Sentiment Propagation via Implicature Constraints

Lingjia Deng
Intelligent Systems Program
University of Pittsburgh
lid29@pitt.edu

Janyce Wiebe
Department of Computer Science
University of Pittsburgh
wiebe@cs.pitt.edu

Abstract

Opinions may be expressed implicitly via inference over explicit sentiments and events that positively/negatively affect entities (*goodFor/badFor* events). We investigate how such inferences may be exploited to improve sentiment analysis, given goodFor/badFor event information. We apply Loopy Belief Propagation to propagate sentiments among entities. The graph-based model improves over explicit sentiment classification by 10 points in precision and, in an evaluation of the model itself, we find it has an 89% chance of propagating sentiments correctly.

1 Introduction

Previous research in sentiment analysis and opinion extraction has largely focused on the interpretation of explicitly stated opinions. However, many opinions are expressed implicitly via *opinion implicature* (i.e., opinion-oriented defeasible inference). Consider the following sentence:

Ex(**1**) The bill would lower health care costs, which would be a tremendous positive change across the entire health-care system.

The writer is clearly positive toward the idea of *lowering health care costs*. But how does s/he feel about the costs? If s/he is **positive** toward the idea of *lowering* them, then, presumably, she is **negative** toward the *costs* themselves (specifically, how high they are). The only explicit sentiment expression, *tremendous positive change*, is positive, yet we can infer a **negative** attitude toward the **object** of the event itself (i.e., *health care costs*).

Going further, since *the bill* is the **agent** of an event toward which the writer is positive, we may (defeasibly) infer that the writer is **positive** toward *the bill*, even though there are no explicit sentiment expressions describing it.

Now, consider *The bill would curb skyrocketing health care costs.* The writer expresses an explicit **negative** sentiment (*skyrocketing*) toward the **object** (*health care costs*) of the event. Note that *curbing* costs, like *lowering* them, is bad for them (the costs are reduced). We can reason that, because the event is bad for something toward which the writer is negative, the writer is **positive** toward the **event**. We can reason from there, as above, that the writer is **positive** toward *the bill*, since it is the **agent** of the positive event.

These examples illustrate how explicit sentiments toward one entity may be propagated to other entities via opinion implicature rules. The rules involve events that positively or negatively affect entities. We call such events *goodFor/badFor* (hereafter *gfbf*)events.

This work investigates how gfbf event interactions among entities, combined with opinion inferences, may be exploited to improve classification of the writer's sentiments toward entities mentioned in the text. We introduce four rule schemas which reveal sentiment constraints among gfbf events and their agents and objects. Those constraints are incorporated into a graph-based model, where a node represents an entity (agent/object), and an edge exists between two nodes if the two entities participate in one or more gfbf events with each other. Scores on the nodes represent the explicit sentiments, if any, expressed by the writer toward the entities. Scores on the edges are based on constraints derived from the rules. *Loopy Belief Propagation (LBP)* (Pearl, 1982) is applied to accomplish sentiment propagation in the graph.

Two evaluations are performed. The first shows that the graph-based model improves over an explicit sentiment classification system. The second evaluates the graph-based model itself (and hence the implicature rules), assessing its ability to correctly propagate sentiments to nodes whose polarities are unknown. We find it has an 89% chance of propagating sentiment values correctly.

This is the first paper to address this type of sentiment propagation to improve sentiment analysis. To eliminate interference introduced by other components, we use manually annotated gfbf information to build the graph. Thus, the evaluations in this paper are able to demonstrate the promise of the overall framework itself.

2 Related Work

Much work in sentiment analysis has been on document-level classification. Since different sentiments may be expressed toward different entities in a document, fine-grained analysis may be more informative for applications.

However, fine-grained sentiment analysis remains a challenging task for NLP systems. For fully-automatic systems evaluated on the MPQA corpus (Wiebe et al., 2005), for example, a recent paper (Johansson and Moschitti, 2013) reports results that improve over previous work, yet the F-measures are in the 40s and 50s.

Most work in NLP addresses explicit sentiment, but some address implicit sentiment. For example, (Zhang and Liu, 2011) identify noun product features that imply opinions, and (Feng et al., 2013) identify objective words that have positive or negative connotations. However, identifying terms that imply opinions is a different task than sentiment propagation between entities. (Dasigi et al., 2012) search for implicit attitudes shared between authors, while we address inferences within a single text.

Several papers apply compositional semantics to determine polarity (e.g., (Moilanen and Pulman, 2007; Choi and Cardie, 2008; Moilanen et al., 2010); see (Liu, 2012) for an overview). The goal of such work is to determine one overall polarity of an expression or sentence. In contrast, our framework commits to a holder having sentiments toward various events and entities in the sentence, possibly of different polarities.

The idea of gfbf events in sentiment analysis is not entirely new. For example, two papers mentioned above (Zhang and Liu, 2011; Choi and Cardie, 2008) include linguistic patterns for the tasks that they address that include gfbf events, but they don't define general implicature rules relating sentiments and gfbf events, agents, and objects as we do. Recently, in linguistics, Anand and Reschke (2010; 2011) identify classes of gfbf terms, and carry out studies involving artificially constructed gfbf triples and corpus examples matching fixed linguistic templates. Our work focuses on gfbf triples in naturally-occurring data and uses generalized implicature rules. Goyal et al. (2012) generate a lexicon of *patient polarity verbs*, which correspond to gfbf events whose spans are verbs. Riloff et al. (2013) investigate sarcasm where the writer holds a positive sentiment toward a negative situation. However, neither of these works performs sentiment inference.

Graph-based models have been used for various tasks in sentiment analysis. Some work (Wang et al., 2011; Tan et al., 2011) apply LBP on a graph capturing the relations between users and tweets in Twitter data . However, they assume the nodes and the neighbors of nodes share the same sentiments. In contrast, we don't assume that neighbors share the same sentiment, and the task we address is different.

3 Opinion Implicatures

This section describes the opinion-implicature framework motivating the design of the graph-based method for sentiment analysis proposed below. The components of the framework are gfbf events, explicit sentiments, and rules operating over gfbf events and sentiments.

The definition of a gfbf event is from (Deng et al., 2013). A GOODFOR event is an event that positively affects an entity (similarly, for BADFOR events). (Deng et al., 2013) point out that gfbf 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 GOODFOR event *baked* (creating something is good for it (Anand and Reschke, 2010)), while *me* is the filler of its benefactive semantic role (Zúñiga and Kittilä, 2010).

Four implicature rule schemas are relevant for this paper.[1] Four individual rules are covered by

[1]*Implicatures* "normally accompany the utterances of a given sentence unless special factors exclude that possibility

each schema. *sent(α)* = β means that the **writer's** sentiment toward α is β, where α is a GOODFOR event, a BADFOR event, or the agent or object of a gfbf event, and β is either *positive* or *negative* (*pos* or *neg*, for short). P $\rightarrow$ Q is to infer Q from P.

Rule1: sent(gfbf event) $\rightarrow$ sent(object)
1.1 sent(GOODFOR) = pos $\rightarrow$ sent(object) = pos
1.2 sent(GOODFOR) = neg $\rightarrow$ sent(object) = neg
1.3 sent(BADFOR) = pos $\rightarrow$ sent(object) = neg
1.4 sent(BADFOR) = neg $\rightarrow$ sent(object) = pos

Rule2: sent(object) $\rightarrow$ sent(gfbf event)
2.1 sent(object) = pos $\rightarrow$ sent(GOODFOR) = pos
2.2 sent(object) = neg $\rightarrow$ sent(GOODFOR) = neg
2.3 sent(object) = pos $\rightarrow$ sent(BADFOR) = neg
2.4 sent(object) = neg $\rightarrow$ sent(BADFOR) = pos

Rule3: sent(gfbf event) $\rightarrow$ sent(agent)
3.1 sent(GOODFOR) = pos $\rightarrow$ sent(agent) = pos
3.2 sent(GOODFOR) = neg $\rightarrow$ sent(agent) = neg
3.3 sent(BADFOR) = pos $\rightarrow$ sent(agent) = pos
3.4 sent(BADFOR) = neg $\rightarrow$ sent(agent) = neg

Rule4: sent(agent) $\rightarrow$ sent(gfbf event)
4.1 sent(agent) = pos $\rightarrow$ sent(GOODFOR) = pos
4.2 sent(agent) = neg $\rightarrow$ sent(GOODFOR) = neg
4.3 sent(agent) = pos $\rightarrow$ sent(BADFOR) = pos
4.4 sent(agent) = neg $\rightarrow$ sent(BADFOR) = neg

To explain the rules, we step through an example:

EX**(2)** Why would [President Obama] **support** [health care reform]? Because [reform] could **lower** [*skyrocketing* health care costs], and **prohibit** [private insurance companies] from **overcharging** [patients].

Suppose a sentiment analysis system recognizes only one explicit sentiment expression, *skyrocketing*. According to the annotations, there are several gfbf events. Each is listed below in the form $\langle$agent, gfbf, object$\rangle$.

E_1: $\langle$reform, lower, costs$\rangle$
E_2: $\langle$reform, prohibit, E_3 $\rangle$
E_3: $\langle$companies, overcharge, patients$\rangle$
E_4: $\langle$Obama, support, reform$\rangle$

In E_1, from the negative sentiment expressed by *skyrocketing* (the writer is negative toward the costs because they are too high), and the fact that *costs* is the object of a BADFOR event (*lower*), Rule2.4 infers **a positive attitude toward E_1** .

Now, Rule3.3 applies. We infer the writer is **positive toward the *reform***, since it is the agent of E_1, toward which the writer is positive.

E_2 illustrates the case where the object is an event. Specifically, the object of E_2 is E_3, a BADFOR event (*overcharging*). As we can see, E_2 keeps E_3 from happening. Events such as E_2 are REVERSERs, because they reverse the polarity of a gfbf event (from BADFOR to GOODFOR, or vice versa). Note that REVERSERs may be seen as BADFOR events, because they make their objects irrealis (i.e., not happen). Similarly, a RETAINER such as *help* in *"help Mary save Bill"* can be viewed as a GOODFOR event. (We call a REVERSER or a RETAINER an INFLUENCER.) In this paper, RETAINERS are treated as GOODFOR events and REVERSERS are treated as BADFOR events.

Above, we inferred that the writer is positive toward *reform*, the agent of E_2. By Rule 4.3, the writer is **positive toward E_2**; then by Rule 1.3, the writer is **negative toward E_3**, the object of E_2.

For E_3, using Rule 1.4 we know the writer is **positive toward *patients*** and using Rule 3.4 we know the writer is **negative toward *companies***.

Turning to E_4, *support health care reform* is GOODFOR *reform*. We already inferred the writer is positive toward *reform*. Rule 2.1 infers that the writer is **positive toward E_4**. Rule 3.1 then infers that the writer is **positive toward the agent of E_4, *Obama***.

In summary, we infer that the writer is positive toward E_1, health care reform, E_2, patients, E_4, and Obama, and negative toward E_3 and private insurance companies.

4 Data

We use the data described in (Deng et al., 2013),[2] which consists of 134 documents about a controversial topic, "the Affordable Care Act." The documents are editorials and blogs, and are full of opinions.

In the data, gfbf triples are annotated specifying the spans of the gfbf event, its agent, and its object, as well as the polarity of the gfbf event (GOODFOR or BADFOR), and the writer's attitude toward the agent and object (positive, negative, or neutral). Influencers are also annotated. The agents of gfbf

(p. 39)." (Huddleston and Pullum, 2002)

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

and influencer events are noun phrases. The object of a gfbf event is a noun phrase, but the object of an influencer is a gfbf event or another influencer. A *triple chain* is a chain of zero or more influencers ending in a gfbf event, where the object of each element of the chain is the following element in the chain. (e.g. in EX(2), the two event *prohibit* and *overcharging* is a triple chain.)

In total, there are 1,762 annotated gfbf triples, out of which 692 are GOODFOR or RETAINER and 1,070 are BADFOR or REVERSER. From the writer's perspective, 1,495 noun phrases are annotated positive, 1,114 are negative and the remaining 8 are neutral. This is not surprising, given that most of the sentences in the data are opinionated.

5 Graph-based Model

We propose a graph-based model of entities and the gfbf relations between them to enable sentiment propagation between entities. In this section, we introduce the definition of the graph (in 5.1), the LBP algorithm (in 5.2), and the definition of its functions for our task (in 5.3 and 5.4).

5.1 Definition of the Entity Graph

We define a gfbf entity graph $EG = \{N, E\}$, in which the node set N consists of nodes, each representing an annotated noun phrase agent or object span. The edge set E consists of edges, each linking two nodes if they co-occur in a triple chain with each other. Consider the triples of EX(2) in Section 3 below.

E_1: $\langle$reform, lower, costs$\rangle$
E_2: $\langle$reform, prohibit, E_3 $\rangle$
E_3: $\langle$companies, overcharge, patients$\rangle$
E_4: $\langle$Obama, support, reform$\rangle$

The node of *reform* is linked to nodes of *costs* via E_1 and *Obama* via E_4.[3] Note that, for E_2 and E_3, the two are linked in a chain: $\langle$*reform, prohibit,* $\langle$*companies, overcharge, patients*$\rangle$ $\rangle$. The three nodes *reform*, *companies* and *patients* participate in this triple chain; thus, pairwise edges exist among them. The edge linking *companies* and *patients* is BADFOR (because of *overcharging*). The edge linking *reform* and *companies* is also a BADFOR since we treat a REVERSER as BADFOR. The edge linking *reform* and *patients* encodes two BADFOR events (*prohibit-overcharge*); computationally we say two BADFORs result in a GOODFOR, so the edge linking the two is GOODFOR.[4]

Given a text, we get the spans of gfbf events and their agents and objects plus the polarities of the events (GOODFOR/BADFOR) from the manual annotations, and then build the graph upon them. However, the manual annotations of the writer's sentiments toward the agents and objects are used as the gold standard for evaluation.

5.2 Sentiment Inference via LBP

```
**initialize** all $m_{i\to j}(pos) = m_{i\to j}(neg) = 1$
**repeat**
    **foreach** $n_i \in N$ **do**
        **foreach** $n_j \in Neighbor(n_i)$ **do**
            **foreach** $y \in pos, neg$ **do**
                calculate $m_{i\to j}(y)$
            normalize $m_{i\to j}(pos) + m_{i\to j}(neg) = 1$
**until** all $m_{i\to j}$ stop changing;
**for each** $n_i \in N$ **assign its polarity as**
    $\underset{y \in pos,neg}{\text{argmax}}\ \Phi_i(y) * \prod_{n_k \in Neighbor(n_i)} m_{k\to i}(y)$
    *neutral*, in case of a tie
```

Table 1: Loopy Belief Propagation

With graph EG containing cycles and no apparent structure, we utilize an approximate collective classification algorithm, *loopy belief propagation (LBP)* (Pearl, 1982; Yedidia et al., 2005), to classify nodes through belief message passing. The algorithm is shown in Table 1.

In LBP, each node has a score, $\Phi_i(y)$, and each edge has a score, $\Psi_{ij}(y_i, y_j)$. In our case, $\Phi_i(y)$ represents the writer's explicit sentiment toward n_i. $\Psi_{ij}(y_i, y_j)$ is the score on edge e_{ij}, representing the likelihood that node n_i has polarity y_i and n_j has polarity y_j. The specific definitions of the two functions are given in Sections 5.3 and 5.4.

LBP is an iterative message passing algorithm. A message from n_i to n_j over edge e_{ij} has two values: $m_{i\to j}(pos)$ is how much information from node n_i indicates node n_j is positive, and $m_{i\to j}(neg)$ is how much information from node n_i indicates node n_j is negative. In each iteration, the two are normalized such that $m_{i\to j}(pos) + m_{i\to j}(neg) = 1$. The message from n_i to its

[3]This assumes that the two instances of "reform" co-refer. However, the system does not resolve co-reference – the methods that we tried did not improve overall performance.

[4]Also, GOODFOR+BADFOR=BADFOR; GOODFOR+GOODFOR=GOODFOR

neighbor n_j is computed as:

$$m_{i\to j}(pos) = \Psi_{ij}(pos,pos) * \Phi_i(pos) * \prod_{n_k \in Neighbor(n_i)/n_j} m_{k\to i}(pos) + \Psi_{ij}(neg,pos) * \Phi_i(neg) * \prod_{n_k \in Neighbor(n_i)/n_j} m_{k\to i}(neg) \tag{1}$$

$$m_{i\to j}(neg) = \Psi_{ij}(neg,neg) * \Phi_i(neg) * \prod_{n_k \in Neighbor(n_i)/n_j} m_{k\to i}(neg) + \Psi_{ij}(pos,neg) * \Phi_i(pos) * \prod_{n_k \in Neighbor(n_i)/n_j} m_{k\to i}(pos) \tag{2}$$

For example, the first part of Equation (1) means that the positive message n_i conveys to n_j (i.e., $m_{i\to j}(pos)$) comes from n_i being positive itself ($\Phi_i(pos)$), the likelihood of edge e_{ij} with its nodes n_i being positive and n_j being positive ($\Psi_{ij}(pos,pos)$), and the positive message n_i's neighbors (besides n_j) convey to it ($\prod_{k\in Neighbor(n_i)/n_j} m_{k\to i}(pos)$).

After convergence, the polarity of each node is determined by its explicit sentiment and the messages its neighbors convey to it, as shown at the end of the algorithm in Table 1.

By this method, we take into account both sentiments and the interactions between entities via gfbf events in order to discover implicit attitudes.

Note that the node and edge scores are determined initially and do not change. Only $m_{i\to j}$ changes from iteration to iteration.

5.3 $\Psi_{ij}(y_i, y_j)$: GFBF Implicature Relations

The score $\Psi_{i,j}$ encodes constraints based on the gfbf relationships that nodes n_i and n_j participate in, together with the implicature rules given above.

Rule schemas 1 and 3 infer sentiments toward entities (agent/object) from sentiments toward gfbf events. All cases covered by them are shown in Table 2 (use s(α) to represent sent(α)).

			Rule 3	Rule1
s(gfbf)	gfbf type	$\rightarrow$	s(agent)	s(object)
pos	GOODFOR	$\rightarrow$	pos	pos
neg	GOODFOR	$\rightarrow$	neg	neg
pos	BADFOR	$\rightarrow$	pos	neg
neg	BADFOR	$\rightarrow$	neg	pos

Table 2: Rule 1 & Rule 3

A table of Rule schemas 2 and 4 would be exactly the same, except that the inference ($\rightarrow$) would be in the opposite direction ($\leftarrow$).

From Table 2, we see that, regardless of the writer's sentiment toward the event, if the event is GOODFOR, then the writer's sentiment toward the agent and object are the same, while if the event is BADFOR, the writer's sentiment toward the agent and object are opposite. Thus, the event type and the writer's sentiments toward the agents and objects give us constraints. Therefore, we define $\Psi_{ij}(pos,pos)$ and $\Psi_{ij}(neg,neg)$ to be 1 if the two nodes are linked by a GOODFOR edge; otherwise, it is 0; and we define $\Psi_{ij}(neg,pos)$ and $\Psi_{ij}(pos,neg)$ to be 1 if the two nodes are linked by a BADFOR edge; otherwise, it is 0.

5.4 $\Phi_i(y)$: Explicit Sentiment Classifier

The score of a node, $\Phi_i(y)$, represents the sentiment explicitly expressed by the writer toward that entity in the document. Since y ranges over (pos, neg), each node has a positive and a negative score; the scores sum to 1. If it is a positive node, then its positive value ranges from 0.5 to 1, and its negative value ranges from 0 to 0.5 (similarly for negative nodes). For any node without explicit sentiment, both the positive and negative values are 0.5, indicating a neutral node.

Thus, we build a sentiment classifier that takes a node as input and outputs a positive and a negative score. It is built from widely-used, freely available resources: the OpinionFinder (Wilson et al., 2005) and General Inquirer (Stone et al., 1966) lexicons and the OpinionFinder system.[5] We also use a new Opinion Extraction system (Johansson and Moschitti, 2013) that shows better performance than previous work on fine-grained sentiment analysis,[6] and a new automatically developed connotation lexicon (Feng et al., 2013).[7]

We implement a weighted voting method among these various sentiment resources. After that, for nodes that have not yet been assigned polar values (positive or negative), we implement a simple local discourse heuristic to try to assign them polar values.

The particular strategies were chosen based only on a separate development set, which is not

[5]http://mpqa.cs.pitt.edu and http://www.wjh.harvard.edu/ inquirer/

[6]As evaluated on the MPQA corpus. Note that the authors ran their system for us on the data we use.

[7]http://www.cs.stonybrook.edu/~ychoi/connotation

included in the data used in the experiments.

5.4.1 Explicit Sentiment Tools

Opinion Extraction outputs a polarity expression with its source, and OpinionFinder outputs a polarity word. But neither of the tools extracts the target. To extract the target, for each word in the opinion expression, we select other words in the sentence which are in a *mod*, *obj* dependency parsing relation with it.

We match up the extracted expressions and the gfbf annotations according to their offsets in the text. For an opinion expression appearing in the sentence with no gfbf annotation, if the root word (in the dependency parse) of the expression span is the same as the root word of a gfbf span, or the root word of an agent span, or the root word of an object span, we assume they match up. Then we assign polarity as follows. If the expression refers only to the agent or object, then the agent or object is assigned the polarity of the expression. If the expression covers the gfbf event and its object, we assume the sentiment is toward the gfbf event and then assign sentiment according to Rule schema 1 (sent(gfbf event) $\rightarrow$ sent(object)).

5.4.2 Lexicons

To classify the sentiment expressed within the span of an agent or object, we check whether the words in the span appear in one or more of the lexicons.[8] If a lexicon finds both positive and negative words in the span, we resolve the conflict by choosing the polarity of the root word in the span. If the root word does not have a polar value, we choose the majority polarity of the sentiment words. If there are an equal number of positive and negative words, the polarity is neutral.

5.4.3 Voting Scheme among Resources

All together we have two sentiment systems and three lexicons. Before explicit sentiment classifying, each node has a positive value of 0.5 and a negative value of 0.5. We give the five votes equal weight (0.1), and add the number of positive votes multiplied by 0.1 to the positive value, and the number of negative votes multiplied by 0.1 to the negative value. After this addition, both values are in the range 0.5 to 1. If the positive value is larger, we maintain the positive value and assign the negative value to be 1-positive value (similarly if the negative value is larger).

[8]The comparison is done after lemmatization, using the wordNet lemmatization in NLTK, and with the same POS, according to the Stanford POStagger toolkit.

5.4.4 Discourse

For a sentence s, we assume the writer's sentiments toward the gfbf events in the clauses of s, the previous sentence, and the next sentence, are the same. Consider Ex(3):

Ex(**3**) ... health-insurance regulations that will prohibit (a) denying coverage for pre-existing conditions, (b) dropping coverage if the client gets sick, and (c) capping insurance company reimbursement...

Ex(3) has three clauses, (a)-(c). Suppose the explicit sentiment classifier recognizes that event (a), *denying coverage for pre-existing conditions*, is negative and it does not find any other explicit sentiments in the sentence. The system assumes the writer's sentiments toward (b) and (c) are negative as well.

After assigning all possible polarities to events within a sentence, polarities are propagated to the other still-neutral gfbf events in the previous and next sentences.

Finally, event-level polarities are propagated to still-neutral objects using Rule schema 1.[9] If there is a conflict, we take the majority sentiment; if there is a tie, the object remains neutral.

However, the confidence of the discourse voting is smaller than the explicit sentiment voting, since discourse structure is complex. If by discourse an object node is classified as positive, the positive value is $0.5 + random(0, 0.1)$ and the negative value is 1-positive value. Thus, the positive value of a positive node is larger than its negative value, but not exceeding too much (similarly for negative nodes).

6 Experiments and Results

6.1 Experiment Data

Of the 134 documents in the dataset, 6 were used as a development set, and 3 do not have any annotation. We use the remaining 125 for experiment.

6.2 Evaluation Metrics

To evaluate the performance of classifying the writer's sentiments toward agents and objects, we

[9]Note that, in the gfbf entity graph, sentiments can be propagated from objects to agents, conceptually via Rule schemas 2 and 3. Thus, here we only classify objects.

define three metrics to evaluate performance. For the entire dataset, accuracy evaluates the percentage of nodes that are classified correctly. Precision and recall are defined to evaluate polar (non-neutral) classification.

$$Accuracy = \frac{\text{\#node auto=gold}}{\text{\#nodes}} \tag{3}$$

$$Precision = \frac{\text{\#node auto=gold \& gold != neutral}}{\text{\#node auto != neutral}} \tag{4}$$

$$Recall = \frac{\text{\#node auto=gold \& gold != neutral}}{\text{\#node gold != neutral}} \tag{5}$$

In the equations, *auto* is the system's output and *gold* is the gold-standard label from annotation.

6.3 Overall Performance

In this section, we evaluate the performance of the overall system. In 6.5, we evaluate the graph model itself.

Two baselines are defined. One is assigning the majority class label, which is positive, to all agents/objects ($Majority(+)$). The second is assuming that agents/objects in a GOODFOR relation are positive and agents/objects in a BADFOR relation are negative ($GFBF$). In addition, we evaluate the explicit sentiment classifier introduced in Section 5.4 ($Explicit$). The results are shown in Table 3.

	Accuracy	Precision	Recall
Majority(+)	0.5438	0.5621	0.5443
GFBF	0.5437	0.5523	0.5444
Explicit	0.3703	0.5698	0.3703
Graph-LBP	0.5412	**0.6660**	0.5419

Table 3: Performance of baselines and graph.

As can be seen, $Majority$ and $GFBF$ give approximately 56% precision. $Explicit$ sentiment classification alone performs hardly better in precision and much lower in recall. As mentioned in Section 2, fine-grained sentiment analysis is still very difficult for NLP systems. However, the graph model improves greatly over $Explicit$ in both precision and recall. While recall of the graph model is comparable to the $Majority$, precision is much higher.

During the experiment, if the LBP does not converge until 100 iterations, it is forced to stop. The average number of iteration is 34.192.

6.4 Error Analysis

Table 4 shows the results of an error analysis to determine what contributes to the graph model's errors.

1	wrong sentiment from voting	0.2132
2	wrong sentiment from discourse	0.0462
3	subgraph with wrong polarity	0.3189
4	subgraph with no polarity	0.4160
5	other	0.0056

Table 4: Errors for graph model.

Rows 1-2 are the error sources for nodes assigned a polar value before graph propagation. Row 1 errors are due to the sentiment-voting system, Row 2 are due to discourse processing.

Rows 3-4 are the error sources for nodes that have not been assigned a polar value by $Explicit$. Such a node receives a polar value only via propagation from other nodes in its subgraph (i.e., the connected component of the graph containing the node). Row 5 is the percentage of other errors.

As shown in Rows 1-2, 25.94% of the errors are due to $Explicit$. These may propagate incorrect labels to other nodes in the graph. As shown in Row 3, 31.89% of the errors are due to nodes not classified polar by $Explicit$, but given incorrect values because their subgraph has an incorrect polarity. Row 4 shows that 41.60% of the errors are due to nodes that are not assigned any polar value. Given non-ideal input from sentiment analysis, how does the graph model increase precision by 10 percentage points?

There are two main ways. For nodes which remain neutral after $Explicit$, they might be classified correctly via the graph. For nodes which are given incorrect polar labels by $Explicit$, they might be fixed by the graph. Table 5 shows the best the graph model could do, given the noisy input from $Explicit$. Over all of the nodes, more propagated labels are incorrect than correct. However, if there are no incorrect, or more correct than incorrect sentiments in the subgraph (connected component), then many more of the propagated labels are correct than incorrect. In all cases, more of the changed labels are correct than incorrect.

6.5 Consistency and Isolated Performance of Graph Model

The implicature rules are defeasible. In this section we introduce an experiment to valid the con-

propagated label correct	propagated label incorrect	changed correctly	changed incorrectly
all subgraphs			
399	536	424	274
subgraphs having no incorrect sentiment			
347	41	260	23
subgraphs having more correct than incorrect sentiment			
356	42	288	35

Table 5: Effects of graph model given $Explicit$ input

sistency of implicature rule. Recall that in Section 5.3, the definition of $\Psi_{i,j}$ is based on implicature rules and sentiment is propagated based on $\Psi_{i,j}$. Thus, this is also an evaluation of the performance of the graph model itself. We performed an experiment to assess the chance of a node being correctly classified only via the graph.

In each subgraph (connected component), we assign one of the nodes in the subgraph with its gold-standard polarity. Then we run LBP on the subgraph and record whether the other nodes in the subgraph are classified correctly or not. The experiment is run on the subgraph $|S|$ times, where $|S|$ is the number of nodes in the subgraph, so that each node is assigned its gold-standard polarity exactly once. Each node is given a propagated value $|S| - 1$ times, as each of the other nodes in its subgraph receives its gold-standard polarity.

To evaluate the chance of a node given a correct propagated label, we use Equations (6) and (7).

$$correct(a|b) = \begin{cases} 1 & \textit{a is correct} \\ 0 & \textit{otherwise} \end{cases} \quad (6)$$

$$correctness(a) = \frac{\sum_{b \in S_a, b \neq a} correct(a|b)}{|S_a| - 1} \quad (7)$$

where S_a is the set of nodes in a's subgraph. Given b being assigned its gold-standard polarity, if a is classified correctly, then $correct(a|b)$ is 1; otherwise 0. $|S_a|$ is the number of nodes in a's subgraph. $correctness(a)$ is the percentage of assignments to a that are correct. If it is 1, then a is correctly classified given the correct classification of any single node in its subgraph.

For example, suppose there are three nodes in a subgraph, A, B and C. For A we (1) assign B its gold label and carry out propagation on the subgraph, (2) assign C its gold label and carry out propagation again, then (3) calculate $correctness(A)$. Then the same process is repeated for B and C.

Some subgraphs contain only two nodes, the agent and the object. In this case, graph propagation corresponds to single applications of two implicature rules. Other subgraphs contain more nodes. Two results are shown in Table 6. One is the result on the whole experiment data, the other is the result for all nodes whose subgraphs have more than two nodes.

Dataset	# subgraph	correctness
all subgraphs	983	0.8874
multi-node subgraphs	169	0.9030

Table 6: Performance of graph model itself.

As we can see, a node has an 89% chance of being correct if there is one correct explicit subjectivity node in its subgraph. If we only consider subgraphs with more than two nodes, the correctness chance is higher. The results indicate that, if given correct sentiments, the graph model will assign the unknown nodes with correct labels 90% of the time. Further, the results indicate that the implicature rules are consistent for most of the times across the corpus.

7 Conclusions

We developed a graph-based model based on implicature rules to propagate sentiments among entities. The model improves over explicit sentiment classification by 10 points in precision and, in an evaluation of the model itself, we find it has an 89% chance of propagating sentiments correctly. An important question for future work is under what conditions do the implicatures not go through in context. Two cases we have discovered involve Rule schema 3: the inference toward the agent is defeated if the action was accidental or if the agent was forced to perform it. We are investigating lexical clues for recognizing such cases.

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 Richard Johansson and Alessandro Moschitti for running their Opinion Extraction systems on our data.

References

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

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 793–801, Honolulu, Hawaii, October. Association for Computational Linguistics.

Pradeep Dasigi, Weiwei Guo, and Mona Diab. 2012. Genre independent subgroup detection in online discussion threads: A study of implicit attitude using textual latent semantics. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 65–69, Jeju Island, Korea, July. Association for Computational Linguistics.

Lingjia Deng, Yoonjung Choi, and Janyce Wiebe. 2013. Benefactive and malefactive event and writer attitude annotation. In *51st Annual Meeting of the Association for Computational Linguistics (ACL-2013, short paper)*.

Song Feng, Jun Sak Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria, Angust. Association for Computational Linguistics.

Amit Goyal, Ellen Riloff, and Hal Daum III. 2012. A computational model for plot units. *Computational Intelligence*, pages 466–488.

Rodney D. Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, April.

Richard Johansson and Alessandro Moschitti. 2013. Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3).

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool.

Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *Proceedings of RANLP 2007*, Borovets, Bulgaria.

Karo Moilanen, Stephen Pulman, and Yue Zhang. 2010. Packed feelings and ordered sentiments: Sentiment parsing with quasi-compositional polarity sequencing and compression. In *Proceedings of the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2010)*, pages 36–43.

J. Pearl. 1982. Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, pages 133–136, Pittsburgh, PA.

Kevin Reschke and Pranav Anand. 2011. Extracting contextual evaluativity. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 370–374, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA, October. Association for Computational Linguistics.

P.J. Stone, D.C. Dunphy, M.S. Smith, and D.M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1397–1405. ACM.

Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming zhou, and Ming Zhang. 2011. Topic sentiment anaylsis in twitter: A graph-based hashtag sentiment classification appraoch. In *CIKM*, pages 1031–1040.

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

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

Jonathan S Yedidia, William T Freeman, and Yair Weiss. 2005. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312.

Lei Zhang and Bing Liu. 2011. Identifying noun product features that imply opinions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 575–580, Portland, Oregon, USA, June. Association for Computational Linguistics.

F. Zúñiga and S. Kittilä. 2010. Introduction. In F. Zúñiga and S. Kittilä, editors, *Benefactives and malefactives*, Typological studies in language. J. Benjamins Publishing Company.