structure to the general problem of polarity consistency and to contextualize the difference between the approaches.

4 Towards Debugging SLs

Simply stating that an SL is *inconsistent* is of little practical use unless accompanying assistance in diagnosing and repairing inconsistencies is provided. Automated assistance is necessary in the face of the scale and complexity of modern SLs: e.g., AL has close to 7,000 entries, SWN annotates the entirety of WordNet, over 206,000 word-sense pairs. There are unique and interesting problems associated with inconsistent SLs, among them: (1) isolate a (small) subset of words/synsets that is polarity inconsistent, but becomes consistent if one of them is removed; we call this an Irreducible Polarity Inconsistent Subset (IPIS); (2) return an IPIS with smallest cardinality (intuitively, such a set is easiest to repair); (3) find all IPISs, and (4) find the largest polarity consistent subset of an inconsistent SL. In the framework of linear systems of constraints, the problems (1) - (4) correspond to (i) the identification of an Irreducible Infeasible Subset (IIS) of constraints within $\Phi(\mathcal{N}, \mathcal{L})$, (ii) finding IIS of minimum cardinality, (iii) finding all IISs and (iv) finding the largest set of constraints in $\Phi(\mathcal{N}, \mathcal{L})$ that is feasible, respectively. An IIS is an infeasible subset of constraints that becomes feasible if any single constraint is removed. Problems (ii) - (iv) are NP-hard and some may even be difficult to approximate (Amaldi and Kann, 1998; Chinneck, 2008; Chakravarti, 1994; Tamiz et al., 1996). We focus on problem (1) in this paper, which we solve via IIS discovery. We keep a bijective mapping from words and synsets to constraints such that for any given constraint, we can uniquely identify the word or synset in $\Phi(\mathcal{N}, \mathcal{L})$ from which it was introduced. Hence, once an IIS is isolated, we know the corresponding words or synsets. Modern LP solvers typically can give an IIS when a system is found to be infeasible, but none give all IISs or the IIS of minimum size.

Example (continued). *The polarity assignments of w_1, w_2, w_3, and w_4, are consistent iff there exist polarity distributions $\langle P_+(s_i), P_-(s_i), P_0(s_i)\rangle$ for $i = 1, 2, 3$, such that:* $y > 0$
$\psi(w_1, +)$: $-.5P_+(s_1) + .5P_+(s_2) + y \leq -\frac{1}{2}$,
$\psi(w_2,0)$:$.29P_+(s_1) + .01P_+(s_2) + .7P_+(s_3) \leq \frac{1}{2}$
AND $.29P_-(s_1) + .01P_-(s_2) + .7P_-(s_3) \leq \frac{1}{2}$,
$\psi(w_3, -) : -P_-(s_1) + y \leq -\frac{1}{2}$,
$\psi(w_4, +) : -P_+(s_1) + y \leq -\frac{1}{2}$,
$\upsilon(s_1)$: $P_+(s_1) + P_-(s_1) \leq 1$ *AND* $P_+(s_1), P_-(s_1) \geq 0$,
$\upsilon(s_2)$: $P_+(s_2) + P_-(s_2) \leq 1$ *AND* $P_+(s_2), P_-(s_2) \geq 0$,
$\upsilon(s_3)$: $P_+(s_3) + P_-(s_3) \leq 1$ *AND* $P_+(s_3), P_-(s_3) \geq 0$.

Upon examination, if $y > 0$, then $\psi(w_3, -)$ implies $P_-(s_1) > \frac{1}{2}$ and $\psi(w_4, +)$ implies $P_+(s_1) > \frac{1}{2}$. Then $P_+(s_1) + P_-(s_1) > 1$, contradicting $\upsilon(s_1)$. Hence, this LP system is infeasible. Moreover $\{\psi(w_3, -), \psi(w_4, +), \upsilon(s_1)\}$ is an IIS. Tracing back we get that the set of words $\{w_3, w_4\}$ is inconsistent. Therefore it is an IPIS.

Isolating IPISs helps focus SL diagnosis and repair efforts. Fixing SLs via IIS isolation proceeds iteratively: (1) isolate an IIS, (2) determine a repair for this IIS, (3) if the model is still infeasible, go to step (1). This approach is well summarized by Greenberg's aphorism: "diagnosis = isolation + explanation" (Greenberg, 1993). The proposed use requires human interaction to effect the changes to the lexicon. One might ask if this involvement is strictly necessary; in response we draw a parallel between our SL debugger and a software debugger. A software debugger can identify a known programming error, say the use of an undefined variable. It informs the programmer, but it does not assign a value to the variable itself. It requires the user to make the desired assignment. Similarly, our debugger can deterministically identify an inconsistent component, but it cannot deterministically decide which elements to adjust. In most cases, this is simply not an objective decision. To illustrate this point, from our example, we know that minimally one

	SL	adj.	adv.	noun	verb	total
SWLs	UN	3,084	940	2,340	1,812	8,176
	AL	1,486	377	2	0	1,865
	GI	1,337	121	1,474	1,050	3,982
	OF	2,608	775	1,907	1,501	6,791
SSLs	SWN	18,156	3,621	82,115	13,767	117,659
	QWN	4,060	40	7,404	4,006	15,510
	MWN	255	30	487	283	1,055

Table 1: Counts of words/synsets in each SL

of $\texttt{pertinacity}(-)$ and $\texttt{tenacity}(+)$ must be adjusted, but the determination as to which requires the subjective analysis of a domain expert.

In this paper, we do not repair any of the discovered inconsistencies. We focus on isolating as many IPISs as possible.

5 Experiments

The purpose of our experimental work is manifold, we show that: (1) inconsistencies exist in and between SLs, (2) our algorithm is effective at uncovering them in the various types of SLs proposed in the literature, (3) fractional polarity representation is more flexible than discrete, giving orders of magnitude fewer inconsistencies, and (4) sentiment analysis is significantly improved when the inconsistencies of a basis SL are corrected.

Experiment Setup: We use four SWLs: GI, AL, OF and their union, denoted UN, and three SSLs: QWN, SWN and MicroWN-Op. The distribution of their entries is given in Table 1. The MAJORITY model (Equation 3) is used in all trials. This allows for direct comparison with Dragut et al. (2012). We implemented our algorithm in Java interfacing with the GUROBI LP solver[2], and ran the tests on a 4×1.70GHz core computer with 6GB of main memory.

5.1 Inconsistencies in SWLs

In this set of experiments, we apply our algorithm to GI, AL, OF and UN. We find *no* inconsistencies in AL, only 2 in GI, and 35 in both UN and OF (Table 2). (Recall that an inconsistency is a *set* of words whose polarities cannot be concomitantly satisfied.) These numbers *do not* represent all possible inconsistencies (See discussion in Section 4). In general, the number of IISs for an infeasible system can be exponential in the size of the system $\Phi(\mathcal{N}, \mathcal{L})$ (Chakravarti, 1994), however our results suggest that in practice this does not occur.

Compared with Dragut et al. (2012), we see a marked decrease in the number of inconsistencies.

[2]www.gurobi.com

	adj.	adv.	noun	verb	total
UN	8	14	5	8	35
AL	0	0	0	-	0
GI	2	0	0	0	2
OF	7	15	4	9	35

Table 2: SWL-Internal Inconsistencies

Inconsistency Ratios

SWL	adj.	adv.	noun	verb	total
UN	0.67	0.89	0.85	0.81	0.78
AL	0.63	0.8	1	-	0.66
GI	0.6	0.41	0.87	0.91	0.78
OF	0.66	0.87	0.82	0.77	0.76

Table 3: SentiWordNet paired with SWLs

They found 249, 2, 14, and 240 inconsistencies in UN, AL, GI, and OF, respectively. These inconsistencies are obtained in the first iteration of their SAT-Solver. *This shows that about 86% of inconsistent words in a discrete framework can be made consistent in a fractional system.*

5.2 Inconsistencies in SSLs

In this set of experiments we check the polarity inconsistencies between SWLs and SSLs. We pair each SSL with each of the SWLs.

SentiWordNet. SWN is an automatically generated SL with a fractional polarity annotation of every synset in WordNet. Since SWN annotates *every* synset in WordNet, there are no free variables in this trial. Each variable $P_{p\in\{+,-,0\}}(s)$ for $s \in \mathcal{S}$ is fully determined by SWN, so this amounts to a constant on the left hand side of each inequality. Our task is to simply check whether the inequality holds between the constant on the left and that on the right. Table 3 gives the proportion of words from each SWL that is inconsistent with SWN. We see there is substantial disagreement between SWN and all of the SWLs, in most cases more than 70% disagreement. For example, 5,260 of the 6,921 words in OF do not agree with the polarities assigned to their senses in SWN. This outcome is deeply surprising given that all these SLs are *domain independent* – no step in their construction processes hints to a specific domain knowledge. This opens up the door to future analysis of SL acquisition. For instance, examining the impact that model choice (e.g., MAJORITY vs. MAX) has on inter-lexicon agreement.

Q-WordNet. QWN gives a discrete polarity for 15,510 WordNet synsets. When a synset is annotated in QWN, its variables, $P_{p\in\{+,-,0\}}(s)$, are assigned the QWN values in Φ; a feasible assignment is sought for the remaining free variables. An inconsistency may occur among a set of words, or

	UN	AL	GI	OF
total	345	34	139	325

Table 4: Q-WordNet paired with SWLs.

a set of words and synsets. Table 4 depicts the outcome of this study. We obtain 345 inconsistencies between QWN and UN. The reduced number of inconsistencies with AL (34) is explained by their limited "overlay" (QWN has only 40 adverb synsets). Dragut et al. (2012) reports 455 inconsistencies between QWN and UN, 110 more than we found here. Again, this difference is due to the rigidity of the discrete case, which leads to more inconsistencies in general.

Micro-WNOp. This is a fractional SSL of 1,105 synsets from WordNet manually annotated by five annotators. The synsets are divided into three groups: 110 annotated by the consensus of the annotators, 496 annotated individually by three annotators, and 499 annotated individually by two annotators. We take the average polarities of groups 2 and 3 and include this data as two additional sets of values. Table 5 gives the inconsistencies per user in each group. For Groups 2 and 3, we give the average number of inconsistencies among the users (Avg. Incons. in Table 5) as well as the inconsistencies of the averaged annotations (Avg. User in Table 5).

Micro-WNOp gives us an opportunity to analyze the robustness of our method by comparing the number of inconsistencies of the individual users to that of the averaged annotation. Intuitively, we expect that the average number of inconsistencies in a group of users to be close to the number of inconsistencies for the user averaged annotations. This is clearly apparent from Table 5, when comparing Lines 4 and 5 in Group 2 and Lines 3 and 4 in Group 3. For example, Group 2 has an average of 68 inconsistencies for OF, which is very close to the number of inconsistencies, 63, obtained for the group averaged annotations. This study suggests a potential application of our algorithm: to estimate the confidence weight (trust) of a user's polarity annotation. A user with good polarity consistency receives a higher weight than one with poor polarity consistency. This can be applied in a multi-annotator SL scenario.

5.3 Computation

We provide information about the runtime execution of our method in this section. Over all of our experiments, the resulting systems of constraints can be as small as 2 constraints with 2 variables and as large as 3,330 constraints with 4,946 variables. We achieve very good overall execution times, 68 sec. on average. At its peak, our algorithm requires 770MB of memory. Compared to the SAT approach by Dragut et al. (2012), which takes about 10 min. and requires about 10GB of memory, our method is several orders of magnitude more efficient and more practical, paving the way to building practical SL debugging tools.

		UN	AL	GI	OF
	Common	45	3	13	43
Group 2	**User 1**	88	10	59	75
	User 2	50	8	24	48
	User 3	97	12	64	82
	Avg. Incons.	78	10	49	68
	Avg. User 1,2,3	69	8	40	63
Group 3	**User 4**	72	9	46	60
	User 5	70	8	46	59
	Avg. Incons.	71	9	46	60
	Avg. User 4,5	68	8	42	57

Table 5: Micro-WNOp – SWD Inconsistencies

5.4 Inconsistency & Sentiment Annotation

This experiment has two objectives: (1) show that two inconsistent SLs give very different results when applied to sentiment analysis tasks and (2) given an inconsistent SL D, and D' an improved version of D with fewer inconsistencies, show that D' gives better results than D in sentiment analysis tasks. We use a third-party sentiment annotation tool that utilizes SLs, Opinion Parser (Liu, 2012). We give the instantiations of D below.

In (1), we use the dataset **aclImdb** (Maas et al., 2011), which consists of 50,000 reviews, and the SLs UN and SWN. Let UN$'$ and SWN$'$ be the subsets of UN and SWN, respectively, with the property that they have the same set of $(word, pos)$ pair entries and $word$ appears in **aclImdb**. UN$'$ and SWN$'$ have 6,003 entries. We select from **aclImdb** the reviews with the property that they contain at least 50 words in SWN$'$ and UN$'$. This gives 516 negative and 567 positive reviews, a total of 1,083 reviews containing a total of 31,701 sentences. Opinion Parser is run on these sentences using SWN$'$ and UN$'$. We obtain that 16,741 (52.8%) sentences acquire different polarities between the two SLs.

In (2), we use 110 randomly selected sentences from **aclImdb**, which we manually tagged with their overall polarities. We use OF and OF$'$, where OF$'$ is the version of OF after just six inconsistencies are manually fixed. We run Opinion Parser on these sentences using OF and OF$'$. We obtain an accuracy of 42% with OF and 47% with OF$'$, an

improvement of 8.5% for just a small fraction of corrected inconsistencies.

These two experiments show a strong correlation between polarity inconsistency in SLs and its effect on sentiment tagging in practice.

6 Conclusion

Resolving polarity inconsistencies helps to improve the accuracy of sentiment analysis tasks. We show that LP theory provides a natural framework for the polarity consistency problem. We give a polynomial time algorithm for deciding whether an SL is polarity consistent. If an SL is found to be inconsistent, we provide an efficient method to uncover sets of words or word senses that are inconsistent and require linguists' attention. Effective SL debugging tools such as this will help in the development of improved SLs for use in sentiment analysis tasks.

7 Acknowledgments

We would like to thank Bing Liu for running the experiments of Section 5.4 on his commercial tool Opinion Parser, Christiane Fellbaum for the discussions on polarity inconsistency, and Prasad Sistla for the discussions on linear programming. We would also like to thank the reviewers for their time, effort, and insightful feedback.

References

Rodrigo Agerri and Ana García-Serrano. 2010. Q-wordnet: Extracting polarity from wordnet senses. In *LREC*.

Edoardo Amaldi and Viggo Kann. 1998. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209.

A. Andreevskaia and S. Bergler. 2006. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *EACL*.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *LREC*.

Avrim Blum, John Lafferty, Mugizi Robert Rwebangira, and Rajashekar Reddy. 2004. Semi-supervised learning using randomized mincuts. In *ICML*.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *IJCAI*.

Juergen Bross and Heiko Ehrig. 2013. Automatic construction of domain and aspect specific sentiment lexicons for customer review mining. In *CIKM*.

S. Cerini, V. Compagnoni, A. Demontis, M. Formentelli, and G. Gandini, 2007. *Language resources and linguistic theory: Typology, second language acquisition, English linguistics.*, chapter Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. Franco Angeli Editore, Milano, IT.

Nilotpal Chakravarti. 1994. Some results concerning post-infeasibility analysis. *European Journal of Operational Research*, 73(1).

Yanqing Chen and Steven Skiena. 2014. Building sentiment lexicons for all major languages. In *ACL*.

John W Chinneck. 2008. *Feasibility and infeasibility in optimization: algorithms and computational methods*. International Series in Operations Research and Management Science. Springer, Dordrecht.

Yejin Choi and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *EMNLP*.

Yoonjung Choi and Janyce Wiebe. 2014. +/-effectwordnet: Sense-level lexicon acquisition for opinion inference. In *EMNLP*.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *WSDM*.

Eduard C. Dragut, Hong Wang, Clement Yu, Prasad Sistla, and Weiyi Meng. 2012. Polarity consistency checking for sentiment dictionaries. In *ACL*.

Weifu Du, Songbo Tan, Xueqi Cheng, and Xiaochun Yun. 2010. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *WSDM*.

A. Esuli and F. Sebastiani. 2006. Determining term subjectivity and term orientation for opinion mining. In *EACL*.

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Commun. ACM*, 56(4).

C. Fellbaum. 1998. *WordNet: An On-Line Lexical Database and Some of its Applications.* MIT Press, Cambridge, MA.

Song Feng, Jun Sak Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *ACL*.

Peter Gács and Laszlo Lovász. 1981. Khachiyans algorithm for linear programming. In *Mathematical Programming at Oberwolfach*, volume 14 of *Mathematical Programming Studies*. Springer Berlin Heidelberg.

Michael R. Garey and David S. Johnson. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.

Laurent E. Ghaoui, Eric Feron, and Vendataramanan Balakrishnan. 1994. *Linear Matrix Inequalities in System & Control Theory (Studies in Applied Mathematics)*, volume 15. SIAM.

Miguel A. Goberna and Margarita M. L. Rodriguez. 2006. Analyzing linear systems containing strict inequalities via evenly convex hulls. *European Journal of Operational Research*, 169(3).

Miguel A. Goberna, Valentin Jornet, and Margarita M.L. Rodriguez. 2003. On linear systems containing strict inequalities. *Linear Algebra and its Applications*, 360(0).

Harvey J. Greenberg. 1993. How to analyze the results of linear programspart 3: Infeasibility diagnosis. *Interfaces*, 23(6).

Ahmed Hassan and Dragomir Radev. 2010. Identifying text polarity using random walks. In *ACL*.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *ACL*.

Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. 2010. Generating focused topic-specific sentiment lexicons. In *ACL*.

Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In *EMNLP-CoNLL*.

J. Kamps, M. Marx, R. Mokken, and M. de Rijke. 2004. Using wordnet to measure semantic orientation of adjectives. In *LREC*.

Richard M. Karp. 2010. Reducibility among combinatorial problems. In *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*. Springer Berlin Heidelberg.

L. G. Khachiyan. 1980. Polynomial algorithms in linear programming. *Zh. Vychisl. Mat. Mat. Fiz.*, 20(1).

Adam Kilgarriff. 2004. How dominant is the commonest sense of a word? In *Text, Speech, and Dialogue*, volume 3206 of *Lecture Notes in Artificial Intelligence*.

M. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *COLING*.

Soo-Min Kim and Eduard Hovy. 2006. Identifying and analyzing judgment opinions. In *HLT-NAACL*.

Beata Beigman Klebanov, Nitin Madnani, and Jill Burstein. 2013. Using pivot-based paraphrasing and sentiment profiles to improve a subjectivity lexicon for essay data. In *ACL*.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *WWW*.

Andrew L. Maas, Raymond E. Daly, Peter Pham, Dan Huang, Andrew Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*.

Saif Mohammad, Cody Dunne, and Bonnie Dorr. 2009. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *EMNLP*.

Wei Peng and Dae Hoon Park. 2011. Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization. In *ICWSM*.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *IJCAI*.

Mark Sanderson. 1999. The impact on retrieval effectiveness of skewed frequency distributions. *ACM Transactions on Information Systems*, 17(4).

Thomas J. Schaefer. 1978. The complexity of satisfiability problems. In *STOC*.

Alexander Schrijver. 1986. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA.

P. Stone, D. Dunphy, M. Smith, and J. Ogilvie. 1966. *The General Inquirer: A computer approach to content analysis*. MIT Press.

M. Taboada and J. Grieve. 2004. Analyzing appraisal automatically. In *AAAI Spring Symposium*.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *ACL*.

M. Tamiz, S. J. Mardle, and D. F. Jones. 1996. Detecting IIS in infeasible linear programmes using techniques from goal programming. *Comput. Oper. Res.*, 23(2).

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *COLING*.

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

Gbolahan K. Williams and Sarabjot Singh Anand. 2009. Predicting the polarity strength of adjectives using wordnet. In *ICWSM*.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*.

Yunfang Wu and Miaomiao Wen. 2010. Disambiguating dynamic sentiment ambiguous adjectives. In *COLING*.