+/-EffectWordNet: Sense-level Lexicon Acquisition for Opinion Inference

Yoonjung Choi and **Janyce Wiebe**
Department of Computer Science
University of Pittsburgh
yjchoi, wiebe@cs.pitt.edu

Abstract

Recently, work in NLP was initiated on a type of opinion inference that arises when opinions are expressed toward events which have positive or negative effects on entities (*+/-effect events*). This paper addresses methods for creating a lexicon of such events, to support such work on opinion inference. Due to significant sense ambiguity, our goal is to develop a sense-level rather than word-level lexicon. To maximize the effectiveness of different types of information, we combine a graph-based method using WordNet[1] relations and a standard classifier using gloss information. A hybrid between the two gives the best results. Further, we provide evidence that the model is an effective way to guide manual annotation to find +/-effect senses that are not in the seed set.

1 Introduction

Opinion mining (or sentiment analysis) identifies positive or negative opinions in many kinds of texts such as reviews, blogs, and news articles. It has been exploited in many application areas such as review mining, election analysis, and information extraction. While most previous research focusses on explicit opinion expressions, recent work addresses a type of opinion inference that arises when opinions are expressed toward events which have positive or negative effects on entities (Deng et al., 2013; Deng and Wiebe, 2014). We call such events *+/-effect* events.[2] Deng and Wiebe (2014) show how sentiments toward one entity may be propagated to other entities via opinion inference rules. They give the following example:

> (1) *The bill would curb skyrocketing health care costs.*

The writer expresses an explicit **negative** sentiment (by *skyrocketing*) toward the **object** (*health care costs*). The event, *curb*, has a **negative effect** on *costs*, since they are reduced. We can reason that the writer is **positive** toward the **event** because it has a negative effect on *costs*, toward which the writer is negative. From there, we can reason that the writer is **positive** toward *the bill*, since it is the agent of the positive event. Deng and Wiebe (2014) show that such inferences may be exploited to significantly improve explicit sentiment analysis systems.

However, to achieve its results, the system developed by Deng and Wiebe (2014) requires that all instances of +/-effect events in the corpus be manually provided as input. For the system to be fully automatic, it needs to be able to recognize +/-effect events automatically. This paper addresses methods for creating lexicons of such events, to support such work on opinion inference. We have discovered that there is significant sense ambiguity, meaning that words often have mixtures of senses among the classes *+effect*, *-effect*, and *Null*. Thus, we develop a sense-level rather than word-level lexicon.

One of our goals is to investigate whether the +/-effect property tends to be shared among semantically-related senses, and another is to use a method that applies to all word senses, not just to the senses of words in a given word-level lexicon. Thus, we build a graph-based model in which each node is a WordNet sense, and edges represent semantic WordNet relations between senses. In addition, we hypothesized that glosses also contain useful information. Thus, we develop

[1] WordNet 3.0, http://wordnet.princeton.edu/

[2] While the term *goodFor/badFor* is used in previous papers (Deng et al., 2013; Deng and Wiebe, 2014; Deng et al., 2014), we have since decided that +/-effect is a better term.

a supervised gloss classifier and define a hybrid model which gives the best overall performance. Finally, because all WordNet verb senses are incorporated into the model, we investigate the ability of the method to identify unlabeled senses that are likely to be +/-effect senses. We find that by iteratively labeling the top-weighted unlabeled senses and rerunning the model, it may be used as an effective method for guiding annotation efforts.

2 Background

There are many varieties of +/-effect events, including *creation/destruction* (changes in states involving existence), *gain/loss* (changes in states involving possession), and *benefit/injury* (Anand and Reschke, 2010; Deng et al., 2013). The *creation*, *gain*, and *benefit* classes are +effect events. For example, *baking a cake* has a positive effect on the cake because it is created;[3] *increasing the tax rate* has a positive effect on the tax rate; and *comforting the child* has a positive effect on the child. The antonymous classes of each are -effect events: *destroying the building* has a negative effect on the building; *demand decreasing* has a negative effect on demand; and *killing Bill* has a negative effect on Bill.[4]

While sentiment (Esuli and Sebastiani, 2006; Wilson et al., 2005; Su and Markert, 2009) and connotation lexicons (Feng et al., 2011; Kang et al., 2014) are related, sentiment, connotation, and +/-effects are not the same; a single event may have different sentiment and +/-effect polarities, for example. Consider the following example:

> **perpetrate**:
> S: (v) perpetrate, commit, pull (perform an act, usually with a negative connotation) "perpetrate a crime"; "pull a bank robbery"

This sense of *perpetuate* has a **negative** connotation, and is an objective term in `SentiWordNet`. However, it has a **positive effect** on the object, *a crime*, since performing a crime brings it into existence.

[3]Deng et al. (2013) point out that +/-effect objects are not equivalent to benefactive/malefactive semantic roles. An example they give is *She baked a cake for me*: *a cake* is the object of the +effect event *baked* as just noted, while *me* is the filler of its benefactive semantic role (Ziga and Kittil, 2010).

[4]Their annotation manual, which gives additional cases, is available with the annotated data at http://mpqa.cs.pitt.edu/.

As we mentioned, the +/-effect ambiguity cannot be avoided in a word-level lexicon. In the +/-effect corpus of Deng et al. (2013),[5] +/-effect events and their agents and objects are annotated at the word level. In that corpus, 1,411 +/-effect instances are annotated; 196 different +effect words and 286 different -effect words appear in these instances. Among them, 10 words appear in both +effect and -effect instances, accounting for 9.07% of all annotated instances. They show that +/-effect events (and the inferences that motivate this work) appear frequently in sentences with explicit sentiment. Further, **all** instances of +/-effect words that are **not** identified as +/-effect events are false hits from the perspective of a recognition system.

The following is an example of a word with senses of different classes:

> **purge**:
> S: (v) purge (oust politically) "Deng Xiao Ping was purged several times throughout his lifetime" **-effect**
> S: (v) purge (clear of a charge) **+effect**
> S: (v) purify, purge, sanctify (make pure or free from sin or guilt) "he left the monastery purified" **+effect**
> S: (v) purge (rid of impurities) "purge the water"; "purge your mind" **+effect**

This is part of the WordNet output for the word *purge*. In the first sense, the polarity is -effect since it has a negative effect on the object, *Deng Xizo Ping*. However, the other cases have positive effect on the object. Moreover, although a word may not have both +effect and -effect senses, it may have mixtures of ((+effect or -effect) and Null). A purely word-based approach is blind to these cases.

3 Related Work

Lexicons are widely used in sentiment analysis and opinion mining. Several works such as Hatzivassiloglou and McKeown (1997), Turney and Littman (2003), Kim and Hovy (2004), Strapparava and Valitutti (2004), and Peng and Park (2011) have tackled automatic lexicon expansion or acquistion. However, in most such work, the lexicons are word-level rather than sense-level.

[5]Called the *goodFor/badFor* corpus in that paper.

For the related (but different) tasks of developing subjectivity, sentiment and connotation lexicons, some do take a sense-level approach. Esuli and Sebastiani (2006) construct `SentiWordNet`. They assume that terms with the same polarity tend to have similar glosses. So, they first expand a manually selected seed set of senses using WordNet lexical relations such as *also-see* and *direct antonymy* and train two classifiers, one for positive and another for negative. As features, a vectorial representation of glosses is adopted. These classifiers were applied to all WordNet senses to measure positive, negative, and objective scores. In extending their work (Esuli and Sebastiani, 2007), the PageRank algorithm is applied to rank senses in terms of how strongly they are positive or negative. In the graph, each sense is one node, and two nodes are connected when they contain the same words in their WordNet glosses. Moreover, a random-walk step is adopted to refine the scores in their recent work (Baccianella et al., 2010). In contrast, our approach uses WordNet relations and graph propagation in addition to gloss classification.

Gyamfi et al. (2009) construct a classifier to label the subjectivity of word senses. The hierarchical structure and domain information in WordNet are exploited to define features in terms of similarity (using the LCS metric in Resnik (1995)) of target senses and a seed set of senses. Also, the similarity of glosses in WordNet is considered. Even though they investigated the hierarchical structure by LCS values, WordNet relations are not exploited directly.

Su and Markert (2009) adopt a semi-supervised mincut method to recognize the subjectivity of word senses. To construct a graph, each node corresponds to one WordNet sense and is connected to two classification nodes (one for subjectivity and another for objectivity) via a weighted edge that is assigned by a classifier. For this classifier, WordNet glosses, relations, and monosemous features are considered. Also, several WordNet relations (e.g., *antonymy*, *similiar-to*, *direct hypernym*, etc.) are used to connect two nodes. Although they make use of both WordNet glosses and relations, and gloss information is utilized for a classifier, this classifier is generated only for weighting edges between sense nodes and classification nodes, not for classifying all senses.

Kang et al. (2014) present a unified model that assigns connotation polarities to both words and senses. They formulate the induction process as collective inference over pairwise-Markov Random Fields, and apply loopy belief propagation for inference. Their approach relies on selectional preferences of *connotative predicates*; the polarity of a connotation predicate suggests the polarity of its arguments. We have not discovered an analogous type of predicate for the problem we address.

Goyal et al. (2010) generate a lexicon of patient polarity verbs (PPVs) that impart positive or negative states on their patients. They harvest PPVs from a Web corpus by co-occurance with Kind and Evil agents and by bootstrapping over conjunctions of verbs. Riloff et al. (2013) learn positive sentiment phrases and negative situation phrases from a corpus of tweets with hashtag "sarcasm". However, both of these methods are word-level rather than sense-level.

Ours is the first NLP research into developing a sense-level lexicon for events that have negative or positive effects on entities.

4 +/-Effect Word-Level Seed Lexicon and Sense Annotations

To create the corpus used in this work, we developed a word-level seed lexicon, and then manually annotated all the senses of the words in that lexicon.

FrameNet[6] is based on a theory of meaning called Frame Semantics. In FrameNet, a Lexical Unit (LU) is a pairing of a word with a meaning, i.e., it corresponds to a sense in WordNet. Each LU of a polysemous word belongs to a different semantic frame, which is a description of a type of event, relation, or entity and, where appropriate, its participants. For instance, in the **Creating** frame, the definition is that a **Cause** leads to the formation of a **Created_entity**. It has a positive effect on the object, **Created_entity**. This frame contains about 10 LUs such as *assemble*, *create*, *yield*, and so on. FrameNet consists of about 1,000 semantic frames and about 10,000 LUs.

FrameNet is a useful resource to select +/-effect words since each semantic frame covers multiple LUs. We believe that using FrameNet to find +/-effect words is easier than finding +/-effect words without any information since words may

[6]FrameNet, https://framenet.icsi.berkeley.edu/fndrupal/

be filtered by semantic frames. To select +/-effect words, an annotator (who is not a co-author) first identified promising frames as +/-effect and extracted all LUs from them. Then, he went through them and picked out the LUs which he judged to be +effect or -effect. In total, 736 +effect LUs and 601 -effect LUs were selected from 463 semantic frames.

While Deng et al. (2013) and Deng and Wiebe (2014) specifically focus on events affecting objects (i.e., themes), we do not want to limit the lexicon to only that case. Sometimes, events have positive or negative effects on agents or other entities as well. Thus, in this paper, we consider a sense to be +effect (-effect) if it has +effect (-effect) on an entity, which may be the agent, the theme, or some other entity.

In a previous paper (Choi et al., 2014), we conducted a study of the sense-level +/-effect property. For the evaluation, two annotators (who are co-authors of that paper) independently annotated senses of selected words, where some are from pure +effect (-effect) words (i.e., all senses of the words are classified into the same class) and some are from mixed words (i.e., the words have both +effect and -effect senses). In the agreement study, we calculated percent agreement and κ (Artstein and Poesio, 2008), and achieved 0.84 percent agreement and 0.75 κ value.

For a seed set and an evaluation set in this paper, we need annotated sense-level +/-effect data. Mappings between FrameNet and WordNet are not perfect. Thus, we opted to manually annotate the senses of the words in the word-level lexicon. We first extracted all words from 736 +effect LUs and 601 -effect LUs; this extracts 606 +effect words and 537 -effect words (the number of words is smaller than the number of LUs because one word can have more than one LU). Among them, 14 words (e.g., *crush, order,* etc.) are in both the +effect word set and the -effect word set. That is, these words have both +effect and -effect meanings. Recall that this annotator was focusing on frames, not on words - he did not look at all the senses of all the words. As we will see just below, when all the senses of all the words are annotated, a much higher percentage of the words have both +effect and -effect senses. We will also see that many of the senses are revealed to be Null, showing that +effect vs. Null and -effect vs. Null ambiguities are quite prevalent.

A different annotator (a co-author) then went through all senses of all the words from the previous step and manually annotated each sense as to whether it is +effect, -effect, or Null. Note that this annotator participated in an agreement study with positive results in Choi et al. (2014).

For the experiments in this paper, we divided this annotated data into two equal-sized sets. One is a fixed test set that is used to evaluate both the graph model and the gloss classifier. The other set is used as a seed set by the graph model, and as a training set by the gloss classifer. Table 1 shows the distribution of the data. In total, there are 258 +effect senses, 487 -effect senses, and 880 Null senses. To avoid too large a bias toward the Null class,[7] we randomly chose half (i.e., the Null set contains 440 senses). Half of each set is used as seed and training data, and the other half is used for evaluation.

	+effect	**-effect**	**Null**
# annotated data	258	487	880
# Seed/TrainSet	129	243	220
# TestSet	129	244	220

Table 1: Distribution of annotated data.

5 Graph-based Semi-Supervised Learning for WordNet Relations

WordNet (Miller et al., 1990) is organized by semantic relations such as *hypernymy*, *troponymy*, *grouping*, and so on. These semantic relations can be used to build a network. Since the most frequently encoded relation is the super-subordinate relation, most verb senses are arranged into hierarchies; verb senses towards the bottom of the graph express increasingly specific manner. Thus, by following this hierarchical information, we hypothesized that +/-effect polarity tends to propagate. We use a graph-based semi-supervised learning (GSSL) method to carry out the label propagation.

5.1 Graph Formulation

We formulate a graph for semi-supervised learning as follows. Let $G = \{X, E, W\}$ be the undirected graph in which X is the set of nodes, E is the set

[7] As mentioned in the introduction, we want our method to be able to identify unlabeled senses that are likely to be +/-effect senses (see Section 8); we resize the Null class to support this goal.

of edges, and W represents the edge weights (i.e., the weight of edge E_{ij} is W_{ij}). The weight matrix is a non-negative matrix.

Each data point in X = $\{x_1, ... ,x_n\}$ is one sense. The labeled data of X is represented as X_L = $\{x_1, ... ,x_l\}$ and the unlabeled data is represented as X_U = $\{x_{l+1}, ... ,x_n\}$). The labeled data X_L is associated with labels Y_L = $\{y_1, ... ,y_l\}$, where $y_i \in \{1, ..., c\}$ (c is the number of classes). As is typical in such settings, $l \ll n$: n is 13,767, i.e., the number of verb senses in WordNet. Seed/TrainSet in Table 1 is the labeled data.

To connect two nodes, WordNet relations are utilized. We first connect nodes by the hierarchical relations. Since *hypernym* relations represent more general senses and *troponym* relations represent more specific verb senses, we hypothesized that hypernyms or troponyms of a verb sense tends to have its same polarity. *Verb groups* relations that represent verb senses having a similar meaning are also promising. Even though verb-group coverage is not large, its relations are reliable since they are manually grouped. The *entailment* relation is defined as the verb Y is entailed by X if you must be doing Y by doing X. Since pairs connected by this relation are co-extensive, we can assume that both are the same type of event. The *synonym* relation is not used because it is already defined in senses (i.e., each node in the graph is a synset), and the *antonym* relation is also not applied since the weight matrix should be non-negative. The weight value of all edges is 1.0.

5.2 Label Propagation

Given a constructed graph, the label inference (or prediction) task is to propagate the seed labels to the unlabeled nodes. One of the classic GSSL label propagation methods is the local and global consistency (LGC) method suggested by Zhou et al. (2004). The LGC method is a graph transduction algorithm which is sufficiently smooth with respect to the intrinsic structure revealed by known labeled and unlabeled data. The cost function typically involves a tradeoff between the smoothness of the predicted labels over the entire graph and the accuracy of the predicted labels in fitting the given labeled nodes X_L. LGC fits in a univariate regularization framework, where the output matrix is treated as the only variable in optimization, and the optimal solutions can be easily obtained by solving a linear system. Thus, we adopt the LGC method in this paper. Although there are some robust GSSL methods for handling noisy labels, we do not need to handle noisy labels because our input is the annotated data.

Let F be a $n \times c$ matrix to save the output values of label propagation. So, we can label each instance x_i as a label $y_i = \text{argmax}_{j \le c} F_{ij}$ after the label propagation. The initial discrete label matrix Y, which is also $n \times c$, is defined as $Y_{ij} = 1$ if x_i is labeled as $y_i = j$ in Y_L, and $Y_{ij} = 0$ otherwise. The vertex degree matrix $D = \text{diag}([D_{11}, ..., Dnn])$ is defined by $D_{ii} = \sum_{j=1}^{n} W_{ij}$.

LGC defines the cost function Q which integrates two penalty components, global smoothness and local fitting (μ is the regularization parameter):

$$Q = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \| \frac{F_i}{\sqrt{D_{ii}}} - \frac{F_j}{\sqrt{D_{jj}}} \|^2 + \mu \sum_{i=1}^{n} \|F_i - Y_i\|^2$$

The first part of the cost function is the *smoothness constraint*: a good classifying function should not change too much between nearby points. That is, if x_i and x_j are connected with an edge, the difference between them should be small. The second is the *fitting constraint*: a good classifying function should not change too much from the initial label assignment. The final label prediction matrix F can be obtained by minimizing the cost function Q.

5.3 Experimental Results

Note that, in the rest of this paper, all tables except the last one give results on the same fixed test set (TestSet in Table 1).

We can apply the graph model in two ways.

- **UniGraph**: All three classes (+effect, -effect, and Null) are represented in one graph.
- **BiGraph**: Two separate graphs are first constructed and then combined. One graph is for classifying +effect and Other (i.e., -effect or Null). This graph is called *+eGraph*. The other graph, called *-eGraph*, is for classifying -effect and Other (i.e., +effect or Null).

		UniGraph	BiGraph	BiGraph*
baseline-accuracy		0.411		
accuracy		**0.630**	**0.623**	**0.658**
+effect	P	0.621	0.610	0.642
	R	0.655	0.647	0.680
	F	0.637	0.628	**0.660**
-effect	P	0.644	0.662	0.779
	R	0.720	0.677	0.612
	F	0.680	0.670	**0.686**
Null	P	0.615	0.583	0.583
	R	0.516	0.550	0.695
	F	0.561	0.561	**0.634**

Table 2: Results of UniGraph, BiGraph, and BiGraph*.

These are combined into one model as follows. Nodes that are labeled as +effect by +eGraph and Other by -eGraph are regarded as +effect, and nodes that are labeled as -effect by -eGraph and Other by +eGraph are regarded as -effect. If nodes are labeled as +effect by +eGraph and -effect by -eGraph, they are deemed to be Null. Nodes that are labeled Other by both graphs are also considered as Null.

We had two motivations for experimenting with the BiGraph model: (1) SVM, the supervised learning method used for gloss classification, tends to have better performance on binary classification tasks, and (2) the two graphs of the combined model can "negotiate" with each other via constraints.

In Table 2, we calculate precision (P), recall (R), and f-measure (F) for all three classes. The baseline shown in the top row is the accuracy of a majority class classifier. The first two columns of Table 2 show the results of UniGraph and BiGraph when they are built using the *hypernym*, *troponym*, and *verb group* relations. UniGraph outperforms BiGraph in this experiment.

To improve the results by performing something possible with BiGraph (but not UniGraph), constraints are added when determining the class. As we explained, the label of instance x_i is determined by F_i in the graph. When the label of x_i is decided to be j, we can say that its confidence value is F_{ij}. There are two constraints as follows.

		H+T	+V	+E
+effect	P	0.653	0.642	0.651
	R	0.660	0.680	0.683
	F	0.656	0.660	**0.667**
-effect	P	0.784	0.779	0.786
	R	0.547	0.612	0.604
	F	0.644	**0.686**	0.683
Null	P	0.557	0.583	0.564
	R	0.735	0.695	0.691
	F	**0.634**	**0.634**	0.621

Table 3: Effect of each relation

- If a sense is labeled as +effect (-effect), but the confidence value is less than a threshold, we count it as Null.
- If a sense is labeled as both +effect and -effect by BiGraph, we choose the label with the higher confidence value only if the higher one is larger than a threshold and the lower one is less than a threshold.

The thresholds are determined on Seed/TrainSet by running BiGraph several times with different thresholds, and choosing the one that gives the best performance **on Seed/TrainSet**. (The chosen value is 0.025 for +effect and 0.03 for -effect).

As can be seen in Table 2, BiGraph with constraints (called *BiGraph**) outperforms not only BiGraph without any constraints but also UniGraph. Especially, for BiGraph*, the recall of the Null class is considerably increased, showing that constraints not only help overall, but are particularly important for detecting Null cases.

Table 3 gives ablation results, showing the contribution of each WordNet relation in BiGraph*. With only hierarchical information (i.e., *hypernym* (H) and *troponym* (T) relations), it already shows good performance for all classes. However, they cannot cover some senses. Among the 13,767 verb senses in WordNet, 1,707 (12.4%) cannot be labeled because there are not sufficient hierarchical links to propagate polarity information. When adding the *verb group* (+V) relation, it shows improvement in both +effect and -effect. Especially, the recall for +effect and -effect is significantly increased. In addition, the coverage of the 13,767 verb senses increases to 95.1%. For *entailment* (+E), whereas adding it shows a slight improvement in +effect (and increases coverage by 1.1 percentage points), the

performance is decreased a little bit in the -effect and Null classes. Since the average f-measure for all classes is the highest with *hypernym* (H), *troponym* (T), and *verb group* (V) relations (not *entailment*), we only consider these three relations when constructing the graph.

6 Supervised Learning applied to WordNet Glosses

In WordNet, each sense contains a gloss consisting of a definition and optional example sentences. Since a gloss consists of several words and there are no direct links between glosses, we believe that a word vector representation is appropriate to utilize gloss information as in Esuli and Sebastiani (2006). For that, we adopt an SVM classifier.

6.1 Features

Two different feature types are used.

Word Features (WF): The bag-of-words model is applied. We do not ignore stop words for several reasons. Since most definitions and examples are not long, each gloss contains a small number of words. Also, among them, the total vocabulary of WordNet glosses is not large. Moreover, some prepositions such as *against* are sometimes useful to determine the polarity (+effect or -effect).

Sentiment Features (SF): Some glosses of +effect (-effect) senses contain positive (negative) words. For instance, the definition of {*hurt#4, injure#4*} is "cause damage or affect negatively." It contains a negative word, *negatively*. Since a given event may positively (negatively) affect entities, some definitions or examples already contain positive (negative) words to express this. Thus, as features, we check how many positive (negative) words a given gloss contains. To detect sentiment words, the subjectivity lexicon provided by Wilson et al. (2005)[8] is utilized.

6.2 Gloss Classifier

We have three classes, +effect, -effect, and Null. Since SVM shows better performance on binary classification tasks, we generate two binary classifiers, one (*+eClassifier*) to determine whether a given sense is +effect or Other, and another (*-eClassifier*) to classify whether a given sense is -effect or Other. Then, they are combined as in BiGraph.

[8] Available at http://mpqa.cs.pitt.edu/

6.3 Experimental Results

Seed/TrainSet in Table 1 is used to train the two classifiers, and TestSet is utilized for the evaluation. So, the training set for +eClassifier consists of 129 +effect instances and 463 Other instances, and the training set for -eClassifier contains 243 -effect instances and 349 Other instances. As a baseline, we adopt a majority class classifier.

Table 4 shows the results on TestSet. Performance is better for the -effect than for the +effect class, perhaps because the -effect class has more instances.

When sentiment features (SF) are added, all metric values increase, providing evidence that sentiment features are helpful to determine +/-effect classes.

		WF	WF+SF
baseline accuracy		0.411	
accuracy		**0.509**	**0.539**
+effect	P	0.541	0.588
	R	0.354	0.393
	F	0.428	**0.472**
-effect	P	0.616	0.672
	R	0.500	0.511
	F	0.552	**0.580**
Null	P	0.432	0.451
	R	0.612	0.657
	F	0.507	**0.535**

Table 4: Results of the gloss classifier.

7 Hybrid Method

To use more combined knowledge, the gloss classifier and BiGraph* can be combined. That is, for WordNet gloss information, the gloss classifier is utilized, and for WordNet relations, BiGraph* is used. With the Hybrid method, we can see not only the effect of propagation by WordNet relations but also the usefulness of gloss information and sentiment features. Also, while BiGraph* cannot cover all senses in WordNet, the Hybrid method can.

The outputs of the gloss classifier and BiGraph* are combined as follows. The label of the gloss classifier is one of +effect, -effect, Null, or Both (when a given sense is classified as both +effect by +eClassifier and -effect by -eClassifier). Possible labels of BiGraph* are +effect, -effect, Null, Both, or None (when a given sense is not

labeled by BiGraph*). There are five rules:

- If both labels are +effect (-effect), it is +effect (-effect).
- If one of them is Both and the other is +effect (-effect), it is +effect (-effect).
- If the label of BiGraph* is None, believe the label of the gloss classifier
- If both labels are Both, it is Null
- Otherwise, it is Null

The results for Hybrid are given in the first row of the lower half of Table 5; the results for BiGraph* are in the first row of the upper half, for comparison. Generally, the Hybrid method shows better performance than the gloss classifier and BiGraph*. In the Hybrid method, since more +/-effect senses are detected than by BiGraph*, while precision is decreased, recall is increased by more. However, by the same token, the overall performance for the Null class is decreased. Actually, that is expected since the Null class is determined by the Other class in the gloss classifier and BiGraph*. Through this experiment, we see that the Hybrid method is better for classifying +/-effect senses.

7.1 Model Comparison

To provide evidence for our assumption that different models are needed for different information to maximize effectiveness, we compare the hybrid method with the supervised learning and the graph-based learning (GSSL) methods, each utilizing both WordNet relations and gloss information.

Supervised Learning (*onlySL*): The gloss classifier is trained with word features and sentiment features for WordNet Gloss. To exploit WordNet relations in supervised learning, especially the hierarchical information, we use least common subsumer (LCS) values as in Gyamfi et al. (2009), which, recall, performs supervised learning of subjective/objective senses. The values are calculated as follows. For a target sense t and a seed set S, the maximum LCS value between a target sense and a member of the seed set is found as:

$$Score(t, S) = max_{s \in S} LCS(t, s)$$

With this LCS feature and the features described in Section 6, we run SVM on the same training and test data. For LCS values, the similarity using the information content proposed by Resnik (1995) is measured. WordNet Similarity[9] package provides pre-computed pairwise similarity values for that.

Table 6 shows results of onlySL. Compared to Table 4, while +effect and Null classes show a slight improvement, the performance is degraded for -effect. This means that the added feature is rather harmful to -effect. Even though the hierarchical feature is very helpful to expand +/-effect, it is not helpful for onlySL since SVM cannot capture propagation according to the hierarchy.

Graph-based Learning (*onlyGraph*): In Section 5, the graph is constructed by using WordNet relations. To apply WordNet gloss information in onlyGraph, we calculate a cosine similarity between glosses. If the similarity value is higher than a threshold, two nodes are connected with this similarity value. The threshold is determined by training and testing on Seed/TrainSet (the chosen value is 0.3).

Comparing Tables 2 and 6, BiGraph* generally outperforms onlyGraph (the exception is precision of +effect). By gloss similarity, many nodes are connected to each other. However, since uncertain connections can cause incorrect propagation in the graph, this negatively affects the performance.

Through this experiment, we see that since each type of information has a different character, we need different models to maximize the effectiveness of each type. Thus, the hybrid method with different models can have better performance.

		Hybrid	**onlySL**	**onlyGraph**
+effect	P	0.610	0.584	0.701
	R	0.735	0.400	0.364
	F	**0.667**	0.475	0.480
-effect	P	0.717	0.778	0.651
	R	0.669	0.316	0.562
	F	**0.692**	0.449	0.603
Null	P	0.556	0.440	0.473
	R	0.520	0.813	0.679
	F	0.538	**0.571**	0.557

Table 6: Comparison to onlySL and onlyGraph.

[9] WordNet Similarity, http://wn-similarity.sourceforge.net/

		+effect			-effect			Null		
		P	R	F	P	R	F	P	R	F
BiGraph*	**Initial**	0.642	0.680	**0.660**	0.779	0.612	**0.686**	0.583	0.695	**0.634**
	1st	0.636	0.684	0.663	0.770	0.632	0.694	0.591	0.672	0.629
	2nd	0.642	0.701	0.670	0.748	0.656	0.699	0.605	0.655	0.629
	3rd	0.636	0.708	0.670	0.779	0.652	0.710	0.599	0.669	0.632
	4th	0.681	0.674	**0.678**	0.756	0.674	**0.712**	0.589	0.669	0.626
Hybrid	**Initial**	0.610	0.735	**0.667**	0.717	0.669	**0.692**	0.556	0.520	**0.538**
	1st	0.614	0.713	0.672	0.728	0.681	0.704	0.562	0.523	0.542
	2nd	0.613	0.743	0.672	0.716	0.697	0.706	0.559	0.497	0.526
	3rd	0.616	0.739	0.672	0.717	0.706	0.712	0.559	0.494	0.525
	4th	0.688	0.681	**0.684**	0.712	0.764	**0.732**	0.565	0.527	0.545

Table 5: Results of an iterative approach.

8 Guided Annotation

Recall that Seed/TrainSet and TestSet, the data used so far, are all the senses of the words in a word-level +/-effect lexicon. This section presents evidence that our method can guide annotation efforts to find other words that have +/-effect senses. A bonus is that the method pinpoints particular +/-effect senses of those words.

All unlabeled data are senses of words that are not included in the original lexicon. Since presumably the majority of verbs do not have any +/-effect senses, a sense randomly selected from WordNet is very likely to be Null. We explore an iterative approach to guided annotation, using BiGraph* and Hybrid as the method for assigning labels.

The system is initially created as described above using Seed/TrainSet as the initial seed set. Each iteration has four steps: 1) rank all unlabeled data (i.e., the data other than TestSet and the current seed set) based on the F_{ij} confidence values (see Section 5.3); 2) choose the top 5% and manually annotate them (the same annotator as above did this); 3) add them to the seed set; 4) rerun the system using the expanded seed set. We performed four iterations in this paper.

The upper and lower parts of Table 5 show the intial results and the results after each iteration for BiGraph* and Hybrid. Recall that these are results on the fixed set, TestSet. Overall for both models, f-measure increases for both the +effect and -effect classes as more seeds are added, mainly due to improvements in recall. The evaluation on the fixed set is also useful in the annotation process because it trades off +/-effect vs. Null annotations. If the new manual annotations were biased, in that they incorrectly label Null senses as +/-effect, then the f-measure results would instead degrade on the fixed TestSet, since the system is created each time using the increased seed set.

We now consider the accuracy of the system on the newly labeled annotated data in Step 2. Note that our method is similar to Active Learning (Tong and Koller, 2001), in that both automatically identify which unlabeled instances the human should annotate next. However, in active learning, the goal is to find instances that are difficult for a supervised learning system. In our case, the goal is to find needles in the haystack of WordNet senses. In Step 3, we add the newly labeled senses to the seed set, enabling the model to find unlabeled senses close to the new seeds when the system is rerun for the next iteration.

We assess the system's accuracy on the newly labeled data by comparing the system's labels with the human's new labels. Accuracy for +effect and -effect is calculated such as:

$$Accuracy_{+effect} = \frac{\text{\# annotated +effect}}{\text{\# top 5\% +effect data}}$$

$$Accuracy_{-effect} = \frac{\text{\# annotated -effect}}{\text{\# top 5\% -effect data}}$$

That is, the accuracy means that out of the top 5% of the +effect (-effect) data as scored by the system, what percentage are correct as judged by a human annotator. Table 7 shows the accuracy for each iteration in the top part and the number of senses labeled in the bottom part. As can be seen, the accuracies range between 60% and 78%; these

values are much higher than what would be expected if labeling senses of words randomly chosen from WordNet.[10] The annotator spent, on average, approximately an hour to label 100 senses. For finding new words with +/-effect usages, it would be much more cost-effective if a significant percentage of the data chosen for annotation are senses of words that in fact have +/-effect senses.

	1st	2nd	3rd	4th
+effect	65.63%	62.50%	63.79%	59.83%
-effect	73.55%	73.97%	77.78%	70.30%
+effect	128	122	116	117
-effect	155	146	153	145
total	283	268	269	262

Table 7: Accuracy and frequency of the top 5% for each iteration

9 Conclusion and Future Work

In this paper, we investigated methods for creating a sense-level +/-effect lexicon. To maximize the effectiveness of each type of information, we combined a graph-based method using WordNet relations and a standard classifier using gloss information. A hybrid between the two gives the best results. Further, we provide evidence that the model is an effective way to guide manual annotation to find +/-effect words that are not in the seed word-level lexicon. This is important, as the likelihood that a random WordNet sense (and thus word) is +effect or -effect is not large.

So as not to limit the inferences that may be drawn, our annotations include events that are +effect or -effect either the agent or object. In future work, we plan to exploit corpus-based methods using patterns as in Goyal et al. (2010) combined with semantic role labeling to refine the lexicon to distinguish which is the affected entity. Further, to actually exploit the acquired lexicon to process corpus data, an appropriate coarse-grained sense disambiguation process must be added, as Akkaya et al. (2009) and Akkaya et al. (2011) did for subjective/objective classification.

We hope the general methodology will be effective for other semantic properties. In opinion mining and sentiment analysis this is particularly needed, because different meanings of *positive* and *negative* are appropriate for different applications. This is a way to create lexicons that are customized with respect to one's own definitions.

It would be promising to combine our method with other methods to enable it to find +effect and -effect senses that are outside the coverage of WordNet. However, a WordNet-based lexicon gives a substantial base to build from.

[10]For reference, in 5th iteration, the +effect accuracy is 60.18% and the -effect accuracy is 69.93%, and in 6th iteration, the +effect accuracy is 59.81% and the -effect accuracy is 69.12%.

Acknowledgments

This work was supported in part by DARPA-BAA-12-47 DEFT grant #12475008 and National Science Foundation grant #IIS-0916046. We would like to thank the reviewers for their helpful suggestions and comments.

References

Cem Akkaya, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity word sense disambiguation. In *Proceedings of EMNLP 2009*, pages 190–199.

Cem Akkaya, Janyce Wiebe, Alexander Conrad, and Rada Mihalcea. 2011. Improving the impact of subjectivity word sense disambiguation on contextual opinion analysis. In *Proceedings of CoNLL 2011*, pages 87–96.

Pranna Anand and Kevin Reschke. 2010. Verb classes as evaluativity functor classes. In *Interdisciplinary Workshop on Verbs. The Identification and Representation of Verb Features*.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Comput. Linguist.*, 34(4):555–596.

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

Yoonjung Choi, Lingjia Deng, and Janyce Wiebe. 2014. Lexical acquisition for opinion inference: A sense-level lexicon of benefactive and malefactive events. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, pages 107–112. Association for Computational Linguistics.

Lingjia Deng and Janyce Wiebe. 2014. Sentiment propagation via implicature constraints. In *Proceedings of EACL*.

Lingjia Deng, Yoonjung Choi, and Janyce Wiebe. 2013. Benefactive/malefactive event and writer attitude annotation. In *Proceedings of 51st ACL*, pages 120–125.

Lingjia Deng, Janyce Wiebe, and Yoonjung Choi. 2014. Joint inference and disambiguation of implicit sentiments via implicature constraints. In *Proceedings of COLING*, page 7988.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of 5th LREC*, pages 417–422.

Andrea Esuli and Fabrizio Sebastiani. 2007. Pageranking wordnet synsets: An application to opinion mining. In *Proceedings of ACL*, pages 424–431.

Song Feng, Ritwik Bose, and Yejin Choi. 2011. Learning general connotation of words using graph-based algorithms. In *Proceedings of EMNLP*, pages 1092–1103.

Amit Goyal, Ellen Riloff, and Hal DaumeIII. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of EMNLP*, pages 77–86.

Yaw Gyamfi, Janyce Wiebe, Rada Mihalcea, and Cem Akkaya. 2009. Integrating knowledge for subjectivity sense labeling. In *Proceedings of NAACL HLT 2009*, pages 10–18.

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

Jun Seok Kang, Song Feng, Leman Akoglu, and Yejin Choi. 2014. Connotationwordnet: Learning connotation over the word+sense network. In *Proceedings of the 52nd ACL*, page 15441554.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of 20th COLING*, pages 1367–1373.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 13(4):235–312.

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

Philip Resnik. 1995. Using information content to evaluate semantic similarity. In *Proceedings of 14th IJCAI*, pages 448–453.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of EMNLP*, pages 704–714.

Carlo Strapparava and Alessandro Valitutti. 2004. Wordnet-affect: An affective extension of wordnet. In *Proceedings of 4th LREC*, pages 1083–1086.

Fangzhong Su and Katja Markert. 2009. Subjectivity recognition on word senses via semi-supervised mincuts. In *Proceedings of NAACL HLT 2009*, pages 1–9.

Simon Tong and Daphne Koller. 2001. Support vector machin active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66.

Peter Turney and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.

Theresa Wilson, Janyce Wiebe, , and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT-EMNLP*, pages 347–354.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Scholkopf. 2004. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16:321–329.

Fernando Ziga and Seppo Kittil. 2010. *Benefactives and malefactives, Typological perspectives and case studies*. John Benjamins Publishing.