Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

Recognizing Opinion Sources Based on a New Categorization of Opinion Types

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

Recognizing sources of opinions is an important task in sentiment analysis. Different from previous works which categorize an opinion according to whether the source is the writer or the source is a noun phrase, we propose a new categorization of opinions according to the role that the source plays. The source of a *participant opinion* is a participant in the event that triggers the opinion. On the contrary, the source of a *non-participant opinion* is not a participant. Based on this new categorization, we classify an opinion using phrase-level embeddings. A transductive learning method is used for the classifier since there is no existing annotated corpora of this new categorization. A joint prediction model of Probabilistic Soft Logic then recognizes the sources of the two types of opinions in a single model. The experiments have shown that our model improves recognizing sources of opinions over baselines and several state-of-the-art works.

1 Introduction

Opinions are ubiquitous in languages. An explicit opinion has three components: the source (whose opinion is it), the opinion expression indicating polarity (positive, negative or neutral), and the target (what is the opinion toward). This paper focuses on recognizing sources of explicit opinions that can be anchored to specific entities, including the writer of the document and the heads of noun phrases (NPs) in the sentence. Most of previous works in sentiment analysis focus on analyzing reviews such as amazon reviews or movie reviews [Liu, 2012; Socher *et al.*, 2013]. Those works assume that the opinion sources are the writers. However, in other genres such as blogs or editorials, the opinion sources are not always the writers. Consider the examples in Table 1.

In (Ex1), there is a negative opinion. The opinion expression is *criticized*. The source is *Tom*. The target is *the student*. In (Ex2), the source of the negative opinion *criticized* is still *Tom*, though it is according to Mary's unexpected thoughts. In (Ex3), there is a positive opinion. The opinion expression is *considerate*. The positive opinion is stated by *Mary*, and the source of it is Mary as well. In (Ex4), the positive opinion toward *Jack* is attributed to the writer (or the speaker). In (Ex5), there is a negative opinion. The opinion expression is *embezzled*. The source is the writer. The target is *He*.

As we can see, the source of an opinion can be the writer or an entity represented by a noun phrase in the sentence (e.g., Mary, Tom). Several previous works contribute to recognizing noun phrases as sources [Choi *et al.*, 2005; 2006; Wiegand and Klakow, 2010]. They use sequence labeling techniques to label phrases as sources or classify noun phrases in the sentence. A few recent works [Yang and Cardie, 2013; Johansson and Moschitti, 2013] develop binary classifiers to determine whether the source is the writer. If not, they use similar methods as previous works did to label noun phrases as sources. In short, previous works categorize opinions as the ones whose sources are the writers or the ones whose sources are noun phrases. According to the previous works, (Ex1), (Ex2) and (Ex3) are in the same category since the sources are noun phrases, while (Ex4) and (Ex5) are in the other category since the sources are the writer.

However, not all the noun phrase sources play the same role in terms of the opinions. The opinion expressions in (Ex1) and (Ex2) are events (i.e., actions) and the sources are the agents of the events. (*Tom* is the agent of the event *criticized* which represents a negative opinion.) However, though the opinion expression in (Ex5) is also an event (*embezzled*), the source is not the agent (*He*) but the writer. The opinion expressions in (Ex3) and (Ex4) are not events and the sources are not semantic roles of the opinion. Further, if we interpret (Ex3) as *Mary:"Jack is very considerate."*, (Ex3) is very similar to (Ex4) since the sources are the persons who state the opinions. Similarly, the source in (Ex5) is the writer who state the opinion instead of anyone participating in the *embezzled* event. Thus, the methods developed to recognize sources in (Ex3), (Ex4) and (Ex5) should be different from the methods developed to recognize sources in (Ex1) and (Ex2). However, previous works develop the same method to recognize sources in (Ex1), (Ex2) and (Ex3) since the sources are all noun phrases. They use semantic role labeling outputs as important features during the training. This may result in failing to find the source in (Ex3) because the source is not a semantic role of the opinion. This may also result in misclassifying *He* as the source in (Ex5) because *He* is the agent of the event. Different from previous works, we first classify (Ex1) and (Ex2) in one category, and classify (Ex3), (Ex4) and (Ex5) in the other category, and then recognize the sources. Specif-

	Sentence (Boldfaced Opinion Expressions)	Source	Agent	Previous Categories	Our Categories
(Ex1)	Tom **criticized** the student.	Tom	Tom	source is noun phrase	source is participant
(Ex2)	Mary didn't expect that Tom **criticized** the student.	Tom	Tom	source is noun phrase	source is participant
(Ex3)	Mary says that Jack is very **considerate**.	Mary	N/A	source is noun phrase	source is non-participant
(Ex4)	Jack is very **considerate**.	writer	N/A	source is writer	source is non-participant
(Ex5)	He **embezzled** the pension.	writer	He	source is writer	source is non-participant

Table 1: Examples of Opinions, Sources of opinions, Agents of opinions, and Categories of opinions

ically, we name opinions such as (Ex1) and (Ex2) as ***participant opinions*** because the sources are participants in the events that trigger the opinions. we name opinions such as (Ex3), (Ex4) and (Ex5) as ***non-participant opinions*** because the sources are non participants. For example, the source can be someone who states the opinions. To our best knowledge, this is the first paper to point out the differences between participant opinions and non-participant opinions in the field of fine-grained sentiment analysis, and to utilize such differences to improve recognizing sources.

In summary, this paper aims at recognizing sources of participant opinions and non-participant opinions where the sources can be the writer or noun phrases in the sentences. To illustrate our work, we first introduce our definitions of participant opinions and non-participant opinions (Section 3). Based on the definitions, we develop a transductive SVM binary classifier to judge whether an opinion is a participant opinion or a non-participant opinion (Section 4). We choose transductive learning because existing opinion-oriented corpora do not contain the gold standard information of whether an opinion has a participant source or a non-participant source. Based on the classification results and a set of automatically extracted source candidates, we use a Probabilistic Soft Logic model to jointly recognize the sources of the two types of opinions in one single model (Section 5). The experiments have shown that our model has achieved better performances in F-measure than state-of-the-art works (Section 6). Finally we give the conclusion (Section 7).

2 Related Work

Different from the works analyzing reviews that assume the sources are the writers [Liu, 2012; Socher *et al.*, 2013], Choi et al., [2005] use Conditional Random Field (CRF) to recognize which phrases are the sources of opinions. Later, Choi et al., [2006] use CRF to automatically extract both opinion expressions and opinion sources. A binary classifier is run to assign sources to opinions. Finally an Integer Linear Programming model is run to choose the best configuration of correspondences of opinions and sources. Wiegand and Klakow [2010] consider all the noun phrases in the sentence and train a binary SVM classifier to judge whether a noun phrase is the source of a given opinion expression. They develop new convolution kernels used in SVM which are able to identify meaningful fragments of sequences or trees. Later Wiegand and Klakow [2012] develop generalization features to improve cross-domain opinion source extractions. Different from all the aforementioned works, this paper focuses on both cases where the sources can be the writers or phrases.

A few previous works extract sources including both writers and phrases in the text [Yang and Cardie, 2013; Johansson and Moschitti, 2013]. They follow a procedure similar to that of Choi et al. [2006]. One of the differences from Choi et al. [2006] is that a binary classifier is run to predict when the source is not a noun phrase. By this classifier, the model tries to recognize writer sources. Though the state-of-the-art works take into account both writer and noun phrases as potential sources, they did not model the distinction between participant opinions from non-participant opinions.

The sources of some non-participant opinions are the people in the text who state the opinion such as (Ex3). Recognizing such sources is similar to speaker attribution in quotation analysis [Glass and Bangay, 2007; Elson and McKeown, 2010; O'Keefe *et al.*, 2012; Pareti *et al.*, 2013]. We did not employ the techniques for speaker attribution in this paper because the features used in speaker attribution are extra-sentences and even extra-paragraphs, while we focus on recognizing sources of opinions within the sentence in this paper.

3 Definitions of Two Types of Opinions

A participant opinion is an opinion attributed to someone who is a participant in the event that triggers the opinion. The opinion expression of a participant opinion is usually an event directly triggering opinions (e.g., criticize in (Ex1) and (Ex2)). The source of it (e.g., Tom in Ex(1) and Ex(2), *participant source*) is usually a noun phrase.

A non-participant opinion is an opinion attributed to someone who is not a participant of the opinion. The opinion expression of a non-participant opinion is usually a description of the target (e.g., considerate in (Ex3) and (Ex4)). The source (*non-participant source*) can be the writer of the document such as (Ex4) and (Ex5) (*writer source*). There is no span of the writer source in the text. A non-participant source can be an entity in the text such as (Ex3) (*nonParticipantNP source*). A nonParticipantNP source is usually a noun phrase.

One thing to point out is that in some opinion-oriented corpora such as MPQA [Wiebe *et al.*, 2005] a few opinions do not have explicit sources in the sentence. Consider *Insulting the Prophet is a violation to human rights*. The event *Insulting* does not have an explicit agent. The negative opinion triggered by the *insulting* event does not have an explicit source as well. The source may refer to anyone in the world. Now consider *Chavez' candidacy raised expectations*. Though it is annotated as a positive opinion, the sentence does not specify whose expectations are raised, neither does it specify who are positive about this. The goal in this paper is to anchor the opinion sources to specific entities (either the writer or heads of noun phrases in the sentence). But such implicit source cannot be anchored to a specific entity. Thus, we do not consider opinions whose sources are implicit. About 5 percent opinions have implicit sources in MPQA.

4 Classifying Two Types of Opinions

We develop a binary classifier to distinguish non-participant opinions from participant opinions. The features used in the classifier are given in Section 4.1. As stated in Section 1, a challenge of this classification is the lack of labeled data. Though fine-grained opinion annotated corpora such as MPQA provide the source annotations of opinions, the annotations do not contain labels specifying whether the source is a participant or not. We describe how to train the binary classifier using the limited resources in Section 4.2.

4.1 Features

We use embeddings of opinion expressions as features for the binary classifier. We did not use any linguistic feature such as Part-Of-Speech or N-gram. Compare (Ex1) and (Ex5) in Section 1. Both opinion expressions are events and they have agents. And both opinion expressions are the words (*criticized* in (Ex1) and *embezzled* in (Ex5)) which can be found in sentiment or connotation lexicons. But (Ex1) is a participant opinion while (Ex5) is a non-participant opinion. Rather, we want to capture the differences by the meanings of the opinions expressions. We follow the same method in [Socher *et al.*, 2011] to generate word-level and phrase-level embeddings, which were used to recognize paraphrases. It is promising to use such embeddings to represent the meanings of the phrases.

In [Socher *et al.*, 2011][1], an unfolding recursive autoencoder (*uRAE*) is used to learn the embeddings for nodes on the binary parse tree where each parent has two children. During the encoding, the parent vector p_1 is computed from the children vectors (c_1,c_2), recursively the parent vector p_2 is computed from the children vectors (c_3,p_1).

$p_1 = f(W_e[c_1; c_2] + b_e) \quad p_2 = f(W_e[c_3; p_1] + b_e)$

The W_e is the encoding matrix to learn. Recursively we encode all the non-terminal nodes on the binary parse tree. Then during the decoding process, the parent vector is decoded to the two children vectors via reconstruction:

$[c_3'; p_1'] = f(W_d[p_2] + b_d) \quad [c_1'; c_2'] = f(W_d[p_1'] + b_d)$

where W_d is the decoding matrix to learn. Similarly to encoding, the decoding is recursively conducted on each node.

For the node p_2 that spans from word c_1, c_2 to c_3, the Euclidean distance between the original input of the leaves and the reconstructed representations of leaves is:

$E(p_2) = \|[c_1'; c_2'; c_3'] - [c_1; c_2; c_3]\|^2$

By minimizing the sum of all the Euclidean distances on all the nodes, we can learn the encoding and decoding matrices.

Following [Socher *et al.*, 2011], we generate a 100-dimension embedding vector for each opinion expression. A binary classifier is trained to learn the weight of each dimension. In the next section we discuss how we obtain the training data and how we use the training data to learn the weights.

[1] Available at http://goo.gl/4vKQGu

4.2 Training

We select non-participant opinion instances and participant opinion instances from the current opinion annotations in MPQA. Since there is no such label corresponding to the distinction between the two types of opinions, we use heuristics to select non-participant opinion instances and participant opinion instances. A selected non-participant opinion instance should have a much higher confidence being a non-participant opinion than being a participant opinion. For non-participant opinion instances, we collect the opinion expressions whose sources are annotated as writer since we are sure such opinions are non-participant opinions. Similarly, a selected participant opinion instance should have a much higher confidence being a participant opinion than being a non-participant opinion. Based on our observation (e.g., (Ex1) and (Ex2)), a participant opinion is usually a predicate and its source is usually the subject (A0) of the predicate. For participant opinion instances, we collect the opinion expressions which are predicates and at the same time their sources are the subjects of the predicates. For the remaining opinion annotations, we treat them as unknown instances.

We use two different training methods to learn the weights. In the first method, we simply use the selected non-participant opinion and participant opinion instances. We train an SVM classifier [Vapnik, 2013; Joachims, 1999a] to learn the weights (*non-transductive SVM*). In the second method, we use all the instances including the unknown instances to train a transductive SVM [Joachims, 1999b] to learn the weights (*transductive SVM*)[2].

The transductive SVM uses the unlabeled data to adjust the boundary so that the hyperlane separates both labeled and unlabeled data in the training set. Note that, only unlabeled data in the **training** set are used to learn the weights. None of the testing data is observed during training. It adds slack variables (ξ_i and $\hat{\xi_u}$), which allow the model to trade-off between misclassifying labeled data (x_i, y_i) and excluding unlabeled data ($\hat{x_u}, \hat{y_u}$) [Joachims, 1999b].

$$\begin{array}{lll} \min\limits_{w,\hat{y_u},\xi_i,\hat{\xi_u}} & \frac{1}{2}||w||^2 + C\sum_i \xi_i + \hat{C}\sum_u \hat{\xi_u} & \\ \text{s.t.} & \forall i \quad y_i(wx_i + b) & \geq 1 - \xi_i \\ & \forall u \quad \hat{y_u}(w\hat{x_u} + b) & \geq 1 - \hat{\xi_u} \end{array}$$

5 Recognizing Sources of Two Types of Opinions

To recognize sources of the two types of opinions, we choose a joint model instead of a pipeline end-to-end system which may suffer from accumulated errors. Different from the joint models used in the previous works which extract both writer sources and noun phrase sources, we choose Probabilistic Soft Logic (PSL) [Broecheler *et al.*, 2010][3]. A PSL model is defined using a set of atoms to be grounded, and using a set of weighted if-then rules as constraints. We choose PSL because the constraints are expressed in first order logic rules which are flexible to create and intuitive to understand.

[2] Available at http://svmlight.joachims.org/

[3] Available at http://psl.umiacs.umd.edu/

(R1) Opinion(o) $\wedge$ NPW(s) $\wedge$ NonParticipantOpinion(o) $\wedge$ NonParticipantNP(o,s) $\rightarrow$ Source(o,s) **(R2)** Opinion(o) $\wedge$ NPW(s) $\wedge$ NonParticipantOpinion(o) $\wedge$ Writer(o,s) $\rightarrow$ Source(o,s)
(R3) Opinion(o) $\wedge$ NPW(s) $\wedge$ ParticipantOpinion(o) $\wedge$ CRF(o,s) $\rightarrow$ Source(o,s) **(R4)** Opinion(o) $\wedge$ NPW(s) $\wedge$ ParticipantOpinion(o) $\wedge$ SemanticAgent(o,s) $\rightarrow$ Source(o,s) **(R5)** Opinion(o) $\wedge$ NPW(s) $\wedge$ ParticipantOpinion(o) $\wedge$ SyntacticAgent(o,s) $\rightarrow$ Source(o,s)
(R6) NonParticipantOpinion(o) $\rightarrow \sim$ ParticipantOpinion(o) **(R7)** ParticipantOpinion(o) $\rightarrow \sim$ NonParticipantOpinion(o)

Table 2: Rules in PSL

For example, we define the atom *Source*(o,s) to represent the grounding that the source of the opinion o is s, where o is an opinion expression and s can be the writer or a noun phrase in the sentence. If o and s are constants, then *Source*(o,s) is a grounded atom. Each grounded atom is assigned a score by an individual system, PSL takes as input all the individual scores and the constraints defined by rules among atoms. In the final output, for example if the score of the grounded atom *Source*(criticized, Jack) is larger than zero, it means that PSL thinks *Jack* is the source of the opinion *criticized*, and the score *Source*(criticized, writer) being 0 represents that PSL thinks the writer is not the source of that opinion.

In this section, we first introduce the PSL model in Section 5.1. Then we introduce the atoms defined for recognizing sources in Section 5.2. Finally the rules used as constraints among atoms are introduced in Section 5.3.

5.1 Probabilistic Soft Logic

PSL [Broecheler *et al.*, 2010] 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 grounded 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. The weights can be learnt.

A key 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 [Klir and Yuan, 1995] 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:

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

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

5.2 Atoms

Two sets of variables are used in the PSL in this paper. The first set consists of opinion expressions, each of which is denoted o. The second set consists of sources, each of which is denoted s. Since this paper focuses on recognizing sources, each o is an opinion expression in the gold standard. Given an opinion expression o in the sentence, we automatically generate a set S_o consisting of different source candidates of o. Each $s_o \in S_o$ is either the writer or the head of an NP. Note that we filter out any s that may not be an entity's head. We require that an entity must meet at least one of the three criteria: (1) it is a named entity; (2) it is a pronoun; (3) it is an animate according to the lexicon [Ji and Lin, 2009]. Each entry of the lexicon consists of an NP, the frequency that NP is used as an animate (labeled as *who*) and the frequency that NP is used as a non-animate (labeled as *which*, *when*, or *where*). We consider an NP as an animate if the frequency of *who* is higher than the frequency of any other label.

First of all, we define three basic atoms.

(A1) Opinion(o): o is an opinion expression

(A2) NPW(s): entity s is a source (NP or writer)

(A3) Source(o,s): the source of opinion o is s

For an opinion o in the gold standard, we create an Opinion(o) and the score is 1.0. For each source candidate s that individual systems generate, we create an NPW(s) and the score is 1.0. The scores of Source(o,s) will be the outputs calculated by the joint model.

Next, we define two atoms to describe o.

(A4) NonParticipantOpinion(o): o is a non-participant opinion

(A5) ParticipantOpinion(o): o is a participant opinion

The classifier in Section 4 outputs a score of each o. If the output is larger than zero, we create a NonParticipantOpinion(o) and the score is the output. If the output is smaller than zero, we create a ParticipantOpinion(o) and the score is the absolute value of the output. The absolute scores outside the range [0,1] is set as 1.0.

Further, we define atoms representing how we automatically generate source candidates. All the atoms defined below represent the grounding that s is a source candidate of o and they are assigned with score 1.0.

(A6) NonParticipantNP(o,s): s is an NP head as non-participant

(A7) Writer(o,s): s is the writer

(A8) CRF(o,s): s is an NP head extracted by a CRF model
(A9) SemanticAgent(o,s): s is the semantic agent of o
(A10) SyntacticAgent(o,s): s is the syntactic agent of o

NonParticipantNP(o,s) is created if o is a clause and s is the NP head that dominates o on the constituency parse tree. Specifically, if o is a clause (e.g., its parent node is labeled as *SBAR* in the parse tree), we go up the parse tree from o till the root, and collect the heads of the noun phrases along the path.

Writer(o,*writer*) is created if no NonParticipantNP atom of o is created.

CRF(o,s) is created if a pre-trained Conditional Random Filed (CRF) model extracts s as the source of o. Previous experiments have shown that CRF is a strong model in extracting noun phrases as sources [Yang and Cardie, 2013; Johansson and Moschitti, 2013]. We expect a CRF model could recognize the participant sources. Note that if the output from CRF is an NP, we choose the head of it as s. The features used in the model are typical linguistic features used in the previous works [Yang and Cardie, 2013].

SemanticAgent(o,s) is created if o is a predicate and s is the head of the subject (A0) of the predicate extracted by a semantic role labeling tool. We use SENNA [Collobert *et al.*, 2011] as the semantic role labeling tool in this paper.

SyntacticAgent(o,s) is created if s is the nsubj of o according to the dependency parser. We add the syntactic agent to retain the recall if the CRF model or semantic role labeling tool misses any source of an opinion. We use Stanford's dependency parser in this paper [Manning *et al.*, 2014].

For an opinion o, the set S_o consists of all the source candidates. PSL assigns scores to each s_o in S_o. A subset $S_o^{\text{non-participant}}$ consists of s_o created for NonParticipantNP and Writer atoms. The other subset $S_o^{\text{participant}}$ consists of s_o created for CRF, SemanticAgent and SyntacticAgent atoms.

5.3 Rules

We define rules used as constraints in PSL to model the relations of atoms, shown in Table 2. In the top box, Rules (R1) and (R2) are defined to find the sources of non-participant opinions. For example, Rule (R2) can be explained as: if the opinion expression o is a non-participant opinion and s is a source candidate which is the writer, then we infer the source of o is s. In the middle box, Rules (R3), (R4) and (R5) are defined to find the sources of participant opinions. In the bottom box, Rules (R6) and (R7) are defined to ensure that the same opinion cannot be both non-participant opinion and participant opinion. As introduced in Section 5.1, each rule is associated with a weight, representing how important the rule is. The weights of Rules (R6) and (R7) are infinite because they are used has hard constraints. The other weights are learnt on the training set in the experiment.

6 Experiments and Results

Our experiments are conducted on MPQA 2.0[4] [Wiebe *et al.*, 2005], a widely used corpus for fine-grained opinion analysis. 135 documents are used as a development set and a different set of 400 documents are used for 10-fold cross-validation. Our gold standard opinion expressions are corresponding to the direct subjectivity annotations and expressive subjectivity annotations. Our gold standard sources are corresponding to the agent annotations. We filter out the opinions whose sources are annotated as *implicit* (as stated in Section 3) and filter out the opinions whose sources are outside the sentence. The set of opinion expressions are the gold standard annotations. There are 11,364 opinion expressions in the cross-validation set. 3,826 opinion sources (33.67%) are annotated as the writer, and the other 7,538 opinion sources (66.33%) are annotated as noun phrases.

In the cross-validation, we use the training set in each fold to train three components of the whole model: (1) the classifier to classify opinion expressions in Section 4; (2) the CRF to extract noun phrases as sources in Section 5.2; (3) the PSL to conduct joint prediction in Section 5. After training, the whole model extracts sources in the testing set in each fold.

For evaluation, similar to previous works [Yang and Cardie, 2013], we use precision (P), recall (R) and F-measure (F1) according to *overlap* and *exact* matching metrics. For both metrics, if the automatically extracted source is the writer and the gold standard annotation is also the writer, it is a correct hit. In other cases, according to exact metric, if the automatically extracted source is the semantic head of the gold standard annotation span, it is a correct hit. According to overlap metric, if the automatically extracted source is within the gold standard annotation span, it is a correct hit.

We have conducted three experiments. The first experiment discusses the performances of our model in recognizing sources, compared to baselines and state-of-the-art works. The second experiment discusses the contribution of transductive SVM in recognizing sources, compared to non-transductive SVM. The third experiment discusses the learnt weights in the trained PSL. Next we talk about the three experiments.

6.1 Performances of Recognizing Sources

We use two baseline methods. For each opinion o from the gold standard, the first baseline (S_o) uses the whole source candidate set S_o as described in Section 5.2. The second baseline ($S_o^{n/p}$) uses a subset of S_o based on the classifier output. If the classifier labels o as a non-participant opinion, then the second baseline chooses the subset $S_o^{\text{non-participant}}$ as described in Section 5.2. If the classifier labels o as a participant opinion, then the second baseline chooses the subset $S_o^{\text{participant}}$. For an opinion o, the outputs from the two baselines S_o and $S_o^{\text{n/p}}$ are sets of sources, which may contain more than one source. The third baseline ($1 \in S_o^{\text{n/p}}$) builds upon the second baseline. It chooses the writer source from $S_o^{\text{non-participant}}$ if o is classified as a non-participant opinion. And it chooses the source candidate extracted by the CRF model if o is classified as a participant opinion. The third baseline always outputs a single source for an opinion. It combines the classification result and current CRF model in a pipeline approach. Our full model (*Joint*) uses the output from PSL. For an o, we take the s that has the highest positive score of Source(o,s). The performances are shown in Table 3. A star in F-measures indicates statistical significance according to t-test ($p < 0.05$).

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

	exact			overlap		
	P	R	F1	P	R	F1
S_o	36.21	**71.57**	48.09	36.50	**72.14**	48.47
$S_o^{\text{n/p}}$	47.50	58.35	52.37	50.19	58.88	54.19
$1 \in S_o^{\text{n/p}}$	49.16	49.16	49.16	49.67	49.67	49.67
Joint	**67.74**	60.20	**63.75***	**68.33**	60.73	**64.31***

Table 3: Performances of Recognizing Sources

The first baseline S_o has the highest recall because it considers all the source candidates. The second baseline $S_o^{\text{n/p}}$ successfully removes some wrong candidates by improving the precision and F-measure over S_o, indicating that classifying opinions can help recognize sources. The third baseline $1 \in S_o^{\text{n/p}}$ has a higher precision but a sharp drop in recall. This is because a pipeline approach may rule out correct candidates. The full model *Joint* achieves the best performance. Note that, the performances using overlap metric is only slightly better than using exact metric. This indicates that when our model recognizes a NP head as the source, in most cases it is the semantic head of the gold standard annotation. Note that the recall of S_o is not 100%. The errors are cases where the heads of sources are not nouns or pronouns. For example, the head of the source span *Those signing the document* is *Those*, whose Part-Of-Speech label is DT.

In addition to the two baselines, we also choose three models from the state-of-the-art works for comparison. The state-of-the-art works were conducted on MPQA with 10-fold cross validations as well. Thus we compare to their reported numbers directly. The first model (*Pipeline*), which is a pipeline approach, [Yang and Cardie, 2013] uses CRF to extract opinion expressions, opinion sources and opinion targets. Then binary classifiers are used to link the extracted sources (including the writer) and targets to opinions. Based on the first model's result, the second model (*ILP*) [Yang and Cardie, 2013] uses ILP to optimize the results. The third model (*Re-Rank*) is very similar to the second model, except that the third model only extracts opinion expressions and opinion sources and it uses a re-ranker to optimize the result. Since the state-of-the-art models automatically extracted opinion expressions, for a better comparison we train a CRF as described in [Yang and Cardie, 2014] to extract opinion expressions as well. The training of CRF is also conducted on the training set in each fold. Our model (*Auto+Joint*) in this experiment is different from the *Joint* in Table 3 since *Auto+Joint* takes as input automatically extracted opinion expressions. Using the evaluation methods in [Yang and Cardie, 2013], we evaluate on the opinions that are correctly extracted by the model, which are a subset of all the opinions in the corpus. Re-Rank calculates the percentage of overlapping tokens if an automatically extract source span overlaps with the gold standard span. The performances are shown in Table 4.

Our model (Auto+Joint) has the highest F-measure. The Pipeline has the lowest F-measure, indicating that a joint approach is more appropriate for recognizing sources. The ILP is better than the Re-Rank, and ILP is slightly worse than our model. It optimizes both the opinion-source relation and the opinion-target relation. It is promising to use PSL to jointly optimize the extraction of sources and targets in the future.

Method	P	R	F1	metric
Auto+Joint	**66.95**	**60.29**	**63.45**	overlap
Pipeline	47.73	54.40	50.84	overlap
ILP	64.97	58.61	61.63	overlap
Re-Rank	53.20	55.10	54.20	percentage

Table 4: Comparisons to State-of-the-art Models

6.2 Contribution of Transductive SVM

In Section 4.2, we introduce two methods to train the classifier — non-transduction and transduction. The F-measure of Joint using the transductive SVM in Table 3 is 63.75%. When we use the classification results from a non-transductive SVM for the Joint method, the F-measure is 61.47%, which is worse and the difference is statistically significant via a t-test ($p < 0.05$). The F-measure of the baseline ($S_o^{\text{n/p}}$) using transductive SVM is also statistically better than a $S_o^{\text{n/p}}$ using non-transductive SVM (52.37% versus 50.59%). Our experiments have shown that using transduction to train the classifier is able to improve recognizing sources.

6.3 Discussion of Trained PSL

We train the PSL model to learn the weights of rules. The weights learnt in each fold follow the same trend. Since the initial scores for Atoms (A6)-(A10) are set to be 1.0, the learnt rule weights are good estimates for how important each source candidate is. (R2) has the highest weight, indicating that if there is a writer source candidate extracted, it is very likely the correct source of an opinion. (R4) has a slightly higher weight than (R3). Though in most cases the CRF candidate is the same as the semantic agent candidate, the model prefers the semantic agent candidate if the two are not the same. (R5) has the lowest weight, which is not surprising because syntactic agent is a weak candidate. Furthermore, we run a PSL model without learning the weights but assigning each rule with the same weight. The performances are slightly worse. This shows that the trained weights of rules help recognize correct sources.

7 Conclusion

This work improves recognizing sources of opinions based on a new categorization of opinions: non-participant opinion or participant opinion. A transductive SVM is built to classify an opinion utilizing existing limited resources. The categorization information is then utilized by a Probabilistic Soft Logic model to jointly recognize sources of the two types of opinions in a single model. The experiments have shown that the model based on this new categorization of opinions achieves better performances over baselines and several state-of-the-art works in recognizing sources.

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

[Beltagy *et al.*, 2014] Islam Beltagy, Katrin Erk, and Raymond Mooney. 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 2014. Association for Computational Linguistics.

[Broecheler *et al.*, 2010] Matthias Broecheler, Lilyana Mihalkova, and Lise Getoor. Probabilistic similarity logic. In *Uncertainty in Artificial Intelligence (UAI)*, 2010.

[Choi *et al.*, 2005] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics, 2005.

[Choi *et al.*, 2006] Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *EMNLP*, pages 431–439. Association for Computational Linguistics, 2006.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[Elson and McKeown, 2010] David K Elson and Kathleen McKeown. Automatic attribution of quoted speech in literary narrative. In *AAAI*. Citeseer, 2010.

[Glass and Bangay, 2007] Kevin Glass and Shaun Bangay. A naive salience-based method for speaker identification in fiction books. In *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA07)*, pages 1–6, 2007.

[Ji and Lin, 2009] Heng Ji and Dekang Lin. Gender and animacy knowledge discovery from web-scale n-grams for unsupervised person mention detection. In *PACLIC*, pages 220–229, 2009.

[Joachims, 1999a] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA, 1999.

[Joachims, 1999b] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.

[Johansson and Moschitti, 2013] Richard Johansson and Alessandro Moschitti. Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3):473–509, 2013.

[Klir and Yuan, 1995] George Klir and Bo Yuan. *Fuzzy sets and fuzzy logic*, volume 4. Prentice hall New Jersey, 1995.

[Liu, 2012] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[O'Keefe *et al.*, 2012] Tim O'Keefe, Silvia Pareti, James R Curran, Irena Koprinska, and Matthew Honnibal. A sequence labelling approach to quote attribution. In *EMNLP*, pages 790–799. Association for Computational Linguistics, 2012.

[Pareti *et al.*, 2013] Silvia Pareti, Timothy O'Keefe, Ioannis Konstas, James R Curran, and Irena Koprinska. Automatically detecting and attributing indirect quotations. In *EMNLP*, pages 989–999, 2013.

[Socher *et al.*, 2011] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809, 2011.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642. Citeseer, 2013.

[Vapnik, 2013] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

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

[Wiegand and Klakow, 2010] Michael Wiegand and Dietrich Klakow. Convolution kernels for opinion holder extraction. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 795–803. Association for Computational Linguistics, 2010.

[Wiegand and Klakow, 2012] Michael Wiegand and Dietrich Klakow. Generalization methods for in-domain and cross-domain opinion holder extraction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 325–335. Association for Computational Linguistics, 2012.

[Yang and Cardie, 2013] Bishan Yang and Claire Cardie. Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[Yang and Cardie, 2014] Bishan Yang and Claire Cardie. Joint modeling of opinion expression extraction and attribute classification. *Transactions of the Association for Computational Linguistics*, 2:505–516, 2014.