Supervised and Unsupervised Methods in Employing Discourse Relations for Improving Opinion Polarity Classification

Swapna Somasundaran, Galileo Namata, Janyce Widbe, Lise Getoor
Motivation

**Goal:** Opinion Polarity Classification

**Motivation:**

- Need to enable discourse-base opinion analysis
- Previous research developed discourse schemes and created manually annotated corpora
- People want to know how well they can be translated into computational models
Framework

Discourse scheme:

* A discourse level opinion interpretation

Computational Models:

* How the discourse scheme can be utilized to improve polarity classification.
  * Supervised framework
    * Iterative Collective Classification (ICA)
  * Unsupervised framework
    * Integer Linear Programming (ILP)

* Experiment results show both of them achieve significantly better accuracies in polarity classification than classifiers using local information only
Method

Data and Discourse scheme:

- Data: AMI meeting corpus (Carletta et al., 2005)

- Discourse scheme: Somasundaran et al. (2008)
  - Annotate Opinions
    - Polarities:
      - positive, negative, neutral
    - Targets relations:
      - same, alternative
    - Frame relations:
      - reinforcing, non-reinforcing
Method

Data and Discourse scheme:

• Discourse scheme:
  
  • e.g.

  DA-1: ... this kind of rubbery material,
  DA-2: it’s a bit more bouncy,
  DA-3: like you said they get chucked around a lot.
  DA-4: A bit more durable and that can also be ergonomic and
  DA-5: it kind of feels a bit different from all the other remote controls.

* The individual opinion expressions shown in bold
* DA segments is provided with AMI corpus

Figure 1. Discourse Relations between DA segments for example
Method

Computational Models:

- Local classifier
  - Motivation: global discourse view will improve upon a classification with only a local view
  - It is a supervised classifier
    - SVM is used in the experiment
- Features:
  - Polarity lexicon
  - DA tags
  - Unigrams
Method

Computational Models:

• Supervised framework – Iterative Collective Classification (ICA)
  
  • Idea: Iteratively improve classification result using additional discourse features.

  • Two classifiers:
    ▶ A local classifier using local features only

    ▶ A relational classifier combining local features and relational features
Method

Computational Models:

- Supervised framework – Iterative Collective Classification (ICA)

```plaintext
Algorithm 1 ICA Algorithm

for each instance i do {bootstrapping}
    Compute polarity for i using local attributes
end for
repeat {iterative}
    Generate ordering I over all instances
    for each i in I do
        Compute polarity for i using local and relational attributes
    end for
until Stopping criterion is met
```

Two classifiers

Local classifier (introduced before)

Relational classifier
Method

Computational Models:

- Supervised framework – Iterative Collective Classification (ICA)
- Relational classifier (SVM in the experiment)
  - Local features plus relational features (listed below, total 59 relational features)

| Percent of neighbors with polarity type $a$ related via frame relation $f'$ |
| Percent of neighbors with polarity type $a$ related via target relation $t'$ |
| Percent of neighbors with polarity type $a$ related via frame relation $f$ and target relation $t$ |
| Percent of neighbors with polarity type $a$ and same speaker related via frame relation $f'$ |
| Percent of neighbors with polarity type $a$ and same speaker related via target relation $t'$ |
| Percent of neighbors with polarity type $a$ related via a frame relation or target relation |
| Percent of neighbors with polarity type $a$ related via a reinforcing frame relation or $same$ target relation |
| Percent of neighbors with polarity type $a$ related via a non-reinforcing frame relation or alt target relation |
| Most common polarity type of neighbors related via a $same$ target relation |
| Most common polarity type of neighbors related via a reinforcing frame relation and $same$ target relation |

Table 1: Relational features: $a \in \{\text{non-neutral (i.e., positive or negative), positive, negative}\}$, $t \in \{\text{same, alt}\}$, $f \in \{\text{reinforcing, non-reinforcing}\}$, $t' \in \{\text{same or alt, same, alt}\}$, $f' \in \{\text{reinforcing or non-reinforcing, reinforcing, non-reinforcing}\}$
Method

Computational Models:

- Unsupervised framework – Integer Linear Programming (ILP)
  - Linear programs (LPs) are problems that can be expressed in canonical form:
    \[
    \begin{align*}
    &\text{maximize} \quad c^T x \\
    &\text{subject to} \quad Ax \leq b \\
    &\text{and} \quad x \geq 0
    \end{align*}
    \]
    where \( x \) represents the vector of variables (to be determined), \( c \) and \( b \) are vectors of (known) coefficients, \( A \) is a (known) matrix of coefficients.

- **ILP** is the name given to LP problems which have the additional constraint that some or all the variables have to be integer:
  \[
  \begin{align*}
  &\text{maximize} \quad c^T x \\
  &\text{subject to} \quad Ax \leq b, \\
  &\quad x \geq 0, \\
  &\text{and} \quad x \text{ integer},
  \end{align*}
  \]
Method

Computational Models:

- Unsupervised framework – Integer Linear Programming (ILP)
- Discourse Constraint on Polarity
  - Idea: The discourse relations between opinions can provide coherence constraint on the way their polarity is interpreted.

<table>
<thead>
<tr>
<th>Target relation + Frame relation</th>
<th>Polarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>same + reinforcing</td>
<td>equal (e)</td>
</tr>
<tr>
<td>same + non-reinforcing</td>
<td>opposite (o)</td>
</tr>
<tr>
<td>alternative + reinforcing</td>
<td>opposite (o)</td>
</tr>
<tr>
<td>alternative + non-reinforcing</td>
<td>equal (e)</td>
</tr>
</tbody>
</table>

Table 2. Discourse relations and their polarity constraints on the related instances.

- e.g. “shapes should be curved, nothing square-like”
Method

Computational Models:

• Unsupervised framework – Integer Linear Programming (ILP)

  Optimization Problem:

  • Minimize $-1 \times \sum_i (p_i x_i + q_i y_i + r_i z_i) + \sum_{i,j} \epsilon_{ij} + \sum_{i,j} \delta_{ij}$

  where the $x_i$, $y_i$, and $z_i$ are binary class variables corresponding to positive, negative and neutral classes, respectively. When a class variable is 1, the corresponding class is chosen. Variables $\epsilon_{ij}$ and $\delta_{ij}$ are binary slack variables that correspond to the discourse constraints between two distinct DA instances $i$ and $j$. When a given slack variable is 1, the corresponding discourse constraint is violated.

  $\sum_i (p_i x_i + q_i y_i + r_i z_i) :$ Maximize the probability provided by local classifier

  $\sum_{i,j} \epsilon_{ij} + \sum_{i,j} \delta_{ij} :$ Minimize the number of discourse constraints violate
Method

Computational Models:

- Unsupervised framework – Integer Linear Programming (ILP)
- Basic Constraints:

\[
x_i \in \{0, 1\}, y_i \in \{0, 1\}, z_i \in \{0, 1\}, \ \forall i
\]
\[
\epsilon_{ij} \in \{0, 1\}, \delta_{ij} \in \{0, 1\}, \ \forall i \neq j
\]
\[
x_i + y_i + z_i = 1, \ \forall i
\]
Method

Computational Models:

- Unsupervised framework – Integer Linear Programming (ILP)

  - equal-polarity Constraints:
    
    \[ |x_i - x_j| \leq 1 - e_{ij} + \epsilon_{ij}, \ \forall i \neq j \]
    
    \[ |y_i - y_j| \leq 1 - e_{ij} + \epsilon_{ij}, \ \forall i \neq j \]
    
    \[-(x_i + y_i) \leq -l_i, \ \forall i \]

  - Opposite-polarity Constraints:
    
    \[ |x_i + x_j - 1| \leq 1 - o_{ij} + \delta_{ij}, \ \forall i \neq j \]
    
    \[ |y_i + y_j - 1| \leq 1 - o_{ij} + \delta_{ij}, \ \forall i \neq j \]

  \( e_{ij} = 1 \) if DA pair \( ij \) is same+reinforcing or alternative+non-reinforcing, 0 otherwise
  
  \( o_{ij} = 1 \) if DA pair \( ij \) is same+non-reinforcing or alternative+reinforcing, 0 otherwise
  
  \( l_i = 1 \) if the instance \( I \) participates in one or more discourse relations
Experiments

Test Dataset:

Test performance

• Over instances related via discourse relations (Connected)

• Over instances not related via discourse relations (Singletons)

• All: Both of connected and singletons

Filtered dataset:

<table>
<thead>
<tr>
<th></th>
<th>Pos</th>
<th>Neg</th>
<th>Neutral</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected</td>
<td>643</td>
<td>343</td>
<td>81</td>
<td>1067</td>
</tr>
<tr>
<td>Singleton</td>
<td>553</td>
<td>233</td>
<td>2753</td>
<td>3539</td>
</tr>
<tr>
<td>All</td>
<td>1196</td>
<td>576</td>
<td>2834</td>
<td>4606</td>
</tr>
</tbody>
</table>

Table 3: Class distribution over connected, single and all instances.
Experiments

Classifiers:

- Base: Distribution-based classifier
- Base-2: Use separate distribution-based classifiers for each dataset
- Local classifier: SVM
- Hybrid classifier (HYB) which selects the ICA as classifier for singletons and ILP as classifier for connected instances

Results: (7-fold cross validation)

<table>
<thead>
<tr>
<th></th>
<th>Base</th>
<th>Base-2</th>
<th>Local</th>
<th>ICA</th>
<th>ILP</th>
<th>HYB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected</td>
<td>24.4</td>
<td>47.56</td>
<td>46.66</td>
<td><strong>55.64</strong></td>
<td>75.07</td>
<td>75.07</td>
</tr>
<tr>
<td>Singleton</td>
<td>51.72</td>
<td>63.23</td>
<td>75.73</td>
<td><strong>78.72</strong></td>
<td>75.73</td>
<td>78.72</td>
</tr>
<tr>
<td>All</td>
<td>45.34</td>
<td>59.46</td>
<td>68.72</td>
<td><strong>73.31</strong></td>
<td>75.35</td>
<td>77.72</td>
</tr>
</tbody>
</table>

Table 4: Accuracies of the classifiers measured over Connected, Singleton and All instances. Performance significantly better than Local are indicated in bold for $p < 0.001$ and underline for $p < 0.01$. 
Experiments

More Results:

- Compare ILP and Local classifier
  - ILP leads in Connected-Recall 40%(pos) and 29%(neg)
  - ILP leads in Connected-F1 24%(pos) and 17%(neg)
- Conclusion: ILP has an overall improvement in precision over local

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local</td>
<td>ILP</td>
<td>Local</td>
</tr>
<tr>
<td>Connected-Prec</td>
<td>78.1</td>
<td>78.2</td>
<td>71.9</td>
</tr>
<tr>
<td>Connected-Recall</td>
<td>45.3</td>
<td><strong>86.3</strong></td>
<td>44.1</td>
</tr>
<tr>
<td>Connected-F1</td>
<td>56.8</td>
<td><strong>81.5</strong></td>
<td>54.0</td>
</tr>
<tr>
<td>All-Prec</td>
<td>56.2</td>
<td><strong>61.3</strong></td>
<td>52.3</td>
</tr>
<tr>
<td>All-Recall</td>
<td>46.6</td>
<td><strong>67.7</strong></td>
<td>44.3</td>
</tr>
<tr>
<td>All-F1</td>
<td>50.4</td>
<td><strong>64.0</strong></td>
<td>46.0</td>
</tr>
</tbody>
</table>

Table 5: Precision, Recall, F-measure for each Polarity category. Performance significantly better than Local are indicated in **bold** (p < 0.001), **underline** (p < 0.01) and **italics** (p < 0.05). The * denotes that ILP does not retrieve any connected node as neutral.
Experiments

More Results:

<table>
<thead>
<tr>
<th>Gold</th>
<th>Local</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pos</td>
<td>Neg</td>
<td>Neut</td>
<td>Total</td>
</tr>
<tr>
<td>Pos</td>
<td>551</td>
<td>113</td>
<td>532</td>
<td>1196</td>
</tr>
<tr>
<td>Neg</td>
<td>121</td>
<td>250</td>
<td>205</td>
<td>576</td>
</tr>
<tr>
<td>Neut</td>
<td>312</td>
<td>135</td>
<td>2387</td>
<td>2834</td>
</tr>
<tr>
<td>Total</td>
<td>984</td>
<td>498</td>
<td>3124</td>
<td>4606</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gold</th>
<th>ILP</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pos</td>
<td>Neg</td>
<td>Neut</td>
<td>Total</td>
</tr>
<tr>
<td>Pos</td>
<td>817</td>
<td>157</td>
<td>222</td>
<td>1196</td>
</tr>
<tr>
<td>Neg</td>
<td>147</td>
<td>358</td>
<td>71</td>
<td>576</td>
</tr>
<tr>
<td>Neut</td>
<td>358</td>
<td>147</td>
<td>2329</td>
<td>2834</td>
</tr>
<tr>
<td>Total</td>
<td>1322</td>
<td>662</td>
<td>2622</td>
<td>4606</td>
</tr>
</tbody>
</table>

Table 6: Contingency table over instances

- Even though ILP makes more polar guesses as compared to Local, a greater proportion of the ILP guesses are correct.
- The number of non-diagonal elements are much smaller for ILP
Experiments

Example:

DA-1: ... this kind of rubbery material,
DA-2: it’s a **bit more bouncy**, 
DA-3: like you said they get chucked around a lot.
DA-4: A **bit more durable** and that can also be **ergonomic** and
DA-5: it kind of feels **a bit different from all the other remote controls**.

Classifying Result of example 1

In the example above, only DA-4 is correctly predicated by local classifier. However
ILP is able to fix the false predication by local classifier by using discourse context.
Experiments

More Example:

- e.g. 2

D-1: I reckon you’re **gonna have to have** a number keypad anyway for the amount of channels these days,
D-2: You **wouldn’t want to** just have to scroll through all the channels to get to the one you want
D-3: You **wanna** enter just the number of it, if you know it
D-4: I reckon **we’re gonna have to have** a number keypad anyway

- Classifying Result of example 2