ConnotationWordNet: Learning Connotation over the Word+Sense Network

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

Abstract

We introduce *ConnotationWordNet*, a connotation lexicon over the network of *words* in conjunction with *senses*. We formulate the lexicon induction problem as collective inference over pairwise-Markov Random Fields, and present a loopy belief propagation algorithm for inference. The key aspect of our method is that it is the *first unified* approach that assigns the polarity of *both* word- and sense-level connotations, exploiting the innate bipartite graph structure encoded in WordNet. We present comprehensive evaluation to demonstrate the quality and utility of the resulting lexicon in comparison to existing connotation and sentiment lexicons.

1 Introduction

We introduce *ConnotationWordNet*, a connotation lexicon over the network of *words* in conjunction with *senses*, as defined in WordNet. A connotation lexicon, as introduced first by Feng et al. (2011), aims to encompass subtle shades of sentiment a word may conjure, even for seemingly objective words such as *"sculpture"*, *"Ph.D."*, *"rosettes"*. Understanding the rich and complex layers of connotation remains to be a challenging task. As a starting point, we study a more feasible task of learning the *polarity* of connotation.

For non-polysemous words, which constitute a significant portion of English vocabulary, learning the *general* connotation at the *word-level* (rather than at the *sense-level*) would be a natural operational choice. However, for polysemous words, which correspond to most frequently used words, it would be an overly crude assumption that the same connotative polarity should be assigned for all senses of a given word. For example, consider *"abound"*, for which lexicographers of WordNet prescribe two different senses:

- (v) **abound**: (be abundant of plentiful; exist in large quantities)
- (v) **abound, burst, bristle**: (be in a state of movement or action) *"The room abounded with screaming children"*; *"The garden bristled with toddlers"*

For the first sense, which is the most commonly used sense for *"abound"*, the general overtone of the connotation would seem positive. That is, although one can use this sense in both positive and negative contexts, this sense of *"abound"* seems to collocate more often with items that are good to be abundant (e.g., *"resources"*), than unfortunate items being abundant (e.g., *"complaints"*).

However, as for the second sense, for which *"burst"* and *"bristle"* can be used interchangeably with respect to this particular sense,[1] the general overtone is slightly more negative with a touch of unpleasantness, or at least not as positive as that of the first sense. Especially if we look up the WordNet entry for *"bristle"*, there are noticeably more negatively connotative words involved in its gloss and examples.

This word sense issue has been a universal challenge for a range of Natural Language Processing applications, including sentiment analysis. Recent studies have shown that it is fruitful to tease out subjectivity and objectivity corresponding to different senses of the same word, in order to improve computational approaches to sentiment analysis (e.g. Pestian et al. (2012), Mihalcea et al. (2012) Balahur et al. (2014)). Encouraged by these recent successes, in this study, we investigate if we can attain similar gains if we model the connotative polarity of senses separately.

There is one potential practical issue we would like to point out in building a sense-level lexical resource, however. End-users of such a lexicon may not wish to deal with Word Sense Disam-

[1] Hence a *sense* in WordNet is defined by *synset (= synonym set)*, which is the set of words sharing the same sense.

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 1544–1554,
Baltimore, Maryland, USA, June 23-25 2014.

biguation (WSD), which is known to be often too noisy to be incorporated into the pipeline with respect to other NLP tasks. As a result, researchers often would need to aggregate labels across different senses to derive the word-level label. Although such aggregation is not entirely unreasonable, it does not seem to be the most optimal and principled way of integrating available resources.

Therefore, in this work, we present the first unified approach that learns *both* sense- and word-level connotations simultaneously. This way, end-users will have access to more accurate sense-level connotation labels if needed, while also having access to more general word-level connotation labels. We formulate the lexicon induction problem as collective inference over pairwise-Markov Random Fields (pairwise-MRF) and derive a loopy belief propagation algorithm for inference.

The key aspect of our approach is that we exploit the innate bipartite graph structure between words and senses encoded in WordNet. Although our approach seems conceptually natural, previous approaches, to our best knowledge, have not directly exploited these relations between words and senses for the purpose of deriving lexical knowledge over words and senses collectively. In addition, previous studies (for both sentiment and connotation lexicons) aimed to produce only either of the two aspects of the polarity: word-level or sense-level, while we address both.

Another contribution of our work is the introduction of loopy belief propagation (loopy-BP) as a lexicon induction algorithm. Loopy-BP in our study achieves statistically significantly better performance over the constraint optimization approaches previously explored. In addition, it runs much faster and it is considerably easier to implement. Last but not least, by using probabilistic representation of pairwise-MRF in conjunction with Loopy-BP as inference, the resulting solution has the natural interpretation as the intensity of connotation. This contrasts to approaches that seek discrete solutions such as Integer Linear Programming(Papadimitriou and Steiglitz, 1998).

ConnotationWordNet, the final outcome of our study, is a new lexical resource that has connotation labels over both words and senses following the structure of WordNet. The lexicon is publicly available at: `http://www.cs.sunysb.edu/~junkang/connotation_wordnet`.)

In what follows, we will first describe the network of words and senses (Section 2), then introduce the representation of the network structure as pairwise Markov Random Fields, and a loopy belief propagation algorithm as collective inference (Section 3). We then present comprehensive evaluation (Section 4 & 5 & 6), followed by related work (Section 7) and conclusion (Section 8).

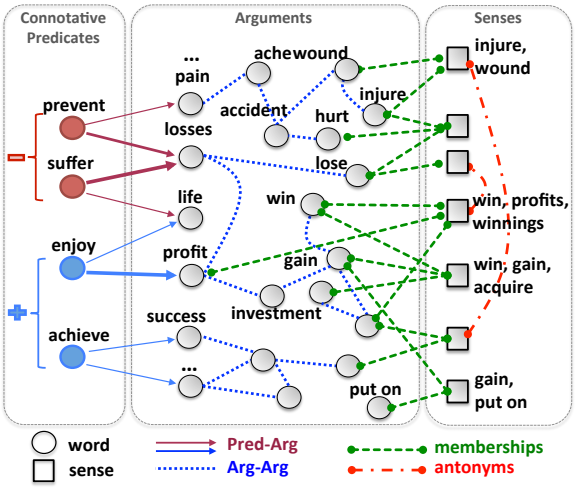


Figure 1: $G^{\text{WORD+SENSE}}$ with words and senses.

2 Network of Words and Senses

The connotation graph, called $G^{\text{WORD+SENSE}}$, is a heterogeneous graph with multiple types of nodes and edges. As shown in Figure 1, it contains two types of nodes; (i) lemmas (i.e., words, 115K) and (ii) synsets (63K), and four types of edges; (t_1) predicate-argument (179K), (t_2) argument-argument (144K), (t_3) argument-synset (126K), and (t_4) synset-synset (3.4K) edges.

The predicate-argument edges, first introduced by Feng et al. (2011), depict the selectional preference of connotative predicates (i.e., the polarity of a predicate indicates the polarity of its arguments) and encode their co-occurrence relations based on the Google Web 1T corpus. The argument-argument edges are based on the distributional similarities among the arguments. The argument-synset edges capture the synonymy between argument nodes through the corresponding synsets. Finally, the synset-synset edges depict the antonym relations between synset pairs.

In general, our graph construction is similar to that of Feng et al. (2013), but there are a few important differences. Most notably, we model both words and synsets explicitly, and exploit the membership relations between words and senses. We expect that edges between words and senses will encourage *senses that belong to the same word* to

receive the same connotation label. Conversely, we expect that these edges will also encourage *words that belong to the same sense* (i.e., synset definition) to receive the same connotation label.

Another benefit of our approach is that for various WordNet relations (e.g., antonym relations), which are defined over synsets (not over words), we can add edges directly between corresponding synsets, rather than projecting (i.e., approximating) those relations over words. Note that the latter, which has been employed by several previous studies (e.g., Kamps et al. (2004), Takamura et al. (2005), Andreevskaia and Bergler (2006), Su and Markert (2009), Lu et al. (2011), Kaji and Kitsuregawa (2007), Feng et al. (2013)), could be a source of noise, as one needs to assume that the semantic relation between a pair of synsets transfers over the pair of words corresponding to that pair of synsets. For polysemous words, this assumption may be overly strong.

3 Pairwise Markov Random Fields and Loopy Belief Propagation

We formulate the task of learning sense- and word-level connotation lexicon as a graph-based classification task (Sen et al., 2008). More formally, we denote the connotation graph $\text{G}^{\text{WORD+SENSE}}$ by $G = (V, E)$, in which a total of n word and synset nodes $V = \{v_1, \ldots, v_n\}$ are connected with typed edges $e(v_i, v_j, t) \in E$, where edge types $t \in \{pred\text{-}arg, arg\text{-}arg, syn\text{-}arg, syn\text{-}syn\}$ depict the four edge types as described in Section 2. A neighborhood function $\mathcal{N}$, where $\mathcal{N}_v = \{u|\ e(u, v) \in E\} \subseteq V$, describes the underlying network structure.

In our collective classification formulation, each node in V is represented as a random variable that takes a value from an appropriate class label domain; in our case, $\mathcal{L} = \{+, -\}$ for positive and negative connotation. In this classification task, we denote by $\mathcal{Y}$ the nodes the labels of which need to be assigned, and let y_i refer to Y_i's label.

3.1 Pairwise Markov Random Fields

We next define our objective function. We propose to use an objective formulation that utilizes pairwise Markov Random Fields (MRFs) (Kindermann and Snell, 1980), which we adapt to our problem setting. MRFs are a class of probabilistic graphical models that are suited for solving inference problems in networked data. An MRF consists of an undirected graph where each node can be in any of a finite number of states (i.e., class labels). The state of a node is assumed to be dependent on each of its neighbors and independent of other nodes in the graph.[2] In pairwise MRFs, the joint probability of the graph can be written as a product of pairwise factors, parameterized over the edges. These factors are referred to as clique potentials in general MRFs, which are essentially functions that collectively determine the graph's joint probability.

Specifically, let $G = (V, E)$ denote a network of random variables, where V consists of the unobserved variables $\mathcal{Y}$ that need to be assigned values from label set $\mathcal{L}$. Let Ψ denote a set of clique potentials that consists of two types of factors:

- For each $Y_i \in \mathcal{Y}$, $\psi_i \in \Psi$ is a *prior* mapping $\psi_i : \mathcal{L} \rightarrow \mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ denotes non-negative real numbers.
- For each $e(Y_i, Y_j, t) \in E$, $\psi_{ij}^t \in \Psi$ is a *compatibility* mapping $\psi_{ij}^t : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}_{\geq 0}$.

Objective formulation Given an assignment $\mathbf{y}$ to all the unobserved variables $\mathcal{Y}$ and $\mathbf{x}$ to observed ones $\mathcal{X}$ (variables with known labels, if any), our objective function is associated with the following joint probability distribution

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{Y_i \in \mathcal{Y}} \psi_i(y_i) \prod_{e(Y_i, Y_j, t) \in E} \psi_{ij}^t(y_i, y_j) \quad (1)$$

where $Z(\mathbf{x})$ is the normalization function. Our goal is then to infer the maximum likelihood assignment of states (i.e., labels) to unobserved variables (i.e., nodes) that will maximize Equation (1).

Problem Definition Having introduced our graph-based classification task and objective formulation, we define our problem more formally.

Given
- a connotation graph $G = (V, E)$ of words and synsets connected with *typed* edges,
- *prior* knowledge (i.e., probabilities) of (some or all) nodes belonging to each class,
- *compatibility* of two nodes with a given pair of labels being connected to each other;

Classify the nodes $Y_i \in \mathcal{Y}$, into one of two classes; $\mathcal{L} = \{+, -\}$, such that the class assignments y_i maximize our objective in Equation (1).

We can further *rank* the network objects by the probability of their connotation polarity.

[2] This assumption yields a *pairwise* Markov Random Field (MRF); a special case of general MRFs (Yedidia et al., 2003).

3.2 Loopy Belief Propagation

Finding the best assignments to unobserved variables in our objective function is the inference problem. The brute force approach through enumeration of all possible assignments is exponential and thus intractable. In general, exact inference is known to be NP-hard and there is no known algorithm which can be theoretically shown to solve the inference problem for general MRFs. Therefore in this work, we employ a computationally tractable (in fact linearly scalable with network size) approximate inference algorithm called Loopy Belief Propagation (LBP) (Yedidia et al., 2003), which we extend to handle typed graphs like our connotation graph.

Our inference algorithm is based on iterative message passing and the core of it can be concisely expressed as the following two equations:

$$m_{i\to j}(y_j) = \alpha \sum_{y_i \in \mathcal{L}} \left(\psi_{ij}^t(y_i, y_j)\ \psi_i(y_i) \prod_{Y_k \in \mathcal{N}_i \cap \mathcal{Y} \setminus Y_j} m_{k\to i}(y_i) \right), \quad \forall y_j \in \mathcal{L} \tag{2}$$

$$b_i(y_i) = \beta\ \psi_i(y_i) \prod_{Y_j \in \mathcal{N}_i \cap \mathcal{Y}} m_{j\to i}(y_i), \forall y_i \in \mathcal{L} \tag{3}$$

A message $m_{i\to j}$ is sent from node i to node j and captures the belief of i about j, which is the probability distribution over the labels of j; i.e. what i "thinks" j's label is, given the current label of i and the *type* of the edge that connects i and j. Beliefs refer to marginal probability distributions of nodes over labels; for example $b_i(y_i)$ denotes the *belief* of node i having label y_i. α and β are the normalization constants, which respectively ensure that each message and each set of marginal probabilities sum to 1. At every iteration, each node computes its belief based on messages received from its neighbors, and uses the compatibility mapping to transform its belief into messages for its neighbors. The key idea is that after enough iterations of message passes between the nodes, the "conversations" are likely to come to a consensus, which determines the marginal probabilities of all the unknown variables.

The pseudo-code of our method is given in Algorithm 1. It first initializes all messages to 1 and *prior*s to unbiased (i.e., equal) probabilities for all nodes except the seed nodes for which the sentiment is known (lines 3-9). It then proceeds by making each $Y_i \in \mathcal{Y}$ communicate messages

Algorithm 1: CONNOTATION INFERENCE

1 **Input:** Connotation graph G=(V, E), prior potentials ψ_s for seed words $s \in S$, and compatibility potentials ψ_{ij}^t
2 **Output:** Connotation label probabilities for each node $i \in V \setminus P$
3 **foreach** $e(Y_i, Y_j, t) \in E$ **do** // *initialize msg.s*
4 **foreach** $y_j \in \mathcal{L}$ **do**
5 $m_{i\to j}(y_j) \leftarrow 1$
6 **foreach** $i \in V$ **do** // *initialize priors*
7 **foreach** $y_j \in \mathcal{L}$ **do**
8 **if** $i \in S$ **then** $\phi_i(y_j) \leftarrow \psi_i(y_j)$ **else** $\phi_i(y_j) \leftarrow 1/|\mathcal{L}|$
9 **repeat** // *iterative message passing*
10 **foreach** $e(Y_i, Y_j, t) \in E$, $Y_j \in \mathcal{Y}^{V\setminus S}$ **do**
11 **foreach** $y_j \in \mathcal{L}$ **do**
12 Use Equation (2)
13 **until** all messages stop changing
14 **foreach** $Y_i \in \mathcal{Y}^{V\setminus S}$ **do** // *compute final beliefs*
15 **foreach** $y_i \in \mathcal{L}$ **do**
16 Use Equation (3)

with their neighbors in an iterative fashion until the messages stabilize (lines 10-14), i.e. convergence is reached.[3] At convergence, we calculate the marginal probabilities, that is of assigning Y_i with label y_i, by computing the final beliefs $b_i(y_i)$ (lines 15-17). We use these maximum likelihood probabilities for label assignment; for each node i, we assign the label $\mathcal{L}_i \leftarrow \max_{y_i} b_i(y_i)$.

To completely define our algorithm, we need to instantiate the potentials Ψ, in particular the priors and the compatibilities, which we discuss next.

Priors The *prior* beliefs ψ_i of nodes can be suitably initialized if there is any prior knowledge for their connotation sentiment (e.g., `enjoy` is positive, `suffer` is negative). As such, our method is flexible to integrate available side information. In case there is no prior knowledge available, each node is initialized equally likely to have any of the possible labels, i.e., $\frac{1}{|\mathcal{L}|}$ as in Algorithm 1 (line 9).

Compatibilities The *compatibility* potentials can be thought of as matrices, with entries

[3] Although convergence is not theoretically guaranteed, in practice LBP converges to beliefs within a small threshold of change (e.g., 10^{-6}) fairly quickly with accurate results (Pandit et al., 2007; McGlohon et al., 2009; Akoglu et al., 2013).

$\psi_{ij}^t(y_i, y_j)$ that give the likelihood of a node having label y_i, given that it has a neighbor with label y_j to which it is connected through a type t edge. A key difference of our method from earlier models is that we use clique potentials that differ for edge types, since the connotation graph is heterogeneous. This is exactly because the compatibility of class labels of two adjacent nodes depends on the type of the edge connecting them: e.g., $+ \xrightarrow{syn\text{-}arg} +$ is highly compatible, whereas $+ \xrightarrow{syn\text{-}syn} +$ is unlikely; as $syn\text{-}arg$ edges capture synonymy; i.e., words-sense memberships, while $syn\text{-}syn$ edges depict antonym relations.

A sample instantiation of the compatibilities is shown in Table 1. Notice that the potentials for $pred\text{-}arg$, $arg\text{-}arg$, and $syn\text{-}arg$ capture homophily, i.e., nodes with the same label are likely to connect to each other through these types of edges.[4] On the other hand, $syn\text{-}syn$ edges connect nodes that are antonyms of each other, and thus the compatibilities capture the reverse relationship among their labels.

Table 1: Instantiation of compatibility potentials. Entry $\psi_{ij}^t(y_i, y_j)$ is the compatibility of a node with label y_i having a neighbor labeled y_j, given the edge between i and j is type t, for small ϵ.

t: t_1	**A**	
P	+	−
+	1-ϵ	ϵ
−	ϵ	1-ϵ

(t_1) $pred\text{-}arg$

t: t_2	**A**	
A	+	−
+	1-2ϵ	2ϵ
−	2ϵ	1-2ϵ

(t_2) $arg\text{-}arg$

t: t_3	**A**	
S	+	−
+	1-ϵ	ϵ
−	ϵ	1-ϵ

(t_3) $syn\text{-}arg$
(synonym relations)

t: t_4	**S**	
S	+	−
+	ϵ	1-ϵ
−	1-ϵ	ϵ

(t_4) $syn\text{-}syn$
(antonym relations)

Complexity analysis Most demanding component of Algorithm 1 is the iterative message passing over the edges (lines 10-14), with time complexity $O(ml^2r)$, where $m = |E|$ is the number of edges in the connotation graph, $l = |\mathcal{L}|$, the classes, and r, the iterations until convergence. Often, l is quite small (in our case, $l = 2$) and $r \ll m$. Thus running time grows linearly with the number of edges and is scalable to large datasets.

[4] $arg\text{-}arg$ edges are based on co-occurrence (see Section 2), which does not carry as strong indication of the same connotation as e.g., synonymy. Thus, we enforce less homophily for nodes connected through edges of $arg\text{-}arg$ type.

4 Evaluation I: Agreement with Sentiment Lexicons

ConnotationWordNet is expected to be the superset of a sentiment lexicon, as it is highly likely for any word with positive/negative sentiment to carry connotation of the same polarity. Thus, we use two conventional sentiment lexicons, General Inquirer (GENINQ) (Stone et al., 1966) and MPQA (Wilson et al., 2005b), as surrogates to measure the performance of our inference algorithm.

4.1 Variants of Graph Construction

The construction of the connotation graph, denoted by $G^{WORD+SENSE}$, which includes words and synsets, has been described in Section 2. In addition to this graph, we tried several other graph constructions, the first three of which have previously been used in (Feng et al., 2013). We briefly describe these graphs below, and compare performance on all the graphs in the proceeding.

G^{WORD} W/ PRED-ARG: This is a (bipartite) subgraph of $G^{WORD+SENSE}$, which only includes the connotative predicates and their arguments. As such, it contains only type t_1 edges. The edges between the predicates and the arguments can be weighted by their Point-wise Mutual Information (PMI)[5] based on the Google Web 1T corpus.

G^{WORD} W/ OVERLAY: The second graph is also a proper subgraph of $G^{WORD+SENSE}$, which includes the predicates and all the argument words. Predicate words are connected to their arguments as before. In addition, argument pairs (a_1, a_2) are connected if they occurred together in the "a_1 and a_2" or "a_2 and a_1" coordination (Hatzivassiloglou and McKeown, 1997; Pickering and Branigan, 1998). This graph contains both type t_1 and t_2 edges. The edges can also be weighted based on the distributional similarities of the word pairs.

G^{WORD}: The third graph is a super-graph of G^{WORD} W/ OVERLAY, with additional edges, where argument pairs in synonym and antonym relation are connected to each other. Note that unlike the connotation graph $G^{WORD+SENSE}$, it does *not* contain any synset nodes. Rather, the words that are synonyms or antonyms of each other are directly linked in the graph. As such, this graph contains all edge types t_1 through t_4.

[5] PMI scores are widely used in previous studies to measure association between words (e.g., (Church and Hanks, 1990), (Turney, 2001), (Newman et al., 2009)).

G$^{\text{WORD+SENSE}}$ W/ SYNSIM: This is a super-graph of our original G$^{\text{WORD+SENSE}}$ graph; that is, it has all the predicate, arguments, and synset nodes, as well as the four types of edges between them. In addition, we add edges of a fifth type t_5 between the synset nodes to capture their similarity. To define similarity, we use the glossary definitions of the synsets and derive three different scores. Each score utilizes the `count`(s_1, s_2) of overlapping nouns, verbs, and adjectives/adverbs among the glosses of the two synsets s_1 and s_2.

G$^{\text{WORD+SENSE}}$ W/ SYNSIM1: We discard edges with `count` less than 3. The weighted version has the `counts` normalized between 0 and 1.

G$^{\text{WORD+SENSE}}$ W/ SYNSIM2: We normalize the `counts` by the length of the gloss (the avg of two lengths), that is, p = `count / avg(len_gloss(`s_1`), len_gloss(`s_2`))` and discard edges with $p < 0.5$. The weighted version contains p values as edge weights.

G$^{\text{WORD+SENSE}}$ W/ SYNSIM3: To further sparsify the graph we discard edges with $p < 0.6$. To weigh the edges, we use the cosine similarity between the gloss vectors of the synsets based on the TF-IDF values of the words the glosses contain.

Note that the connotation inference algorithm, as given in Algorithm 1, remains exactly the same for all the graphs described above. The only difference is the set of parameters used; while G$^{\text{WORD}}$ W/ PRED-ARG and G$^{\text{WORD}}$ W/ OVERLAY contain one and two edge types, respectively and only use compatibilities (t_1) and (t_2), G$^{\text{WORD}}$ uses all four as given in Table 1. The G$^{\text{WORD+SENSE}}$ W/ SYNSIM graphs use an additional compatibility matrix for the synset similarity edges of type t_5, which is the same as the one used for t_1, i.e., similar synsets are likely to have the same connotation label. This flexibility is one of the key advantages of our algorithm as new types of nodes and edges can be added to the graph seamlessly.

4.2 Sentiment-Lexicon based Performance

In this section, we first compare the performance of our connotation graph G$^{\text{WORD+SENSE}}$ to graphs that do not include synset nodes but only words. Then we analyze the performance when the additional synset similarity edges are added. First, we briefly describe our performance measures.

The sentiment lexicons we use as gold standard are small, compared to the size (i.e., number of words) our graphs contain. Thus, we first find the `overlap` between each graph and a sentiment lexicon. Note that the `overlap` size may be smaller than the `lexicon size`, as some sentiment words may be missing from our graphs. Then, we calculate the number of `correct` label assignments. As such, precision is defined as (`correct` / `overlap`), and recall as (`correct` / `lexicon size`). Finally, F1-score is their harmonic mean and reflects the overall accuracy.

	GENINQ			MPQA
	P	**R**	**F**	**F**
Variations of G$^{\text{WORD}}$				
W/ PRED-ARG	88.0	67.6	76.5	57.3
W/ PRED-ARG-W	84.9	68.9	76.1	57.8
W/ OVERLAY	87.8	70.4	78.1	58.4
W/ OVERLAY-W	82.2	67.7	74.2	54.2
G$^{\text{WORD}}$	88.5	83.1	85.7	69.7
G$^{\text{WORD}}$-W	75.5	71.5	73.4	53.2
Variations of G$^{\text{WORD+SENSE}}$				
G$^{\text{WORD+SENSE}}$	**88.8**	**84.1**	**86.4**	**70.0**
G$^{\text{WORD+SENSE}}$-W	76.8	73.0	74.9	54.6
W/ SYNSIM1	87.2	83.3	85.2	67.9
W/ SYNSIM2	83.9	80.8	82.3	65.1
W/ SYNSIM3	86.5	83.2	84.8	67.8
W/ SYNSIM1-W	88.0	**84.3**	86.1	69.2
W/ SYNSIM2-W	86.4	83.7	85.0	68.5
W/ SYNSIM3-W	86.7	83.4	85.0	68.2

Table 2: Connotation inference performance on various graphs. '-W' indicates weighted versions (see §4.1). **P**: precision, **R**: recall, **F**: F1-score (%).

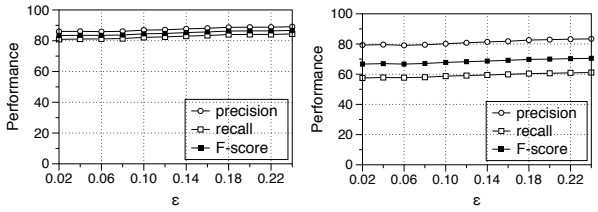
As shown in Table 2 (top), we first observe that including the synonym and antonym relations in the graph, as with G$^{\text{WORD}}$ and G$^{\text{WORD+SENSE}}$, improve the performance significantly, almost by an order of magnitude, over graphs G$^{\text{WORD}}$ W/ PRED-ARG and G$^{\text{WORD}}$ W/ OVERLAY that do not contain those relation types. Furthermore, we notice that the performances on the G$^{\text{WORD+SENSE}}$ graph are better than those on the word-only graphs. This shows that including the synset nodes explicitly in the graph structure is beneficial. What is more, it gives us a means to obtain connotation labels for the synsets themselves, which we use in the evaluations in the next sections. Finally, we note that using the unweighted versions of the graphs provide relatively more robust performance, potentially due to noise in the relative edge weights.

Next we analyze the performance when the new edges between synsets are introduced, as given in Table 2 (bottom). We observe that connecting the synset nodes by their gloss-similarity (at least in the ways we tried) does not yield better performance than on our original G$^{\text{WORD+SENSE}}$ graph. Different from earlier, the weighted versions of the similarity based graphs provide better perfor-

mance than their unweighted counterparts. This suggests that glossary similarity would be a more robust means to correlate nodes; we leave it as future work to explore this direction for predicate-argument and argument-argument relations.

4.3 Parameter Sensitivity

Our belief propagation based connotation sentiment inference algorithm has one user-specified parameter ϵ (see Table 1). To study the sensitivity of its performance to the choice of ϵ, we reran our experiments for $\epsilon = \{0.02, 0.04, \ldots, 0.24\}$[6] and report the accuracy results on our $G^{\text{WORD+SENSE}}$ in Figure 2 for the two lexicons. The results indicate that the performances remain quite stable across a wide range of the parameter choice.



(a) GENINQ EVAL (b) MPQA EVAL

Figure 2: Performance is stable across various ϵ.

5 Evaluation II: Human Evaluation on ConnotationWordNet

In this section, we present the result of human evaluation we executed using Amazon Mechanical Turk (AMT). We collect two separate sets of labels: a set of labels at the word-level, and another set at the sense-level. We first describe the labeling process of sense-level connotation: We selected 350 polysemous words and one of their senses, and each Turker was asked to rate the connotative polarity of a given word (or of a given sense), from -5 to 5, 0 being the neutral.[7] For each word, we asked 5 Turkers to rate and we took the average of the 5 ratings as the connotative intensity score of the word. We labeled a word as *negative* if its intensity score is less than 0 and *positive* otherwise. For word-level labels we apply similar procedure as above.

[6]Note that for $\epsilon > 0.25$, compatibilities of ψ^{t_2} in Table 1 are reversed, hence the maximum of 0.24.

[7]Because senses in WordNet can be tricky to understand, care should be taken in designing the task so that the Turkers will focus only on the corresponding sense of a word. Therefore, we provided the part of speech tag, the WordNet gloss of the selected sense, and a few examples as given in WordNet. As an incentive, each Turker was rewarded $0.07 per hit which consists of 10 words to label.

Lexicon	Word-level	Sense-level
SentiWordNet	27.22	14.29
OpinionFinder	31.95	-
Feng2013	62.72	-
$G^{\text{WORD+SENSE}}(95\%)$	84.91	83.43
$G^{\text{WORD+SENSE}}(99\%)$	84.91	83.71
E-$G^{\text{WORD+SENSE}}(95\%)$	**86.98**	**86.29**
E-$G^{\text{WORD+SENSE}}(99\%)$	86.69	85.71

Table 3: Word-/Sense-level evaluation results

5.1 Word-Level Evaluation

We first evaluate the word-level assignment of connotation, as shown in Table 3. The agreement between the new lexicon and human judges varies between 84% and 86.98%. Sentiment lexicons such as SentiWordNet (Baccianella et al. (2010)) and OpinionFinder (Wilson et al. (2005a)) show low agreement rate with human, which is somewhat as expected: human judges in this study are labeling for subtle connotation, not for more explicit sentiment. OpinionFinder's low agreement rate was mainly due to the low hit rate of the words (successful look-up rate, 33.43%). Feng2013 is the lexicon presented in (Feng et al., 2013) and it showed a relatively higher 72.13% hit rate.

Note that belief propagation was run until 95% and 99% of the nodes were converged in their beliefs. In addition, the seed words with known connotation labels originally consist of 20 positive and 20 negative predicates. We also extended the seed set with the sentiment lexicon words and denote these runs with E- for 'Extended'.

5.2 Sense-Level Evaluation

We also examined the agreement rates on the sense-level. Since OpinionFinder and Feng2013 do not provide the polarity scores at the sense-level, we excluded them from this evaluation. Because sense-level polarity assignment is a harder (more subtle) task, the performance of all lexicons decreased to some degree in comparison to that of word-level evaluations.

5.3 Pair-wise Intensity Ranking

A notable goodness of our induction algorithm is that the outcome of the algorithm can be interpreted as an *intensity* of the corresponding connotation. But are these values meaningful? We answer this question in this section. We formulate a pair-wise ranking task as a binary decision task as follows: given a pair of words, we ask which one is more positive (or more negative) than the other. Since we collect human labels based on *scales*, we

Lexicon	Correct	Undecided
SentiWordNet	33.77	23.34
$G^{\text{WORD+SENSE}}$(95%)	74.83	0.58
$G^{\text{WORD+SENSE}}$(99%)	73.01	0.58
E-$G^{\text{WORD+SENSE}}$(95%)	73.84	1.16
E-$G^{\text{WORD+SENSE}}$(99%)	74.01	1.16

Table 4: Results of pair-wise intensity evaluation, for intensity difference threshold = 2.0

already have this information at hand. Because different human judges have different notion of scales however, subtle differences are more likely to be noisy. Therefore, we experiment with varying degrees of differences in their scales, as shown in Figure 3. Threshold values (ranging from 0.5 to 3.0) indicate the minimum differences in scales for any pair of words, for the pair to be included in the test set. As expected, we observe that the performance improves as we increase the threshold (as pairs get better separated). Within range [0.5, 1.5] (249 pairs examined), the accuracies are as high as 68.27%, which shows that even the subtle differences of the connotative intensities are relatively well reflected in the new lexicons.

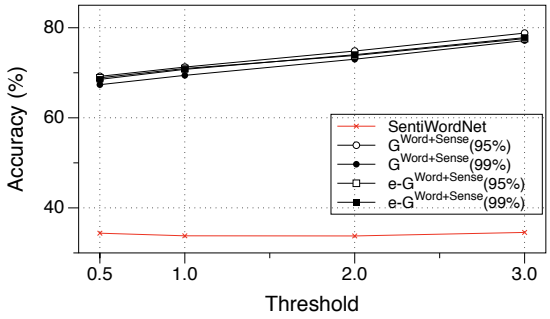


Figure 3: Trend of accuracy for pair-wise intensity evaluation over threshold

The results for pair-wise intensity evaluation (threshold=2.0, 1,208 pairs) are given in Table 4. Despite that intensity is generally a harder property to measure (than the coarser binary categorization of polarities), our connotation lexicons perform surprisingly well, reaching up to 74.83% accuracy. Further study on the incorrect cases reveals that SentiWordNet has many pair of words with the same polarity score (23.34%). Such cases seems to be due to the limited score patterns of SentiWordNet. The ratio of such cases are accounted as *Undecided* in Table 4.

6 Evaluation III: Sentiment Analysis using ConnotationWordNet

Finally, to show the utility of the resulting lexicon in the context of a concrete sentiment analysis task, we perform lexicon-based sentiment analysis. We experiment with SemEval dataset (Strapparava and Mihalcea, 2007) that includes the human labeled dataset for predicting whether a news headline is a *good news* or a *bad news*, which we expect to have a correlation with the use of *connotative* words that we focus on in this paper. The good/bad news are annotated with scores (ranging from -100 to 87). We construct several data sets by applying different thresholds on scores. For example, with the threshold set to 60, we discard the instances whose scores lie between -60 and 60. For comparison, we also test the connotation lexicon from (Feng et al., 2013) and the combined sentiment lexicon GENINQ+MPQA.

Note that there is a difference in how humans judge the orientation and the degree of connotation for a given word out of context, and how the use of such words in context can be perceived as *good/bad* news. In particular, we conjecture that humans may have a bias toward the use of positive words, which in turn requires calibration from the readers' minds (Pennebaker and Stone, 2003). That is, we might need to tone down the level of positiveness in order to correctly measure the actual intended positiveness of the message.

With this in mind, we tune the appropriate calibration from a small training data, by using 1 fold from N fold cross validation, and using the remaining $N-1$ folds as testing. We simply learn the mixture coefficient λ to scale the contribution of positive and negative connotation values. We tune this parameter λ[8] for other lexicons we compare against as well. Note that due to this parameter learning, we are able to report better performance for the connotation lexicon of (Feng et al., 2013) than what the authors have reported in their paper (labeled with *) in Table 5.

Table 5 shows the results for N=15, where the new lexicon consistently outperforms other competitive lexicons. In addition, Figure 4 shows that the performance does not change much based on the size of training data used for parameter tuning (N=$\{5, 10, 15, 20\}$).

7 Related Work

Several previous approaches explored the use of graph propagation for sentiment lexicon induction (Velikovich et al., 2010) and connotation lexicon

[8]What is reported is based on $\lambda \in \{20, 40, 60, 80\}$. More detailed parameter search does not change the results much.

Lexicon	SemEval Threshold			
	20	40	60	80
Instance Size	955	649	341	86
Feng2013	71.5	77.1	81.6	90.5
GENINQ+MPQA	72.8	77.2	80.4	86.7
$G^{\text{WORD+SENSE}}(95\%)$	74.5	**79.4**	86.5	**91.9**
$G^{\text{WORD+SENSE}}(99\%)$	**74.6**	**79.4**	**86.8**	**91.9**
E-$G^{\text{WORD+SENSE}}(95\%)$	72.5	76.8	82.3	87.2
E-$G^{\text{WORD+SENSE}}(99\%)$	72.6	76.9	82.5	87.2
Feng2013*	70.8	74.6	80.8	93.5
GENINQ+MPQA*	64.5	69.0	74.0	80.5

Table 5: SemEval evaluation results, for N=15

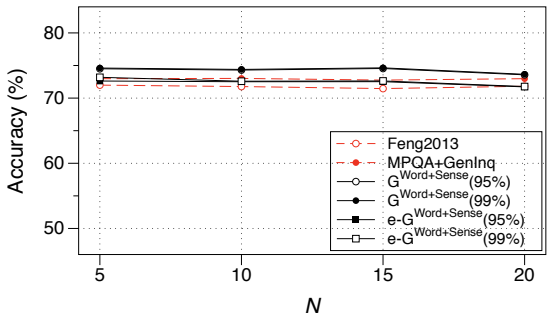


Figure 4: Trend of SemEval performance over N, the number of CV folds

induction (Feng et al., 2013). Our work introduces the use of loopy belief propagation over pairwise-MRF as an alternative solution to these tasks. At a high-level, both approaches share the general idea of propagating confidence or belief over the graph connectivity. The key difference, however, is that in our MRF representation, we can explicitly model various types of word-word, sense-sense and word-sense relations as edge potentials. In particular, we can naturally encode relations that encourage the same assignment (e.g., synonym) as well as the opposite assignment (e.g., antonym) of the polarity labels. Note that integration of the latter is not straightforward in the graph propagation framework.

There have been a number of previous studies that aim to construct a word-level sentiment lexicon (Wiebe et al., 2005; Qiu et al., 2009) and a sense-level sentiment lexicon (Esuli and Sebastiani, 2006). But none of these approaches considered to induce the polarity labels at both the word-level and sense-level. Although we focus on learning connotative polarity of words and senses in this paper, the same approach would be applicable to constructing a sentiment lexicon as well.

There have been recent studies that address word sense disambiguation issues for sentiment analysis. SentiWordNet (Esuli and Sebastiani, 2006) was the very first lexicon developed for sense-level labels of sentiment polarity. In recent years, Akkaya et al. (2009) report a successful empirical result where WSD helps improving sentiment analysis, while Wiebe and Mihalcea (2006) study the distinction between objectivity and subjectivity in each different sense of a word, and their empirical effects in the context of sentiment analysis. Our work shares the high-level spirit of accessing the sense-level polarity, while also deriving the word-level polarity.

In recent years, there has been a growing research interest in investigating more fine-grained aspects of lexical sentiment beyond positive and negative sentiment. For example, Mohammad and Turney (2010) study the affects words can *evoke* in people's minds, while Bollen et al. (2011) study various moods, e.g., "tension", "depression", beyond simple dichotomy of positive and negative sentiment. Our work, and some recent work by Feng et al. (2011) and Feng et al. (2013) share this spirit by targeting more subtle, nuanced sentiment even from those words that would be considered as objective in early studies of sentiment analysis.

8 Conclusion

We have introduced a novel formulation of lexicon induction operating over both words and senses, by exploiting the innate structure between the words and senses as encoded in WordNet. In addition, we introduce the use of loopy belief propagation over *pairwise*-Markov Random Fields as an effective lexicon induction algorithm. A notable strength of our approach is its expressiveness: various types of prior knowledge and lexical relations can be encoded as node potentials and edge potentials. In addition, it leads to a lexicon of better quality while also offering faster run-time and easiness of implementation. The resulting lexicon, called *ConnotationWordNet*, is the first lexicon that has polarity labels over both words and senses. *ConnotationWordNet* is publicly available for research and practical use.

Acknowledgments

This research was supported by the Army Research Office under Contract No. W911NF-14-1-0029, Stony Brook University Office of Vice President for Research, and gifts from Northrop Grumman Aerospace Systems and Google. We thank reviewers for many insightful comments and suggestions.

References

Cem Akkaya, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 190–199. Association for Computational Linguistics.

Leman Akoglu, Rishi Chandy, and Christos Faloutsos. 2013. Opinion fraud detection in online reviews by network effects.

Alina Andreevskaia and Sabine Bergler. 2006. Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *EACL*, pages 209–216.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Alexandra Balahur, Rada Mihalcea, and Andrés Montoyo. 2014. Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications. *Computer Speech & Language*, 28(1):1–6.

Johan Bollen, Huina Mao, and Alberto Pepe. 2011. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*.

K. W. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 1(16):22–29.

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.

Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *The Association for Computer Linguistics*, pages 1774–1784.

Vasileios Hatzivassiloglou and Kathleen McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the Joint ACL/EACL Conference*, pages 174–181.

Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In *EMNLP-CoNLL*, pages 1075–1083.

Jaap Kamps, MJ Marx, Robert J Mokken, and Maarten De Rijke. 2004. Using wordnet to measure semantic orientations of adjectives.

Ross Kindermann and J. L. Snell. 1980. *Markov Random Fields and Their Applications*.

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.

Mary McGlohon, Stephen Bay, Markus G. Anderle, David M. Steier, and Christos Faloutsos. 2009. Snare: a link analytic system for graph labeling and risk detection. In John F. Elder IV, Franoise Fogelman-Souli, Peter A. Flach, and Mohammed Zaki, editors, *KDD*, pages 1265–1274. ACM.

Rada Mihalcea, Carmen Banea, and Janyce Wiebe. 2012. Multilingual subjectivity and sentiment analysis. In *Tutorial Abstracts of ACL 2012*, pages 4–4. Association for Computational Linguistics.

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.

David Newman, Sarvnaz Karimi, and Lawrence Cavedon. 2009. External evaluation of topic models. In *Australasian Document Computing Symposium*, pages 11–18, Sydney, December.

Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. 2007. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *WWW*, pages 201–210.

Christos H Papadimitriou and Kenneth Steiglitz. 1998. *Combinatorial optimization: algorithms and complexity*. Courier Dover Publications.

James W Pennebaker and Lori D Stone. 2003. Words of wisdom: language use over the life span. *Journal of personality and social psychology*, 85(2):291.

John P Pestian, Pawel Matykiewicz, Michelle Linn-Gust, Brett South, Ozlem Uzuner, Jan Wiebe, K Bretonnel Cohen, John Hurdle, Christopher Brew, et al. 2012. Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*, 5(Suppl. 1):3.

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

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *IJCAI*, volume 9, pages 1199–1204.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine*, 29(3):93–106.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics.

Fangzhong Su and Katja Markert. 2009. Subjectivity recognition on word senses via semi-supervised mincuts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1–9. Association for Computational Linguistics.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 133–140. Association for Computational Linguistics.

Peter D. Turney. 2001. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-01)*, pages 491–502, Freiburg, Germany.

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 and Rada Mihalcea. 2006. Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1065–1072. 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. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005b. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Vancouver, CA.

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. 2003. Understanding belief propagation and its generalizations. In *Exploring AI in the new millennium*, pages 239–269.