Joint Prediction for Entity/Event-Level Sentiment Analysis using Probabilistic Soft Logic Models

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

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

Abstract

In this work, we build an entity/event-level sentiment analysis system, which is able to recognize and infer both explicit and implicit sentiments toward entities and events in the text. We design Probabilistic Soft Logic models that integrate explicit sentiments, inference rules, and +/-effect event information (events that positively or negatively affect entities). The experiments show that the method is able to greatly improve over baseline accuracies in recognizing entity/event-level sentiments.

1 Introduction

There are increasing numbers of opinions expressed in various genres, including reviews, newswire, editorials, and forums. While much early work was at the document or sentence level, to fully understand and utilize opinions, researchers are increasingly carrying out more fine-grained sentiment analysis to extract components of **opinion frames**: the source (whose sentiment is it), the polarity, and the target (what is the sentiment toward). Much fine-grained analysis is span or aspect based (Yang and Cardie, 2014; Pontiki et al., 2014). In contrast, this work contributes to **entity/event**-level sentiment analysis. A system that could recognize sentiments toward entities and events would be valuable in an application such as Automatic Question Answering, to support answering questions such as "Who is negative/positive toward X?" (Stoyanov et al., 2005), where X could be any entity or event.

Let us consider an example from the MPQA opinion annotated corpus (Wiebe et al., 2005a; Wilson, 2007; Deng and Wiebe, 2015).

Ex(1) When the Imam (may God be satisfied with him $_1$) issued the fatwa against $_2$ Salman Rushdie for insulting $_3$ the Prophet (peace be upon him $_4$), the countries that are so-called $_5$ supporters of human rights protested against $_6$ the fatwa.

There are several sentiment expressions annotated in MPQA. In the first clause, the writer is positive toward Imam and Prophet as expressed by *may God be satisfied with him* (1) and *peace be upon him* (4), respectively. Imam is negative toward Salman Rushdie and the insulting event, as revealed by the expression *issued the fatwa against* (2). And Salman Rushdie is negative toward *Prophet*, as revealed by the expression *insulting* (3). In the second clause, the writer is negative toward the countries, as expressed by *so-called* (5). And the countries are negative toward fatwa, as revealed by the expression *protested against* (6). Using the source and the target, we summarize the positive opinions above in a set P, and the negative opinions above in another set N. Thus, P contains {(writer, Imam), (writer, Prophet)}, and N contains {(Imam, Rushdie), (Imam, insulting), (Rushdie, Prophet), (writer, countries), (countries, fatwa)}.[1]

An (ideal) explicit sentiment analysis system is expected to extract the above sentiments expressed by (1)-(6). However, there are many more sentiments communicated by the writer but not expressed via explicit expressions. First, Imam is positive toward the Prophet, because Rushdie insults the Prophet and Imam is angry that he does

[1] Sources in MPQA are nested, having the form ⟨writer⟩ or ⟨writer, $S_1, \ldots, S_n$⟩. This work only deals with the rightmost source, writer or S_n. Also, actions like *issuing a fatwa* are treated the same as private states. Please see (Wiebe et al., 2005a).

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 179–189,
Lisbon, Portugal, 17-21 September 2015.

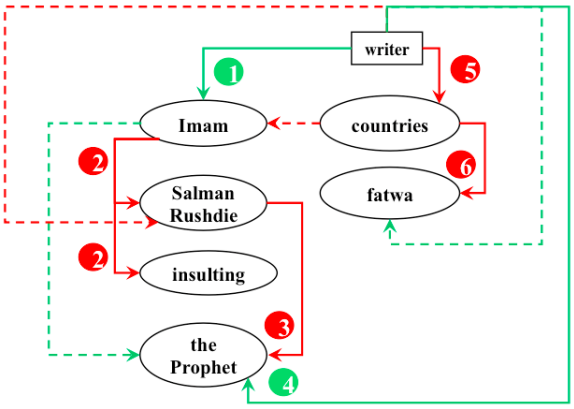

Figure 1: Explicit and implicit sentiments in Ex(1).

so. Second, the writer is negative toward Rushdie, because the writer is positive toward the Prophet but Rushdie insults him! Also, the writer is probably positive toward the fatwa since it is against Rushdie. Third, the countries are probably negative toward Imam, because the countries are negative toward fatwa and it is Imam who issued the fatwa. Thus, the set P should also contain {(Imam, Prophet), (writer, fatwa)}, and the set N should also contain {(writer, Rushdie), (countries, Imam)}. These opinions are not directly expressed, but are **inferred** by a human reader.[2] The explicit and implicit sentiments are summarized in Figure 1, where each green line represents a positive sentiment and each red line represents a negative sentiment. The solid lines are explicit sentiments and the dashed lines are implicit sentiments.

In this work, we detect sentiments such as those in P and N, where the sources are entities (or the writer) and the targets are entities and events.

Previous work in sentiment analysis mainly focuses on detecting explicit opinions. Recently there is emerging focus on sentiment inference, which recognizes implicit sentiments by inferring them from explicit sentiments via inference rules. Current works in sentiment inference differ on how the sentiment inference rules are defined and how they are expressed. For example, Zhang and Liu (2011) define linguistic templates to recognize phrases that express implicit sentiments, while previously we (Deng et al., 2014) represent a few simple rules as (in)equality constraints in Integer Linear Programming. In contrast to previous work, we propose a more general set of inference rules and encode them in a probabilistic soft logic (PSL) framework (Bach et al., 2015). We chose PSL because it is designed to have efficient inference and, as similar methods in Statistical Relational Learning do, it allows probabilistic models to be specified in first-order logic, an expressive and natural way to represent if-then rules, and it supports joint prediction. Joint prediction is critical for our task because it involves multiple, mutually constraining ambiguities (the source, polarity, and target).

Thus, this work aims at detecting both implicit and explicit sentiments expressed by an entity toward another entity/event (i.e., an **eTarget**) within the sentence. The contributions of this work are: (1) defining a method for entity/event-level sentiment analysis to provide a deeper understanding of the text; (2) exploiting first-order logic rules to infer such sentiments, where the source is not limited to the writer, and the target may be any entity, event, or even another sentiment; and (3) developing a PSL model to jointly resolve explicit and implicit sentiment ambiguities by integrating inference rules.

2 Related Work

Fined-grained sentiment analysis. Most fine-grained sentiment analysis is span or aspect based. Previous work differs from the entity/event-level sentiment analysis task we address in terms of targets and sources. In terms of targets, in a span-based sentiment analysis system, the target is a span instead of the exact head of the phrase referring to the target. The target in a span-based system is evaluated by measuring the overlapping proportion of an extracted span against the gold standard phrase (Yang and Cardie, 2013), while the eTarget in an entity/event-level system is evaluated against the exact word (i.e., head of NP/VP) in the gold standard. It is a stricter evaluation. While the targets in aspect-based sentiment analysis are often entity targets, they are mainly product aspects, which are a predefined set.[3] In contrast, the target in the entity/event-level task may be any noun or verb. In terms of sources, previous work in sentiment analysis trained on review data assumes that the source is the writer of the review (Hu and Liu, 2004; Titov and McDonald, 2008).

[2] Note that the inferences are conversational implicatures; they are defeasible and may not go through in context (Deng et al., 2014; Wiebe and Deng, 2014).

[3] As stated in SemEval-2014: "we annotate only aspect terms naming particular aspects".

Our work is rare in that it allows sources other than the writer *and* finds sentiments toward eTargets which may be any entity or event.

Sentiment Inference. There is some recent work investigating features that directly indicate implicit sentiments (Zhang and Liu, 2011; Feng et al., 2013). That work assumes the source is only the writer. Further, as it uses features to directly extract implicit sentiments, it does not perform general sentiment inference.

Previously, we (Deng et al., 2013; Deng and Wiebe, 2014; Deng et al., 2014) develop rules and models to infer sentiments related to *+/-effect events*, events that positively or negatively affect entities. That work assumes that the source is only the writer, and the targets are limited to entities that participate in +/-effect events. Further, our previous models all require certain manual (oracle) annotations to be input. In this work we use an expanded set of more general rules. We allow sources other than the writer, and targets that may be any entity or event. In fact, under our new rules, the targets of sentiments may be other sentiments; we model such novel "sentiment toward sentiment" structures in Section 4.3. Finally, our method requiring no manual annotations as input when the inference is conducted.

Previously, we also propose a set of sentiment inference rules and develop a rule-based system to infer sentiments (Wiebe and Deng, 2014). However, the rule-based system requires *all* information regarding explicit sentiments and +/-effect events to be provided as oracle information by manual annotations.

Probabilistic Soft Logic. Probabilistic Soft Logic (PSL) is a variation of Markov Logic Networks, which is a framework for probabilistic logic that employs weighted formulas in first-order logic to compactly encode complex undirected probabilistic graphical models (i.e., Markov networks) (Bach et al., 2015; Beltagy et al., 2014). PSL is a new statistical relational learning method that has been applied to many NLP and other machine learning tasks in recent years (Beltagy et al., 2014; London et al., 2013; Pujara et al., 2013; Bach et al., 2013; Huang et al., 2013; Memory et al., 2012). Previously, PSL has not been applied to entity/event-level sentiment analysis.

3 Task Definition

In this section, we introduce the definition of the entity/event-level sentiment analysis task, followed by a description of the gold standard corpus.

For each sentence s, we define a set E consisting of entities, events, and the writer of s, and sets P and N consisting of positive and negative sentiments, respectively. Each element in P is a tuple, representing a **positive pair** of two entities, (e_1, e_2) where e_1, $e_2 \in E$, and e_1 is positive toward e_2. A positive pair (e_1,e_2) aggregates all the positive sentiments from e_1 to e_2 in the sentence. N is the corresponding set for **negative pairs**.

The goal of this work is to automatically recognize a set of positive pairs (P_{auto}) and a set of negative pairs (N_{auto}). We compare the system output ($P_{\text{auto}} \cup N_{\text{auto}}$) against the gold standard ($P_{\text{gold}} \cup N_{\text{gold}}$) for each sentence.

3.1 Gold Standard Corpus: MPQA 3.0

MPQA 3.0 is a recently developed corpus with entity/event-level sentiment annotations (Deng and Wiebe, 2015).[4] It is built on the basis of MPQA 2.0 (Wiebe et al., 2005b; Wilson, 2007), which includes editorials, reviews, news reports, and scripts of interviews from different news agencies, and covers a wide range of topics.

In both MPQA 2.0 and 3.0, the top-level annotations include **direct subjective**s (**DS**). Each DS has a **nested-source** annotation. Each DS has one or more attitude links, meaning that all of the attitudes share the same nested source. The attitudes differ from one another in their attitude types, polarities, and/or targets. Moreover, both corpora contain **expressive subjective element** (**ESE**) annotations, which pinpoint specific expressions used to express subjectivity. We ignore neutral ESEs and only consider ESEs whose polarity is positive or negative.

MPQA 2.0 and 3.0 differ in their target annotations. In 2.0, each target is a span. A target annotation of an opinion captures the most important target this opinion is expressed toward. Since the exact boundaries of the spans are hard to define even for human annotators (Wiebe et al., 2005a; Yang and Cardie, 2013), the target span in MPQA 2.0 could be a single word, an NP or VP, or a text span covering more than one constituent. In contrast, in MPQA 3.0, each target is anchored to the head of an NP or VP, which is a single word. It is called an

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

eTarget since it is an entity or an event. In MPQA 2.0, only attitudes have target-span annotations. In MPQA 3.0, both attitudes and ESEs have eTarget annotations. Importantly, the eTargets include the targets of both explicit and implicit sentiments.

Recall Ex(1) in Section 1. $P_{\text{gold}} = \{$(writer, Imam), (writer, Prophet), (Imam, Prophet), (writer, fatwa)$\}$, and $N_{\text{gold}} = \{$(Imam, Rushdie), (Imam, insulting), (Rushdie, Prophet), (writer, countries), (countries, fatwa), (writer, Rushdie), (countries, Imam)$\}$.

4 PSL for Sentiment Analysis

We need to resolve three components for an opinion frame: the source, the polarity, and the eTarget. Each of these ambiguities has several candidates. For example in Ex(1), the eTarget of the opinion expression *insulting* is an ambiguity. The candidates include *Prophet*, *countries*, and so on.

In this work, we use Probabilistic Soft Logic (PSL). A PSL model is defined using a set of atoms to be grounded, and a set of weighted if-then rules expressed in first-order logic. For example, we define the atom ETARGET(y,t) to represent an opinion y having eTarget t. If y and t are constants, then ETARGET(y,t) is a ground atom (e.g., ETARGET(insulting, Prophet)). Each ground atom is assigned a score by a local system. PSL takes as input all the local scores as well as the constraints defined by the rules among atoms, so that it is able to jointly resolve all the ambiguities. In the final output, for example, the score ETARGET(insulting, Prophet) > 0 means that PSL considers Prophet to be an eTarget of *insulting*, while ETARGET(insulting, countries) $= 0$ means that PSL does not consider *countries* to be an eTarget of *insulting*.

In this section, we first introduce PSL in Section 4.1. We then present three PSL models in turn. PSL1 (Section 4.2) aggregates span-based opinions into P_{auto} and N_{auto}. PSL2 (Section 4.3) adds sentiment inference rules to PSL1. For PSL3 (Section 4.4), rules involving +/-effect events are added to PSL2, resulting in the richest overall model.

4.1 Probabilistic Soft Logic

PSL (Bach et al., 2015) uses logical representations to compactly define large graphical models with continuous variables, and includes methods for performing efficient probabilistic inference for the resulting models (Beltagy et al., 2014). As mentioned above, a PSL model is defined using a set of atoms to be grounded, and a set of weighted if-then rules in first-order logic. For example,

friend(x,y) $\wedge$ votesFor(y,z) $\Rightarrow$ votesFor(x,z)

means that a person may vote for the same person as his/her friend. Each predicate in the rule is an atom (e.g., friend(x,y)). A ground atom is produced by replacing variables with constants (e.g., friend(Tom, Mary)). Each rule is associated with a weight, indicating the importance of this rule in the whole rule set.

A key distinguishing feature of PSL is that each ground atom a has a soft, continuous truth value in the interval [0, 1], denoted as $I(a)$, rather than a binary truth value as in Markov Logic Networks and most other probabilistic logic frameworks (Beltagy et al., 2014). To compute soft truth values for logical formulas, Lukasiewicz relaxations are used:

$l_1 \wedge l_2 = \max\{0, I(l_1) + I(l_2) - 1\}$
$l_1 \vee l_2 = \min\{I(l_1) + I(l_2), 1\}$
$\neg l_1 = 1 - I(l_1)$

A rule $r \equiv r_{body} \rightarrow r_{head}$, is satisfied (i.e. $I(r) = 1$) iff $I(r_{body}) \leq I(r_{head})$. Otherwise, a distance to satisfaction $d(r)$ is calculated, which defines how far a rule r is from being satisfied: $d(r) = \max\{0, I(r_{body}) - I(r_{head})\}$. Using $d(r)$, PSL defines a probability distribution over all possible interpretations I of all ground atoms:

$$p(I) = \frac{1}{Z} \exp\{-1 * \sum_{r \in R} \lambda_r (d(r))^p\}$$

where Z is the normalization constant, λ_r is the weight of rule r, R is the set of all rules, and p defines loss functions. PSL seeks the interpretation with the minimum distance $d(r)$ and which satisfies all rules to the extent possible.

4.2 PSL for Sentiment Aggregation (PSL1)

The first PSL model, PSL1, aggregates span-based opinions into P_{auto} and N_{auto}. We call this *sentiment aggregation* because, instead of building an entity/event-level sentiment system from scratch, we choose to fully utilize previous work on span-based sentiment analysis. PSL1 aggregates span-based opinions into entity/event-level opinions.

Consistent with the task definition in Section 3, we define two atoms in PSL:
(1) POSPAIR(s,t): a positive pair from s toward t
(2) NEGPAIR(s,t): a negative pair from s toward t

Both s and t are chosen from the set E. The values of ground atoms (1) and (2) are not observed and are inferred by PSL.

Then, we define atoms to model an entity/event-level opinion:

(3) POS(y): y is a positive sentiment
(4) NEG(y): y is a negative sentiment
(5) SOURCE(y,s): the source of y is s
(6) ETARGET(y,t): the eTarget of y is t

Two rules are defined to aggregate various opinions extracted by span-based systems into positive pairs and negative pairs, shown in Part 1 of Table 1 as Rules 1.1 and 1.2. Thus, under our representation, the PSL model not only finds a set of eTargets of an opinion (ETARGET(y,t)), but also represents the aggregated sentiments among entities and events (POSPAIR(s,t) and NEGPAIR(s,t)) in the sentence.

Next, we turn to assigning local scores to ground atoms (3)-(6).

POS(y) and NEG(y): We build upon three span-based sentiment analysis systems. The first, S1 (Yang and Cardie, 2013), and the second, S2 (Yang and Cardie, 2014), are both trained on MPQA 2.0, which does not contain any eTarget annotations. S1 extracts triples of ⟨source span, opinion span, target span⟩, but does not extract opinion polarities. S2 extracts opinion spans and opinion polarities, but it does not extract sources or targets. The third system, S3 (Socher et al., 2013), is trained on movie review data. It extracts opinion spans and polarities. The source is always assumed to be the writer.

We take the union set of opinions extracted by S1, S2 and S3. For each opinion y, a ground atom is created, depending on the polarity (POS(y) if y is positive and NEG(y) is y is negative). The polarity is determined as follows. If S2 assigns a polarity to y, then that polarity is used. If S3 but not S2 assigns a polarity to y, then S3's polarity is used. In both cases, the score assigned to the ground atom is 1.0. If neither S2 nor S3 assigns a polarity to y, we use the MPQA subjectivity lexicon to determine its polarity. The score assigned to the ground atom is the proportion of the words in the opinion span that are included in the subjectivity lexicon.

SOURCE(y,s): S1 extracts the source of each opinion, S2 does not extract the source, and S3 assumes the source is always the writer. Thus, for an opinion y, if the source s is assigned by S1, a ground atom SOURCE(y,s) is created with score 1.0. Otherwise, if S3 extracts opinion y, a ground atom SOURCE(y,writer) is created with score 1.0 (since S3 assumes the source is always the writer). Otherwise, we run the Stanford named entity recognizer (Manning et al., 2014; Finkel et al., 2005) to extract named entities in the sentence. The nearest named entity to the opinion span on the dependency parse graph will be treated as the source. The score is the reciprocal of the length of the path between the opinion span and the source span in the dependency parse.

ETARGET(y,t): Though each eTarget is an entity or event, it is difficult to determine which nouns and verbs should be considered. Taking into consideration the trade-off between precision and recall, we experimented with three methods to select eTarget candidates. For each opinion y, a ground atom ETARGET(y,t) is created for each eTarget candidate t.

ET1 considers all the nouns and verbs in the sentence, to provide a full recall of eTargets.

ET2 considers all the nouns and verbs in the target spans and opinion spans that are automatically extracted by systems S1, S2 and S3. We hypothesized that ET2 would be useful because most of the eTargets in MPQA 3.0 appear within the opinion or the target spans of MPQA 2.0.

ET3 considers the heads of the target and opinion spans that are automatically extracted by systems S1, S2 and S3.[5] ET3 also considers the heads of siblings of target spans and opinion spans. Among the three methods, ET3 has the lowest recall but the highest precision.

In addition, for the eTarget candidate set extracted by ET2, or ET3, we run the Stanford co-reference system (Manning et al., 2014; Recasens et al., 2013; Lee et al., 2013) to expand the set in two ways. First, for each eTarget candidate t, the co-reference system extracts the entities that co-refer with t. We add the referring entities into the candidate set. Second, the co-reference system extracts words which the Stanford system judges to be entities, regardless of whether they have any referent or not. We add this set of entities to the candidate set as well.

We train an SVM classifier (Cortes and Vapnik, 1995) to assign a score to the ground atom ETARGET(y,t). Syntactic features describing the

[5]The head of a phrase is extracted by the Collins head finder in the Stanford parser (Manning et al., 2014).

Part 1.	Aggregation Rules.	
1.1	SOURCE(y,s) $\wedge$ ETARGET(y,t) $\wedge$ POS(y)	$\Rightarrow$ POSPAIR(s,t)
1.2	SOURCE(y,s) $\wedge$ ETARGET(y,t) $\wedge$ NEG(y)	$\Rightarrow$ NEGPAIR(s,t)
Part 2.	Inference Rules.	
2.1	POSPAIR(s_1,y_2) $\wedge$ SOURCE(y_2,s_2)	$\Rightarrow$ POSPAIR(s_1,s_2)
2.2	POSPAIR(s_1,y_2) $\wedge$ ETARGET(y_2,t_2) $\wedge$ POS(y_2)	$\Rightarrow$ POSPAIR(s_1,t_2)
2.3	POSPAIR(s_1,y_2) $\wedge$ ETARGET(y_2,t_2) $\wedge$ NEG(y_2)	$\Rightarrow$ NEGPAIR(s_1,t_2)
2.4	NEGPAIR(s_1,y_2) $\wedge$ SOURCE(y_2,s_2)	$\Rightarrow$ NEGPAIR(s_1,s_2)
2.5	NEGPAIR(s_1,y_2) $\wedge$ ETARGET(y_2,t_2) $\wedge$ POS(y_2)	$\Rightarrow$ NEGPAIR(s_1,t_2)
2.6	NEGPAIR(s_1,y_2) $\wedge$ ETARGET(y_2,t_2) $\wedge$ NEG(y_2)	$\Rightarrow$ POSPAIR(s_1,t_2)
Part 3.	Inference Rules w.r.t +/-Effect Event Information.	
3.1	POSPAIR(s,x) $\wedge$ AGENT(x,a)	$\Rightarrow$ POSPAIR(s,a)
3.2	POSPAIR(s,x) $\wedge$ THEME(x,h) $\wedge$ +EFFECT(x)	$\Rightarrow$ POSPAIR(s,h)
3.3	POSPAIR(s,x) $\wedge$ THEME(x,h) $\wedge$ -EFFECT(x)	$\Rightarrow$ NEGPAIR(s,h)
3.4	NEGPAIR(s,x) $\wedge$ AGENT(x,a)	$\Rightarrow$ NEGPAIR(s,a)
3.5	NEGPAIR(s,x) $\wedge$ THEME(x,h) $\wedge$ +EFFECT(x)	$\Rightarrow$ NEGPAIR(s,h)
3.6	NEGPAIR(s,x) $\wedge$ THEME(x,h) $\wedge$ -EFFECT(x)	$\Rightarrow$ POSPAIR(s,h)
3.7	POSPAIR(s,a) $\wedge$ AGENT(x,a)	$\Rightarrow$ POSPAIR(s,x)
3.8	POSPAIR(s,h) $\wedge$ THEME(x,h) $\wedge$ +EFFECT(x)	$\Rightarrow$ POSPAIR(s,x)
3.9	POSPAIR(s,h) $\wedge$ THEME(x,h) $\wedge$ -EFFECT(x)	$\Rightarrow$ NEGPAIR(s,x)
3.10	NEGPAIR(s,a) $\wedge$ AGENT(x,a)	$\Rightarrow$ NEGPAIR(s,x)
3.11	NEGPAIR(s,h) $\wedge$ THEME(x,h) $\wedge$ +EFFECT(x)	$\Rightarrow$ NEGPAIR(s,x)
3.12	NEGPAIR(s,h) $\wedge$ THEME(x,h) $\wedge$ -EFFECT(x)	$\Rightarrow$ POSPAIR(s,x)

Table 1: Rules in First-Order Logic.

relations between an eTarget and the extracted opinion span and target span are considered, including: (1) whether the eTarget is in the opinion/target span; (2) the unigrams and bigrams on the path from the eTarget to the opinion/target span in the constituency parse tree; and (3) the unigrams and bigrams on the path from the eTarget to the opinion/target word in the dependency parse graph. We normalize the SVM scores into the range of a ground atom score, [0,1].

4.3 PSL for Sentiment Inference (PSL2)

The two rules defined in Section 4.2 aggregate various opinions into positive pairs and negative pairs, but inferences have not yet been introduced. PSL2 is defined using the atoms and rules in PSL1. But it also includes some rules defined in (Wiebe and Deng, 2014), represented here in first-order logic in Part 2 of Table 1. Let us go through an example inference for Ex(1), in particular, the inference that Imam is positive toward the Prophet. Rule 2.6 supports this inference. Recall the two explicit sentiments: Imam is negative toward the insulting sentiment (revealed by *issued the fatwa against*), and Rushdie is negative toward the Prophet (revealed by *insulting*). Thus, we can instantiate Rule 2.6, where s_1 is Imam, y_2 is the negative sentiment (*insulting*), and t_2 is the Prophet. The inference is: since Imam is negative that there is any negative opinion expressed toward the Prophet, we infer that Imam is positive toward the Prophet.

NEGPAIR(Imam, insulting)
$\wedge$ ETARGET(insulting, Prophet)
$\wedge$ NEG(insulting)
$\Rightarrow$ POSPAIR(Imam, Prophet).

The inference rules in Part 2 of Table 1 are novel in that eTargets may be sentiments (e.g., NEGPAIR(Imam,insulting) means that Imam is negative toward the negative sentiment revealed by *insulting*). The inference rules link sentiments to sentiments and, transitively, link entities to entities (e.g., from Imam to Rushdie to the Prophet).

To support such rules, more groundings of ETARGET(y,t) are created in PSL2 than in PSL1. For two opinions y_1 and y_2, if the target span of y_1 overlaps with the opinion span of y_2, we create ETARGET(y_1,y_2) as a ground atom representing that y_2 is an eTarget of y_1.

4.4 PSL Augmented with +/-Effect Events (PSL3)

Finally, for PSL3, +/-effect event atoms and rules are added to PSL2 for the inference of additional sentiments.

According to (Deng et al., 2013), a +effect event has positive effect on the theme (examples are *help*, *increase*, and *save*), and a -effect event has negative effect on the theme (examples are *obstruct*, *decrease*, and *kill*).[6] We define the following atoms to represent such events:

(7) +EFFECT(x): x is a +effect event
(8) -EFFECT(x): x is a -effect event
(9) AGENT(x,a): the agent of x is a
(10) THEME(x,h): the theme of x is h

Next we assign scores to these ground atoms.

+EFFECT(x) and -EFFECT(x): We use the +/-effect sense-level lexicon (Choi and Wiebe, 2014)[7] to extract the +/-effect events in each sentence. The score of +EFFECT(x) is the fraction of that word's senses that are +effect senses according to the lexicon, and the score of -EFFECT(x) is the fraction of that word's senses that are -effect senses according to the lexicon. If a word does not appear in the lexicon, we do not treat it as a +/-effect event, and thus assign 0 to both +EFFECT(x) and -EFFECT(x).

AGENT(x,a) and THEME(x,h): We consider all nouns in the same or in sibling constituents of a +/-effect event as potential agents or themes. An SVM classifier is run to assign scores to AGENT(x,a), and another SVM classifier is run to assign scores to THEME(x,h). Both SVM classifiers are trained on a separate corpus, the +/-effect corpus (Deng et al., 2013) used in (Deng et al., 2014), which is annotated with +/-effect event, agent, and theme spans. The features we use to train the agent and theme classifier include unigram, bigram and syntax information.

Generalizations of the inference rules used in (Deng et al., 2014) are expressed in first-order logic, shown in Part 3 of Table 1. Let us go through an example inference for Ex(1), in particular, the inference that the countries are negative toward Imam. Recall that we infer this because the countries are negative toward the fatwa and it is Imam who issued the fatwa. The rules supporting this inference are Rules 3.11 and 3.4 in Table 1, where s is the countries, h is the fatwa, x is the issue event, and a is Imam.

The application of Rule 3.11 can be explained as follows. The countries are negative toward the fatwa, and the issue event is a +effect event with theme fatwa (the issue event is +effect for the fatwa because it creates the fatwa; creation is one type of +effect event identified in (Deng et al., 2013)); thus, the countries are negative toward the issue event.

NEGPAIR(countries, fatwa)
$\wedge$ THEME(issue, fatwa)
$\wedge$ +EFFECT(issue)
$\Rightarrow$ NEGPAIR(countries, issue) .

The application of Rule 3.4 can be explained as follows. The countries are negative toward the issue event, and it is Imam who conducted the event; thus, the countries are negative toward Imam.

NEGPAIR(countries, issue)
$\wedge$ AGENT(issue, Imam)
$\Rightarrow$ NEGPAIR(countries, Imam) .

Finally, to support the new inferences, more groundings of ETARGET(y,t) are defined in PSL3. For a +/-effect event x whose agent is a, if one of x and a is an eTarget candidate of y, the other will be added to the eTarget candidate set for y (sentiments toward both +effect and -effect events and their agents have the same polarity according to the rules (Deng et al., 2014)). For +effect event x whose theme is h, if one of x and h is an eTarget candidate of y, the other is added to the eTarget candidate set for y (sentiments toward +effect events and their themes have the same polarity).

5 Experiments

We carry out experiments on the MPQA 3.0 corpus. Currently, there are 70 documents, 1,634 sentences, and 1,921 DS and ESEs in total. The total number of POSPAIR(s,t) and NEGPAIR(s,t) are 867 and 1,975, respectively. Though the PSL inference does not need supervision and the SVM classifier for agents and themes in Section 4.4 is trained on a separate corpus, we still have to train the eTarget SVM classifier to assign local scores as described in Section 4.2. Thus, the experiments are carried out using 5-fold cross validation. For each fold test set, the eTarget classifier is trained on the other folds. The trained classifier is then run on the test set, and PSL inference is carried out on the test set.

[6] In (Deng et al., 2013), such events are called *goodFor/badFor* events; they are later renamed as *+/-effect* events.

[7] Available at: http://mpqa.cs.pitt.edu/lexicons/effect_lexicon/

In total, we have three methods for eTarget candidate selection (ET1, ET2, ET3) and three models for sentiment analysis (PSL1, PSL2, PSL3).

Baselines. Since each noun and verb may be an eTarget, the first baseline (All NP/VP) regards all the nouns and verbs as eTargets. The first baseline estimates the difficulty of this task.

The second baseline (SVM) uses the SVM local classification results from Section 4.2. The score of ETARGET(y,t) is assigned by the SVM classifier. Then it is normalized as input into PSL. Before normalization, if the score assigned by the SVM classifier is above 0, the SVM baseline considers it as a correct eTarget.

5.1 Evaluations

First, we examine the performance of the PSL models on correctly recognizing eTargets of a particular opinion. This evaluation is carried out on a subset of the corpus: we only examine the opinions which are automatically extracted by the span-based systems (S1, S2 and S3). If an opinion expression in the gold standard is not extracted by any span-based system, it is not input into PSL, so PSL cannot possibly find its eTargets.

The second and third evaluations assess performance of the PSL models on correctly extracting positive and negative pairs. Note that our sentiment analysis system has the capability, through inference, to recognize positive and negative pairs even if corresponding opinion expressions are not extracted. Thus, the second and third evaluations are carried out on the entire corpus. The second evaluation uses ET3, and compares PSL1, PSL2 and PSL3. The third evaluation uses PSL3 and compares performance using ET1, ET2 and ET3. The results for the other combinations follow the same trends.

ETargets of An Opinion. According to the gold standard in Section 3.1, each opinion has a set of eTargets. But not all eTargets are equally important. Thus, our first evaluation assesses the performance of extracting the most important eTarget. As introduced in Section 3.1, a span-based target annotation of an opinion in MPQA 2.0 captures the most important target this opinion is expressed toward. Thus, the head of the target span can be considered to be the most important eTarget of an opinion. We model this as a ranking problem to compare models. For an opinion y automatically extracted by a span-based system, both the SVM baseline and PSL assign scores to ETARGET(y,t). We rank the eTargets according to the scores. Because the ALL NP/VP baseline does not assign scores to the nouns and verbs, we do not compare with that baseline in this ranking experiment. We use the Precision@N evaluation metric. If the top N eTargets of an opinion contain the head of target span, we consider it as a correct hit. The results are in Table 2.

	Prec@1	Prec@3	Prec@5
SVM	0.0370	0.0556	0.0820
PSL1	0.5105	0.6905	0.7831
PSL2	0.5317	0.7486	0.7883
PSL3	0.5503	0.7434	0.8148

Table 2: Precision@N of Most Important ETarget.

Table 2 shows that SVM is poor at ranking the most important eTarget. The PSL models are much better, even PSL1, which does not include any inference rules. This shows that SVM, which only uses local features, cannot distinguish the most important eTarget from the others. But the PSL models consider all the opinions, and can recognize a true negative even if it ranks high in the local results. The ability of PSL to rule out true negative candidates will be repeatedly shown in the later evaluations.

We not only evaluate the ability to recognize the most important eTarget of a particular opinion, we also evaluate the ability to extract all the eTargets of that opinion. The F-measure of SVM is 0.2043, while the F-measures of PSL1, PSL2 and PSL3 are 0.3135, 0.3239, and 0.3275, respectively. Correctly recognizing all the eTargets is difficult, but all the PSL models are better than the baseline.

Positive Pairs and Negative Pairs. Now we evaluate the performance in a stricter way. We compare automatically extracted sets of sentiment pairs: $P_{\text{auto}} = \{\text{POSPAIR}(s,t) > 0\}$ and $N_{\text{auto}} = \{\text{NEGPAIR}(s,t) > 0\}$, against the gold standard sets P_{gold} and N_{gold}. Table 3 shows the accuracies using ET3. Note that higher accuracies can be achieved, as shown later. Here we use ET3 just to show the trend of results.

As shown in Table 3, the low accuracy of baseline All NP/VP shows that entity/event-level sentiment analysis is a difficult task. Even the SVM baseline does not have good accuracy. Note that the SVM baseline in Table 3 uses ET3. The baseline classifies the heads of target spans and opin-

	PosPair	NegPair
All NP/VP	0.1280	0.1654
SVM	0.0765	0.0670
PSL1	0.3356	0.3754
PSL2	0.3705	0.3705
PSL3	0.4315	0.3892

Table 3: Accuracy comparing PSL models (ET3 used for all)

ion spans, which are extracted by state-of-the-art span-based sentiment analysis systems. This shows the results from span-based sentiment analysis systems do not provide enough accurate information for the more fine-grained entity/event-level sentiment analysis task. In contrast, PSL1 achieves much higher accuracy than the baselines. PSL2 and PSL3, which add sentiment toward sentiment and +/-effect event inferences, give further improvements. A reason is that SVM uses a hard constraint to cut off many eTarget candidates, while the PSL models take the scores as soft constraints.

A more critical reason is due to the definition of accuracy: (TruePositive+TrueNegative)/All. A significant benefit of using PSL is correctly recognizing true negative eTarget candidates and eliminating them from the set. Interestingly, even though both PSL2 and PSL3 introduce more eTarget candidates, both are able to recognize more true negatives and improve the accuracy.

Note that F-measure does not count true negatives. Precision is $\frac{TP}{TP+FP}$, and recall is $\frac{TP}{TP+FN}$; neither considers true negatives (TN). As shown in Table 4, the increment of PSL model over baselines on F-measure is not as large as the increase in accuracy. Comparing PSL2 and PSL3 to PSL1, the inference rules largely increase recall but lower precision. However, the accuracy in Table 3 keeps growing. Thus, the biggest advantage of PSL models is to correctly rule out true negative eTargets. For the baselines, though the SVM baseline has higher precision, it eliminates so many eTarget candidates that the F-measure is not high.

ETarget Selection. To assess the methods for eTarget selection, we run PSL3 (the fullest PSL model) using each method in turn. The F-measures and accuracies are listed in Table 5. The F-measure of ET1 is slightly lower than the F-measures of ET2 and ET3, while the accuracy of

	Precision	Recall	F-measure
	PosPair		
All NP/VP	0.1481	0.4857	0.2270
SVM	0.3791	0.0870	0.1415
PSL1	0.2234	0.2687	0.2440
PSL2	0.1666	0.2738	0.2072
PSL3	0.1659	0.3523	0.2256
	NegPair		
All NP/VP	0.1824	0.6408	0.2840
SVM	0.3568	0.0761	0.1254
PSL1	0.2857	0.3872	0.3288
PSL2	0.2772	0.3883	0.3235
PSL3	0.2586	0.4529	0.3292

Table 4: F-measure comparing PSL models (ET3 used for all)

ET1 is much better than the accuracies of ET2 and ET3. Again, this is because PSL recognizes true negatives in the eTarget candidates. Since ET1 considers more eTarget candidates, ET1 gives PSL a greater opportunity to remove true negatives, leading to an overall increase in accuracy.

	PosPair		NegPair	
	F	Acc.	F	Acc.
ET1	0.2192	0.4963	0.3157	0.4461
ET2	0.2374	0.4433	0.3261	0.3969
ET3	0.2256	0.4315	0.3295	0.3892

Table 5: Comparison of eTarget selection methods (PSL3 used for all)

6 Conclusion

This work builds upon state-of-the-art span-based sentiment analysis systems to perform entity/event-level sentiment analysis covering both explicit and implicit sentiments expressed among entities and events in text. Probabilistic Soft Logic models incorporating explicit sentiments, inference rules and +/-effect event information are able to jointly disambiguate the ambiguities in the opinion frames and improve over baseline accuracies in recognizing entity/event-level sentiments.

Acknowledgements. This work was supported in part by DARPA-BAA-12-47 DEFT grant #12475008. We thank the anonymous reviewers for their helpful comments.

References

Stephen H Bach, Bert Huang, and Lise Getoor. 2013. Learning latent groups with hinge-loss markov random fields. In *Inferning: ICML Workshop on Interactions between Inference and Learning*.

Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2015. Hinge-loss markov random fields and probabilistic soft logic. arXiv:1505.04406 [cs.LG].

Islam Beltagy, Katrin Erk, and Raymond Mooney. 2014. Probabilistic soft logic for semantic textual similarity. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1210–1219, Baltimore, Maryland, June. Association for Computational Linguistics.

Yoonjung Choi and Janyce Wiebe. 2014. +/-effectwordnet: Sense-level lexicon acquisition for opinion inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1181–1191, Doha, Qatar, October. Association for Computational Linguistics.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Lingjia Deng and Janyce Wiebe. 2014. Sentiment propagation via implicature constraints. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 377–385, Gothenburg, Sweden, April. Association for Computational Linguistics.

Lingjia Deng and Janyce Wiebe. 2015. Mpqa 3.0: An entity/event-level sentiment corpus. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1323–1328, Denver, Colorado, May–June. Association for Computational Linguistics.

Lingjia Deng, Yoonjung Choi, and Janyce Wiebe. 2013. Benefactive/malefactive event and writer attitude annotation. In *ACL 2013 (short paper)*. Association for Computational Linguistics.

Lingjia Deng, Janyce Wiebe, and Yoonjung Choi. 2014. Joint inference and disambiguation of implicit sentiments via implicature constraints. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 79–88, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

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.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Bert Huang, Angelika Kimmig, Lise Getoor, and Jennifer Golbeck. 2013. A flexible framework for probabilistic models of social trust. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 265–273. Springer.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Ben London, Sameh Khamis, Stephen H. Bach, Bert Huang, Lise Getoor, and Larry Davis. 2013. Collective activity detection using hinge-loss Markov random fields. In *CVPR Workshop on Structured Prediction: Tractability, Learning and Inference*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Alex Memory, Angelika Kimmig, Stephen Bach, Louiqa Raschid, and Lise Getoor. 2012. Graph summarization in annotated data using probabilistic soft logic. In *Proceedings of the 8th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2012)*, volume 900, pages 75–86.

Maria Pontiki, Haris Papageorgiou, Dimitrios Galanis, Ion Androutsopoulos, John Pavlopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *The Semantic Web–ISWC 2013*, pages 542–557. Springer.

Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The life and death of discourse entities: Identifying singleton mentions. In *HLT-NAACL*, pages 627–633.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng,

and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Citeseer.

Veselin Stoyanov, Claire Cardie, and Janyce Wiebe. 2005. Multi-Perspective Question Answering using the OpQA corpus. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 923–930, Vancouver, Canada.

Ivan Titov and Ryan T McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, volume 8, pages 308–316. Citeseer.

Janyce Wiebe and Lingjia Deng. 2014. An account of opinion implicatures. *arXiv*, 1404.6491[cs.CL].

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005a. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

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

Theresa Wilson. 2007. *Fine-grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of private states*. Ph.D. thesis, Intelligent Systems Program, University of Pittsburgh.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *ACL (1)*, pages 1640–1649.

Bishan Yang and Claire Cardie. 2014. Context-aware learning for sentence-level sentiment analysis with posterior regularization. In *Proceedings of ACL*.

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.