Connotation Lexicon: A Dash of Sentiment Beneath the Surface Meaning

Song Feng **Jun Seok Kang** **Polina Kuznetsova** **Yejin Choi**
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794-4400
songfeng, junkang, pkuznetsova, ychoi@cs.stonybrook.edu

Abstract

Understanding the connotation of words plays an important role in interpreting subtle shades of sentiment beyond denotative or surface meaning of text, as seemingly objective statements often allude nuanced sentiment of the writer, and even purposefully conjure emotion from the readers' minds. The focus of this paper is drawing nuanced, connotative sentiments from even those words that are objective on the surface, such as *"intelligence"*, *"human"*, and *"cheesecake"*. We propose induction algorithms encoding a diverse set of linguistic insights (semantic prosody, distributional similarity, semantic parallelism of coordination) and prior knowledge drawn from lexical resources, resulting in the first *broad-coverage* connotation lexicon.

1 Introduction

There has been a substantial body of research in sentiment analysis over the last decade (Pang and Lee, 2008), where a considerable amount of work has focused on recognizing sentiment that is generally explicit and pronounced rather than implied and subdued. However in many real-world texts, even seemingly objective statements can be opinion-laden in that they often allude nuanced sentiment of the writer (Greene and Resnik, 2009), or purposefully conjure emotion from the readers' minds (Mohammad and Turney, 2010). Although some researchers have explored formal and statistical treatments of those implicit and implied sentiments (e.g. Wiebe et al. (2005), Esuli and Sebastiani (2006), Greene and Resnik (2009), Davidov et al. (2010)), automatic analysis of them largely remains as a big challenge.

In this paper, we concentrate on understanding the connotative sentiments of words, as they play an important role in interpreting subtle shades of sentiment beyond denotative or surface meaning of text. For instance, consider the following:

> Geothermal replaces oil-heating; it helps reducing greenhouse *emissions*.[1]

Although this sentence could be considered as a factual statement from the general standpoint, the subtle effect of this sentence may not be entirely objective: this sentence is likely to have an influence on readers' minds in regard to their opinion toward *"geothermal"*. In order to sense the subtle overtone of sentiments, one needs to know that the word *"emissions"* has generally negative connotation, which geothermal *reduces*. In fact, depending on the pragmatic contexts, it could be precisely the intention of the author to transfer his opinion into the readers' minds.

The main contribution of this paper is a *broad-coverage* connotation lexicon that determines the connotative polarity of even those words with ever so subtle connotation beneath their surface meaning, such as *"Literature"*, *"Mediterranean"*, and *"wine"*. Although there has been a number of previous work that constructed sentiment lexicons (e.g., Esuli and Sebastiani (2006), Wilson et al. (2005a), Kaji and Kitsuregawa (2007), Qiu et al. (2009)), which seem to be increasingly and inevitably expanding over words with (strongly) connotative sentiments rather than explicit sentiments alone (e.g., "gun"), little prior work has directly tackled this problem of learning connotation,[2] and much of the subtle connotation of many seemingly objective words is yet to be determined.

[1] Our learned lexicon correctly assigns negative polarity to *emission*.

[2] A notable exception would be the work of Feng et al.

POSITIVE	NEGATIVE
FEMA, Mandela, Intel, Google, Python, Sony, Pulitzer, Harvard, Duke, Einstein, Shakespeare, Elizabeth, Clooney, Hoover, Goldman, Swarovski, Hawaii, Yellowstone	Katrina, Monsanto, Halliburton, Enron, Teflon, Hiroshima, Holocaust, Afghanistan, Mugabe, Hutu, Saddam, Osama, Qaeda, Kosovo, Helicobacter, HIV

Table 1: Example Named Entities (Proper Nouns) with Polar Connotation.

A central premise to our approach is that it is collocational statistics of words that affect and shape the polarity of connotation. Indeed, the etymology of *"connotation"* is from the Latin *"com-"* ("together or with") and *"notare"* ("to mark"). It is important to clarify, however, that we do not simply assume that words that collocate share the same polarity of connotation. Although such an assumption played a key role in previous work for the analogous task of learning sentiment lexicon (Velikovich et al., 2010), we expect that the same assumption would be less reliable in drawing subtle connotative sentiments of words. As one example, the predicate "cure", which has a positive connotation typically takes arguments with negative connotation, e.g., "disease", when used as the "relieve" sense.[3]

Therefore, in order to attain a broad coverage lexicon while maintaining good precision, we guide the induction algorithm with multiple, carefully selected linguistic insights: [1] distributional similarity, [2] semantic parallelism of coordination, [3] selectional preference, and [4] semantic prosody (e.g., Sinclair (1991), Louw (1993), Stubbs (1995), Stefanowitsch and Gries (2003))), and also exploit existing lexical resources as an additional inductive bias.

We cast the connotation lexicon induction task as a collective inference problem, and consider approaches based on three distinct types of algorithmic framework that have been shown successful for conventional sentiment lexicon induction:

Random walk based on HITS/PageRank (e.g., Kleinberg (1999), Page et al. (1999), Feng et al. (2011) Heerschop et al. (2011), Montejo-Ráez et al. (2012))

Label/Graph propagation (e.g., Zhu and Ghahramani (2002), Velikovich et al. (2010))

Constraint optimization (e.g., Roth and Yih (2004), Choi and Cardie (2009), Lu et al. (2011)).

We provide comparative empirical results over several variants of these approaches with comprehensive evaluations including lexicon-based, human judgments, and extrinsic evaluations.

It is worthwhile to note that not all words have connotative meanings that are distinct from denotational meanings, and in some cases, it can be difficult to determine whether the overall sentiment is drawn from denotational or connotative meanings exclusively, or both. Therefore, we encompass any sentiment from either type of meanings into the lexicon, where non-neutral polarity prevails over neutral one if some meanings lead to neutral while others to non-neutral.[4]

Our work results in the first broad-coverage connotation lexicon,[5] significantly improving both the coverage and the precision of Feng et al. (2011). As an interesting by-product, our algorithm can be also used as a proxy to measure the general connotation of real-world named entities based on their collocational statistics. Table 1 highlights some example common nouns included in the final lexicon.

The rest of the paper is structured as follows. In §2 we describe three types of induction algorithms followed by evaluation in §3. Then we revisit the induction algorithms based on constraint optimization in §4 to enhance quality and scalability. §5 presents comprehensive evaluation with human judges and extrinsic evaluations. Related work and conclusion are in §6 and §7.

(2011) but with practical limitations. See §3 for detailed discussion.

[3]Note that when "cure" is used as the "preserve" sense, it expects objects with non-negative connotation. Hence word-sense-disambiguation (WSD) presents a challenge, though not unexpectedly. In this work, we assume the general connotation of each word over statistically prevailing senses, leaving a more cautious handling of WSD as future work.

[4]In general, polysemous words do not seem to have conflicting non-neutral polarities over different senses, though there are many exceptions, e.g., *"heat"*, or *"fine"*. We treat each word in each part-of-speech as a separate word to reduce such cases, otherwise aim to learn the most prevalent polarity in the corpus with respect to each part-of-speech of each word.

[5]Available at `http://www.cs.stonybrook.edu/~ychoi/connotation`.

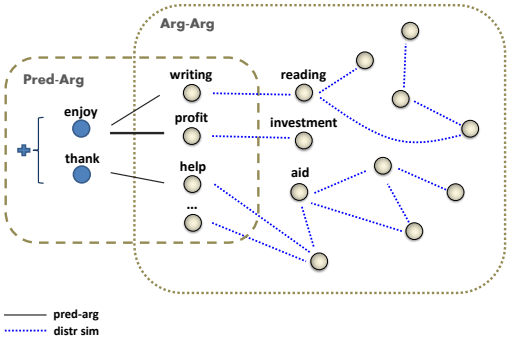

Figure 1: Graph for Graph Propagation (§2.2).

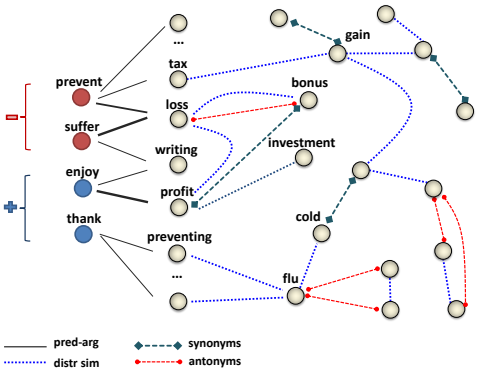

Figure 2: Graph for ILP/LP (§2.3, §4.2).

2 Connotation Induction Algorithms

We develop induction algorithms based on three distinct types of algorithmic framework that have been shown successful for the analogous task of sentiment lexicon induction: HITS & PageRank (§2.1), Label/Graph Propagation (§2.2), and Constraint Optimization via Integer Linear Programming (§2.3). As will be shown, each of these approaches will incorporate additional, more diverse linguistic insights.

2.1 HITS & PageRank

The work of Feng et al. (2011) explored the use of HITS (Kleinberg, 1999) and PageRank (Page et al., 1999) to induce the general connotation of words hinging on the linguistic phenomena of selectional preference and semantic prosody, i.e., *connotative predicates* influencing the connotation of their arguments. For example, the object of a negative connotative predicate *"cure"* is likely to have negative connotation, e.g., *"disease"* or *"cancer"*. The bipartite graph structure for this approach corresponds to the left-most box (labeled as "pred-arg") in Figure 1.

2.2 Label Propagation

With the goal of obtaining a broad-coverage lexicon in mind, we find that relying only on the structure of semantic prosody is limiting, due to relatively small sets of connotative predicates available.[6] Therefore, we extend the graph structure as an *overlay of two sub-graphs* (Figure 1) as described below:

[6]For connotative predicates, we use the seed predicate set of Feng et al. (2011), which comprises of 20 positive and 20 negative predicates.

Sub-graph #1: Predicate–Argument Graph

This sub-graph is the bipartite graph that encodes the selectional preference of connotative predicates over their arguments. In this graph, connotative predicates p reside on one side of the graph and their co-occurring arguments a reside on the other side of the graph based on Google Web 1T corpus.[7] The weight on the edges between the predicates p and arguments a are defined using Point-wise Mutual Information (PMI) as follows:

$$w(p \rightarrow a) := PMI(p, a) = \log_2 \frac{P(p, a)}{P(p)P(a)}$$

PMI scores have been widely used in previous studies to measure association between words (e.g., Turney (2001), Church and Hanks (1990)).

Sub-graph #2: Argument–Argument Graph

The second sub-graph is based on the distributional similarities among the arguments. One possible way of constructing such a graph is simply connecting all nodes and assign edge weights proportionate to the word association scores, such as PMI, or distributional similarity. However, such a completely connected graph can be susceptible to propagating noise, and does not scale well over a very large set of vocabulary.

We therefore reduce the graph connectivity by exploiting *semantic parallelism of coordination* (Bock (1986), Hatzivassiloglou and McKeown

[7]We restrict predicte-argument pairs to verb-object pairs in this study. Note that Google Web 1T dataset consists of n-grams upto $n = 5$. Since n-gram sequences are too short to apply a parser, we extract verb-object pairs approximately by matching part-of-speech tags. Empirically, when overlaid with the second sub-graph, we found that it is better to keep the connectivity of this sub-graph as uni-directional. That is, we only allow edges to go from a predicate to an argument.

	POSITIVE	NEGATIVE	NEUTRAL
n.	avatar, adrenaline, keynote, debut, stakeholder, sunshine, cooperation	unbeliever, delay, shortfall, gunshot, misdemeanor, mutiny, rigor	header, mark, clothing, outline, grid, gasoline, course, preview
v.	handcraft, volunteer, party, accredit, personalize, nurse, google	sentence, cough, trap, scratch, debunk, rip, misspell, overcharge	state, edit, send, put, arrive, type, drill, name, stay, echo, register
a.	floral, vegetarian, prepared, ageless, funded, contemporary	debilitating, impaired, swollen, intentional, jarring, unearned	same, cerebral, west, uncut, automatic, hydrated, unheated, routine

Table 2: Example Words with Learned Connotation: Nouns(n), Verbs(v), Adjectives(a).

(1997), Pickering and Branigan (1998)). In particular, we consider an undirected edge between a pair of arguments a_1 and a_2 only if they occurred together in the "a_1 *and* a_2" or "a_2 *and* a_1" coordination, and assign edge weights as:

$$w(a_1 - a_2) = CosineSim(\overrightarrow{a_1}, \overrightarrow{a_2}) = \frac{\overrightarrow{a_1} \cdot \overrightarrow{a_2}}{||\overrightarrow{a_1}||\ ||\overrightarrow{a_2}||}$$

where $\overrightarrow{a_1}$ and $\overrightarrow{a_2}$ are co-occurrence vectors for a_1 and a_2 respectively. The co-occurrence vector for each word is computed using PMI scores with respect to the top n co-occurring words.[8] n (=50) is selected empirically. The edge weights in two sub-graphs are normalized so that they are in the comparable range.[9]

Limitations of Graph-based Algorithms

Although graph-based algorithms (§2.1, §2.2) provide an intuitive framework to incorporate various lexical relations, limitations include:

1. They allow only *non-negative* edge weights. Therefore, we can encode only positive (supportive) relations among words (e.g., distributionally similar words will endorse each other with the same polarity), while missing on exploiting negative relations (e.g., antonyms may drive each other into the opposite polarity).
2. They induce positive and negative polarities in isolation via separate graphs. However, we expect that a more effective algorithm should induce both polarities simultaneously.
3. The framework does not readily allow incorporating a diverse set of *soft* and *hard* constraints.

[8]We discard edges with cosine similarity ≤ 0, as those indicate either independence or the opposite of similarity.

[9]Note that cosine similarity does not make sense for the first sub-graph as there is no reason why a predicate and an argument should be distributionally similar. We experimented with many different variations on the graph structure and edge weights, including ones that include any word pairs that occurred frequently enough together. For brevity, we present the version that achieved the best results here.

2.3 Constraint Optimization

Addressing limitations of graph-based algorithms (§2.2), we propose an induction algorithm based on Integer Linear Programming (ILP). Figure 2 provides the pictorial overview. In comparison to Figure 1, two new components are: (1) dictionary-driven relations targeting enhanced *precision*, and (2) dictionary-driven words (i.e., unseen words with respect to those relations explored in Figure 1) targeting enhanced *coverage*. We formulate insights in Figure 2 using ILP as follows:

Definition of sets of words:

1. $\mathcal{P}^+$: the set of positive seed predicates.
   $\mathcal{P}^-$: the set of negative seed predicates.
2. $\mathcal{S}$: the set of seed sentiment words.
3. $\mathcal{R}^{syn}$: word pairs in synonyms relation.
   $\mathcal{R}^{ant}$: word pairs in antonyms relation.
   $\mathcal{R}^{coord}$: word pairs in coordination relation.
   $\mathcal{R}^{pred}$: word pairs in pred-arg relation.
   $\mathcal{R}^{pred^{+(-)}}$: $\mathcal{R}^{pred}$ based on $\mathcal{P}^+$ ($\mathcal{P}^-$).

Definition of variables: For each word i, we define binary variables $x_i, y_i, z_i \in \{0, 1\}$, where $x_i = 1$ ($y_i = 1$, $z_i = 1$) if and only if i has a positive (negative, neutral) connotation respectively. For every pair of word i and j, we define binary variables d_{ij}^{pq} where $p, q \in \{+, -, 0\}$ and $d_{ij}^{pq} = 1$ if and only if the polarity of i and j are p and q respectively.

Objective function: We aim to maximize:

$$F = \Phi^{prosody} + \Phi^{coord} + \Phi^{neu}$$

where $\Phi^{prosody}$ is the scores based on semantic prosody, Φ^{coord} captures the distributional similarity over coordination, and Φ^{neu} controls the sensitivity of connotation detection between positive (negative) and neutral. In particular,

$$\Phi^{prosody} = \sum_{i,j}^{\mathcal{R}^{pred}} w_{i,j}^{pred} \left(d_{i,j}^{++} + d_{i,j}^{--} - d_{i,j}^{+-} - d_{i,j}^{-+}\right)$$

$$\Phi^{coord} = \sum_{i,j}^{\mathcal{R}^{coord}} w_{i,j}^{coord} \left(d_{i,j}^{++} + d_{i,j}^{--} + d_{i,j}^{00}\right)$$

$$\Phi^{neu} = \alpha \sum_{i,j}^{\mathcal{R}^{pred}} w_{i,j}^{pred} \cdot z_j$$

Soft constraints (edge weights): The weights in the objective function are set as follows:

$$w^{pred}(p,a) = \frac{freq(p,a)}{\sum_{(p,x)\in\mathcal{R}^{pred}} freq(p,x)}$$

$$w^{coord}(a_1,a_2) = CosSim(\overrightarrow{a_1},\overrightarrow{a_2}) = \frac{\overrightarrow{a_1}\cdot\overrightarrow{a_2}}{||\overrightarrow{a_1}||\,||\overrightarrow{a_2}||}$$

Note that the same $w^{coord}(a_1, a_2)$ has been used in graph propagation described in Section 2.2. α controls the sensitivity of connotation detection such that higher value of α will promote neutral connotation over polar ones.

Hard constrains for variable consistency:

1. Each word i has one of $\{+, -, \emptyset\}$ as polarity: $\forall i,\ x_i + y_i + z_i = 1$
2. Variable consistency between d_{ij}^{pq} and x_i, y_i, z_i:

$$\begin{aligned} x_i + x_j - 1 &\leq 2d_{i,j}^{++} \leq x_i + x_j \\ y_i + y_j - 1 &\leq 2d_{i,j}^{--} \leq y_i + y_j \\ z_i + z_j - 1 &\leq 2d_{i,j}^{00} \leq z_i + z_j \\ x_i + y_j - 1 &\leq 2d_{i,j}^{+-} \leq x_i + y_j \\ y_i + x_j - 1 &\leq 2d_{i,j}^{-+} \leq y_i + x_j \end{aligned}$$

Hard constrains for WordNet relations:

1. $\mathcal{C}^{ant}$: Antonym pairs will not have the same positive or negative polarity:

$$\forall (i,j) \in \mathcal{R}^{ant},\ x_i + x_j \leq 1,\ y_i + y_j \leq 1$$

   For this constraint, we only consider antonym pairs that share the same root, e.g., *"sufficient"* and *"insufficient"*, as those pairs are more likely to have the opposite polarities than pairs without sharing the same root, e.g., *"east"* and *"west"*.

2. $\mathcal{C}^{syn}$: Synonym pairs will not have the opposite polarity:

$$\forall (i,j) \in \mathcal{R}^{syn},\ x_i + y_j \leq 1,\ x_j + y_i \leq 1$$

3 Experimental Result I

We provide comprehensive comparisons over variants of three types of algorithms proposed in §2. We use the Google Web 1T data (Brants and Franz (2006)), and POS-tagged ngrams using Stanford POS Tagger (Toutanova and Manning (2000)). We filter out the ngrams with punctuations and other special characters to reduce the noise.

3.1 Comparison against Conventional Sentiment Lexicon

Note that we consider the connotation lexicon to be inclusive of a sentiment lexicon for two practical reasons: first, it is highly unlikely that any word with non-neutral sentiment (i.e., positive or negative) would carry connotation of the opposite, i.e., conflicting[10] polarity. Second, for some words with distinct sentiment or strong connotation, it can be difficult or even unnatural to draw a precise distinction between connotation and sentiment, e.g., *"efficient"*. Therefore, sentiment lexicons can serve as a surrogate to measure a subset of connotation words induced by the algorithms, as shown in Table 3 with respect to General Inquirer (Stone and Hunt (1963)) and MPQA (Wilson et al. (2005b)).[11]

Discussion Table 3 shows the agreement statistics with respect to two conventional sentiment lexicons. We find that the use of label propagation alone [PRED-ARG (CP)] improves the performance substantially over the comparable graph construction with different graph analysis algorithms, in particular, HITS and PageRank approaches of Feng et al. (2011). The two completely connected variants of the graph propagation on the Pred-Arg graph, [⊗ PRED-ARG (PMI)] and [⊗ PRED-ARG (CP)], do not necessarily improve the performance over the simpler and computationally lighter alternative, [PRED-ARG (CP)]. The [OVERLAY], which is based on both Pred-Arg and Arg-Arg subgraphs (§2.2), achieves the best performance among graph-based algorithms, significantly improving the precision over all other baselines. This result suggests:

1 The sub-graph #2, based on the semantic parallelism of coordination, is simple and yet very powerful as an inductive bias.

2 The performance of graph propagation varies significantly depending on the graph topology and the corresponding edge weights.

Note that a direct comparison against ILP for top N words is tricky, as ILP does not *rank* results. Only for comparison purposes however, we assign

[10] We consider "positive" and "negative" polarities conflict, but "neutral" polarity does *not* conflict with any.

[11] In the case of General Inquirer, we use words in POSITIV and NEGATIV sets as words with positive and negative labels respectively.

	GENINQ EVAL					MPQA EVAL				
	100	1,000	5,000	10,000	ALL	100	1,000	5,000	10,000	ALL
ILP	**97.6**	94.5	**84.5**	80.8	80.4	**98.0**	89.7	**84.6**	**81.2**	**78.4**
OVERLAY	97.0	**95.1**	78.8	(78.3)	78.3	**98.0**	**93.4**	82.1	77.7	77.7
⊗ PRED-ARG (PMI)	91.0	91.4	76.1	(76.1)	76.1	88.0	89.1	78.8	75.1	75.1
⊗PRED-ARG (CP)	88.0	85.4	76.2	(76.2)	76.2	87.0	82.6	78.0	76.3	76.3
PRED-ARG (CP)	91.0	91.0	81.0	**(81.0)**	**81.0**	88.0	91.5	80.0	78.3	78.3
HITS-ASYMT	77.0	68.8	-	-	66.5	86.3	81.3	-	-	72.2
PAGERANK-ASYMF	77.0	68.5	-	-	65.7	87.2	80.3	-	-	72.3

Table 3: Evaluation of Induction Algorithms (§2) with respect to Sentiment Lexicons (precision%).

ranks based on the frequency of words for ILP. Because of this issue, the performance of top ∼1k words of ILP should be considered only as a conservative measure. Importantly, when evaluated over more than top 5k words, ILP is overall the top performer considering both precision (shown in Table 3) and coverage (omitted for brevity).[12]

4 Precision, Coverage, and Efficiency

In this section, we address three important aspects of an ideal induction algorithm: *precision*, *coverage*, and *efficiency*. For brevity, the remainder of the paper will focus on the algorithms based on constraint optimization, as it turned out to be the most effective one from the empirical results in §3.

Precision In order to see the effectiveness of the induction algorithms more sharply, we had used a limited set of seed words in §3. However to build a lexicon with substantially enhanced precision, we will use as a large seed set as possible, e.g., entire sentiment lexicons[13].

Broad coverage Although statistics in Google 1T corpus represent a very large amount of text, words that appear in pred-arg and coordination relations are still limited. To substantially increase the coverage, we will leverage dictionary words (that are not in the corpus) as described in §2.3 and Figure 2.

Efficiency One practical problem with ILP is efficiency and scalability. In particular, we found that it becomes nearly impractical to run the ILP formulation including all words in WordNet plus all words in the argument position in Google Web 1T. We therefore explore an alternative approach based on Linear Programming in what follows.

[12]In fact, the performance of PRED-ARG variants for top 10K w.r.t. GENINQ is not meaningful as no additional word was matched beyond top 5k words.

[13]Note that doing so will prevent us from evaluating against the same sentiment lexicon used as a seed set.

4.1 Induction using Linear Programming

One straightforward option for Linear Programming formulation may seem like using the same Integer Linear Programming formulation introduced in §2.3, only changing the variable definitions to be real values $\in [0, 1]$ rather than integers. However, because the hard constraints in §2.3 are defined based on the assumption that all the variables are binary integers, those constraints are not as meaningful when considered for real numbers. Therefore we revise those hard constraints to encode various semantic relations (WordNet and semantic coordination) more directly.

Definition of variables: For each word i, we define variables $x_i, y_i, z_i \in [0, 1]$. i has a positive (negative) connotation if and only if the x_i (y_i) is assigned the greatest value among the three variables; otherwise, i is neutral.

Objective function: We aim to maximize:

$$F = \Phi^{prosody} + \Phi^{coord} + \Phi^{syn} + \Phi^{ant} + \Phi^{neu}$$

$$\Phi^{prosody} = \sum_{i,j}^{\mathcal{R}^{pred^+}} w_{i,j}^{pred^+} \cdot x_j + \sum_{i,j}^{\mathcal{R}^{pred^-}} w_{i,j}^{pred^-} \cdot y_j$$

$$\Phi^{coord} = \sum_{i,j}^{\mathcal{R}^{coord}} w_{i,j}^{coord} \cdot (dc_{i,j}^{++} + dc_{i,j}^{--})$$

$$\Phi^{syn} = W^{syn} \sum_{i,j}^{\mathcal{R}^{syn}} (ds_{i,j}^{++} + ds_{i,j}^{--})$$

$$\Phi^{ant} = W^{ant} \sum_{i,j}^{\mathcal{R}^{ant}} (da_{i,j}^{++} + da_{i,j}^{--})$$

$$\Phi^{neu} = \alpha \sum_{i,j}^{\mathcal{R}^{pred}} w_{i,j}^{pred} \cdot z_j$$

Hard constraints We add penalties to the objective function if the polarity of a pair of words is not consistent with its corresponding semantic relations. For example, for synonyms i and j, we introduce a penalty W^{syn} (a positive constant) for $ds_{i,j}^{++}, ds_{i,j}^{--} \in [-1, 0]$, where we set the upper bound of $ds_{i,j}^{++}$ ($ds_{i,j}^{--}$) as the signed distance of

	FORMULA	POSITIVE			NEGATIVE			ALL		
		R	P	F	R	P	F	R	P	F
ILP	$\Phi^{prosody} + \mathcal{C}^{syn} + \mathcal{C}^{ant}$	51.4	85.7	64.3	44.7	87.9	59.3	48.0	86.8	61.8
	$\Phi^{prosody} + \mathcal{C}^{syn} + \mathcal{C}^{ant} + \mathcal{C}^{\mathcal{S}}$	61.2	93.3	73.9	52.4	**92.2**	66.8	56.8	**92.8**	70.5
	$\Phi^{prosody} + \Phi^{coord} + \mathcal{C}^{syn} + \mathcal{C}^{ant}$	67.3	75.0	70.9	53.7	84.4	65.6	60.5	79.7	68.8
	$\Phi^{prosody} + \Phi^{coord} + \mathcal{C}^{syn} + \mathcal{C}^{ant} + \mathcal{C}^{\mathcal{S}}$	62.2	**96.0**	75.5	51.5	89.5	65.4	56.9	**92.8**	70.5
LP	$\Phi^{prosody} + \Phi^{syn} + \Phi^{ant}$	24.4	76.0	36.9	23.6	78.8	36.3	24.0	77.4	36.6
	$\Phi^{prosody} + \Phi^{syn} + \Phi^{ant} + \Phi^{\mathcal{S}}$	71.6	87.8	78.9	68.8	84.6	75.9	70.2	86.2	77.4
	$\Phi^{prosody} + \Phi^{coord} + \Phi^{syn} + \Phi^{ant}$	67.9	92.6	78.3	64.6	89.1	74.9	66.3	90.8	76.6
	$\Phi^{prosody} + \Phi^{coord} + \Phi^{syn} + \Phi^{ant} + \Phi^{\mathcal{S}}$	**78.6**	90.5	**84.1**	**73.3**	87.1	**79.6**	**75.9**	88.8	**81.8**

Table 4: ILP/LP Comparison on MQPA′ (%).

x_i and x_j (y_i and y_j) as shown below:

For $(i,j) \in \mathcal{R}^{syn}$,

$$ds^{++}_{i,j} \leq x_i - x_j, \quad ds^{++}_{i,j} \leq x_j - x_i$$
$$ds^{--}_{i,j} \leq y_i - y_j, \quad ds^{--}_{i,j} \leq y_j - y_i$$

Notice that $ds^{++}_{i,j}, ds^{--}_{i,j}$ satisfying above inequalities will be always of negative values, hence in order to maximize the objective function, the LP solver will try to minimize the absolute values of $ds^{++}_{i,j}, ds^{--}_{i,j}$, effectively pushing i and j toward the same polarity. Constraints for semantic coordination $\mathcal{R}^{coord}$ can be defined similarly. Lastly, following constraints encode antonym relations:

For $(i,j) \in \mathcal{R}^{ant}$,

$$da^{++}_{i,j} \leq x_i - (1 - x_j), \quad da^{++}_{i,j} \leq (1 - x_j) - x_i$$
$$da^{--}_{i,j} \leq y_i - (1 - y_j), \quad da^{--}_{i,j} \leq (1 - y_j) - y_i$$

Interpretation Unlike ILP, some of the variables result in fractional values. We consider a word has positive or negative polarity only if the assignment indicates 1 for the corresponding polarity and 0 for the rest. In other words, we treat all words with fractional assignments over different polarities as neutral. Because the optimal solutions of LP correspond to extreme points in the convex polytope formed by the constraints, we obtain a large portion of words with non-fractional assignments toward non-neutral polarities. Alternatively, one can round up fractional values.

4.2 Empirical Comparisons: ILP v.s. LP

To solve the ILP/LP, we run ILOG CPLEX Optimizer (CPLEX, 2009)) on a 3.5GHz 6 core CPU machine with 96GB RAM. Efficiency-wise, LP runs within 10 minutes while ILP takes several hours. Table 4 shows the results evaluated against MPQA for different variations of ILP and LP. We find that LP variants much better recall and F-score, while maintaining comparable precision. Therefore, we choose the connotation lexicon by LP (C-LP) in the following evaluations in §5.

5 Experimental Results II

In this section, we present comprehensive intrinsic §5.1 and extrinsic §5.2 evaluations comparing three representative lexicons from §2 & §4: C-LP, OVERLAY, PRED-ARG (CP), and two popular sentiment lexicons: SentiWordNet (Baccianella et al., 2010) and GI+MPQA.[14] Note that C-LP is the largest among all connotation lexicons, including ∼71,000 polar words.[15]

5.1 Intrinsic Evaluation: Human Judgements

We evaluate 4000 words[16] using Amazon Mechanical Turk (AMT). Because we expect that judging a connotation can be dependent on one's cultural background, personality and value systems, we gather judgements from 5 people for each word, from which we hope to draw a more general judgement of connotative polarity. About 300 unique Turkers participated the evaluation tasks. We gather gold standard only for those words for which more than half of the judges agreed on the same polarity. Otherwise we treat them as ambiguous cases.[17] Figure 3 shows a part of the AMT task, where Turkers are presented with questions that help judges to determine the subtle connotative polarity of each word, then asked to rate the degree of connotation on a scale from -5 (most negative) and 5 (most positive). To draw

[14] GI+MPQA is the union of General Inquirer and MPQA. The GI, we use words in the "Positiv" & "Negativ" set. For SentiWordNet, to retrieve the polarity of a given word, we sum over the polarity scores over all senses, where positive (negative) values correspond to positive (negative) polarity.

[15] ∼14k adj, ∼6k verbs, ∼29k nouns, ∼22k proper nouns.

[16] We choose words that are not already in GI+MPQA and obtain most frequent 10,000 words based on the unigram frequency in Google-Ngram, then randomly select 4000 words.

[17] We allow Turkers to mark words that can be used with both positive and negative connotation, which results in about 7% of words that are excluded from the gold standard set.

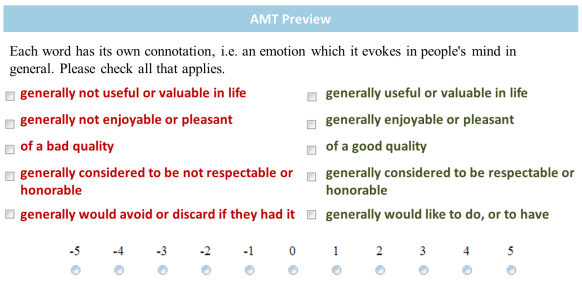

Figure 3: A Part of AMT Task Design.

	YES		NO	
QUESTION	%	Avg	%	Avg
"Enjoyable or pleasant"	43.3	2.9	16.3	-2.4
"Of a good quality"	56.7	2.5	6.1	**-2.7**
"Respectable / honourable"	21.0	**3.3**	14.0	-1.1
"Would like to do or have"	52.5	2.8	11.5	-2.4

Table 5: Distribution of Answers from AMT.

the gold standard, we consider two different voting schemes:

- Ω^{Vote}: The judgement of each Turker is mapped to neutral for $-1 \leq$ score ≤ 1, positive for score ≥ 2, negative for score ≤ 2, then we take the majority vote.
- Ω^{Score}: Let $\sigma(i)$ be the sum (weighted vote) of the scores given by 5 judges for word i. Then we determine the polarity label $l(i)$ of i as:

$$l(i) = \begin{cases} positive & \text{if } \sigma(i) > 1 \\ negative & \text{if } \sigma(i) < -1 \\ neutral & \text{if } -1 \leq \sigma(i) \leq 1 \end{cases}$$

The resulting distribution of judgements is shown in Table 5 & 6. Interestingly, we observe that *among the relatively frequently used English words, there are overwhelmingly more positively connotative words than negative ones.*

In Table 7, we show the percentage of words with the same label over the mutual words by the two lexicon. The highest agreement is 77% by C-LP and the gold standard by AMT^{Vote}. How good is this? It depends on what is the natural degree of agreement over subtle connotation among people. Therefore, we also report the degree of agreement among human judges in Table 7, where we compute the agreement of one Turker with respect to the gold standard drawn from the rest of the Turkers, and take the average across over all five Turkers[18]. Interestingly, the performance of

[18]In order to draw the gold standard from the 4 remaining Turkers, we consider adjusted versions of Ω^{Vote} and Ω^{Score} schemes described above.

	POS	NEG	NEU	UNDETERMINED
Ω^{Vote}	50.4	14.6	24.1	10.9
Ω^{Score}	67.9	20.6	11.5	n/a

Table 6: Distribution of Connotative Polarity from AMT.

	C-LP	SENTIWN	HUMAN JUDGES
Ω^{Vote}	77.0	71.5	66.0
Ω^{Score}	73.0	69.0	69.0

Table 7: Agreement (Accuracy) against AMT-driven Gold Standard.

Turkers is not as good as that of C-LP lexicon. We conjecture that this could be due to generally varying perception of different people on the connotative polarity,[19] while the corpus-driven induction algorithms focus on the *general* connotative polarity corresponding to the most prevalent senses of words in the corpus.

5.2 Extrinsic Evaluation

We conduct lexicon-based binary sentiment classification on the following two corpora.

SemEval From the SemEval task, we obtain a set of news headlines with annotated scores (ranging from -100 to 87). The positive/negative scores indicate the degree of positive/negative polarity orientation. We construct several sets of the positive and negative texts by setting thresholds on the scores as shown in Table 8. "$\lessgtr n$" indicates that the positive set consists of the texts with scores $\geq n$ and the negative set consists of the texts with scores $\leq -n$.

Emoticon tweets The sentiment Twitter data[20] consists of tweets containing either a smiley emoticon (positive sentiment) or a frowny emoticon (negative sentiment). We filter out the tweets with question marks or more than 30 words, and keep the ones with at least two words in the union of all polar words in the five lexicons in Table 8, and then randomly select 10000 per class.

We denote the short text (e.g., content of tweets or headline texts from SemEval) by t. w represents the word in t. W^+/W^- is the set of posi-

[19]Pearson correlation coefficient among turkers is 0.28, which corresponds to a positive small to medium correlation. Note that when the annotation of turkers is aggregated, we observe agreement as high as 77% with respect to the learned connotation lexicon.

[20]http://www.stanford.edu/~alecmgo/cs224n/twitterdata.2009.05.25.c.zip

LEXICON	DATA				
	TWEET	SEMEVAL			
		⩽20	⩽40	⩽60	⩽80
C-LP	**70.1**	**70.8**	**74.6**	**80.8**	**93.5**
OVERLAY	68.5	70.0	72.9	76.8	89.6
PRED-ARG (CP)	60.5	64.2	69.3	70.3	79.2
SENTIWN	67.4	61.0	64.5	70.5	79.0
GI+MPQA	65.0	64.5	69.0	74.0	80.5

Table 8: Accuracy on Sentiment Classification (%).

tive/negative words of the lexicon. We define the weight of w as $s(w)$. If w is adjective, $s(w) = 2$; otherwise $s(w) = 1$. Then the polarity of each text is determined as follows:

$$pol(t) = \begin{cases} positive & \text{if } \sum_{w \in t}^{W^+} s(w) \geq \sum_{w \in t}^{W^-} s(w) \\ negative & \text{if } \sum_{w \in t}^{W^+} s(w) < \sum_{w \in t}^{W^-} s(w) \end{cases}$$

As shown in Table 8, C-LP generally performs better than the other lexicons on both corpora. Considering that only very simple classification strategy is applied, the result by the connotation lexicon is quite promising.

Finally, Table 1 highlights interesting examples of proper nouns with connotative polarity, e.g., *"Mandela"*, *"Google"*, *"Hawaii"* with positive connotation, and *"Monsanto"*, *"Halliburton"*, *"Enron"* with negative connotation, suggesting that our algorithms could potentially serve as a proxy to track the general connotation of real world entities. Table 2 shows example proper nouns with connotative polarity.

5.3 Practical Remarks on WSD and MWEs

In this work we aim to find the polarity of most prevalent senses of each word, in part because it is not easy to perform unsupervised word sense disambiguation (WSD) on a large corpus in a reliable way, especially when the corpus consists primarily of short n-grams. Although the resulting lexicon loses on some of the polysemous words with potentially opposite polarities, per-word connotation (rather than per-sense connotation) does have a practical value: it provides a convenient option for users who wish to avoid the burden of WSD before utilizing the lexicon. Future work includes handling of WSD and multi-word expressions (MWEs), e.g., *"Great Leader"* (for Kim Jong-Il), *"Inglourious Basterds"* (a movie title).[21]

[21] These examples credit to an anonymous reviewer.

6 Related Work

A very interesting work of Mohammad and Turney (2010) uses Mechanical Turk in order to build the lexicon of emotions evoked by words. In contrast, we present an automatic approach that infers the general connotation of words. Velikovich et al. (2010) use graph propagation algorithms for constructing a web-scale polarity lexicon for sentiment analysis. Although we employ the same graph propagation algorithm, our graph construction is fundamentally different in that we integrate stronger inductive biases into the graph topology and the corresponding edge weights. As shown in our experimental results, we find that judicious construction of graph structure, exploiting multiple complementing linguistic phenomena can enhance both the performance and the efficiency of the algorithm substantially. Other interesting approaches include one based on min-cut (Dong et al., 2012) or LDA (Xie and Li, 2012). Our proposed approaches are more suitable for encoding a much diverse set of linguistic phenomena however. But our work use a few seed predicates with selectional preference instead of relying on word similarity. Some recent work explored the use of constraint optimization framework for inducing domain-dependent sentiment lexicon (Choi and Cardie (2009), Lu et al. (2011)). Our work differs in that we provide comprehensive insights into different formulations of ILP and LP, aiming to learn the much different task of learning the general connotation of words.

7 Conclusion

We presented a broad-coverage connotation lexicon that determines the subtle nuanced sentiment of even those words that are objective on the surface, including the general connotation of real-world named entities. Via a comprehensive evaluation, we provided empirical insights into three different types of induction algorithms, and proposed one with good precision, coverage, and efficiency.

Acknowledgments

This research was supported in part by the Stony Brook University Office of the Vice President for Research. We thank reviewers for many insightful comments and suggestions, and for providing us with several very inspiring examples to work with.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

J. Kathryn Bock. 1986. Syntactic persistence in language production. *Cognitive psychology*, 18(3):355–387.

Thorsten Brants and Alex Franz. 2006. {Web 1T 5-gram Version 1}.

Yejin Choi and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 590–598, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16:22–29, March.

ILOG CPLEX. 2009. High-performance software for mathematical programming and optimization. *U RL http://www.ilog.com/products/cplex*.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 107–116, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xishuang Dong, Qibo Zou, and Yi Guan. 2012. Set-similarity joins based semi-supervised sentiment analysis. In *Neural Information Processing*, pages 176–183. Springer.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, pages 417–422.

Song Feng, Ritwik Bose, and Yejin Choi. 2011. Learning general connotation of words using graph-based algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1092–1103. Association for Computational Linguistics.

Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 503–511, Boulder, Colorado, June. Association for Computational Linguistics.

Vasileios Hatzivassiloglou and Kathleen R McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 174–181. Association for Computational Linguistics.

Bas Heerschop, Alexander Hogenboom, and Flavius Frasincar. 2011. Sentiment lexicon creation from lexical resources. In *Business Information Systems*, pages 185–196. Springer.

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

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *JOURNAL OF THE ACM*, 46(5):604–632.

Bill Louw. 1993. Irony in the text or insincerity in the writer. *Text and technology: In honour of John Sinclair*, pages 157–176.

Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of the 20th international conference on World wide web*, pages 347–356. ACM.

Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles, CA, June. Association for Computational Linguistics.

Arturo Montejo-Ráez, Eugenio Martínez-Cámara, M. Teresa Martín-Valdivia, and L. Alfonso Ureña López. 2012. Random walk weighting over sentiwordnet for sentiment polarity detection on twitter. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 3–10, Jeju, Korea, July. Association for Computational Linguistics.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

Martin J Pickering and Holly P Branigan. 1998. The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and Language*, 39(4):633–651.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 1199–1204, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dan Roth and Wen-tau Yih. 2004. *A linear programming formulation for global inference in natural language tasks*. Defense Technical Information Center.

John Sinclair. 1991. *Corpus, concordance, collocation*. Describing English language. Oxford University Press.

Anatol Stefanowitsch and Stefan Th Gries. 2003. Collostructions: Investigating the interaction of words and constructions. *International journal of corpus linguistics*, 8(2):209–243.

Philip J. Stone and Earl B. Hunt. 1963. A computer approach to content analysis: studies using the general inquirer system. In *Proceedings of the May 21-23, 1963, spring joint computer conference*, AFIPS '63 (Spring), pages 241–256, New York, NY, USA. ACM.

Michael Stubbs. 1995. Collocations and semantic profiles: on the cause of the trouble with quantitative studies. *Functions of language*, 2(1):23–55.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *In EMNLP/VLC 2000*, pages 63–70.

Peter Turney. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation (formerly Computers and the Humanities)*, 39(2/3):164–210.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005a. Opinionfinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35, Morristown, NJ, USA. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005b. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Morristown, NJ, USA. Association for Computational Linguistics.

Rui Xie and Chunping Li. 2012. Lexicon construction: A topic model approach. In *Systems and Informatics (ICSAI), 2012 International Conference on*, pages 2299–2303. IEEE.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. In *Technical Report CMU-CALD-02-107*. CarnegieMellon University.