# A Pilot Study on Argument Simplification in Stance-Based Opinions

Pavithra Rajendran, Danushka Bollegala and Simon Parsons

October 6, 2019

# Introduction

- Argument mining is a relatively new field combining concepts drawn from natural language processing and computational argumentation to extract arguments and their relations from social media texts.

# Problem Statement

- Opinionated texts (e.g. online reviews) contain a lot of information in which stance is expressed implicitly.
- In prior work, we extracted opinions from a set of hotel reviews and manually annotated as explicit or implicit based on how the stance in the opinion is expressed.
  - Here, stance is derived from linguistics and is defined as *expression of judgment, attitude in the content towards the standpoint taken in the message.*
- **Given a set of opinions, does classifying the opinions into explicit and implicit opinions help to identify an explicit opinion as a simplified argument for an implicit opinion?**

# Explicit/Implicit Opinions: Examples

| **Explicit opinions** |
| --- |
| "worst **hotel** ever!!!" |
| "just spent 3 nights at this hotel 5th march 04 -8th march 04. the **location** is excellent and the **hotel** is very grand. " |
| "the **prices** are very high, even for a 5 star hotel." |
| "not the **service** we expected " |
| "**Parking** was expensive at $35 per night (2003)." |

| **Implicit opinions** |
| --- |
| "during the rest of my stay i also noted peeling wallpaper in some areas and in others the walls were covered with pencil scribbles - the **room** was better than the first but was still pretty tired looking." |
| "the **bathroom** is small and outdated." |
| "Paying this sort of money, I expected, rightly or wrongly so, to have some sort of standard of **service**" |
| "Upon our return we were told a table was not ready and that we should go up to the bar and they would let us know when a table was ready" (*aspect 'service' is implicitly implied*) |
| "initially, a new **receptionist** mistakenly gave us a smoking room but the very capable and pleasant assistant general **manager** laura rectified this problem the next day." |

Some examples of explicit and implicit opinions. Bold text represents the aspect(s) present in the opinions.

# Argument simplification: Examples

Some examples of implicit opinions and their corresponding explicit opinions as simplified arguments:

| Implicit opinion | Explicit opinion |
| --- | --- |
| rooms had plenty of room and nice and quiet (no noise from the hallway hardwood floors as suggested by some - all carpeted) | room was great |
| we received a lukewarm welcome at check in (early evening) and a very weak offer of help with parking and our luggage | we were extremely unimpressed by the quality of service we encountered |
| i have been meaning to write a review on this hotel because of the fact that staying here made me dislike Barcelona (hotels really can affect your overall view of a place, unfortunately) | this hotel was just a great disappointment |

# Proposed Approach

- The argument simplification problem is formulated as a maximum cost $K$ ranked bipartite-graph matching problem using a set of explicit and implicit opinions.
- For every implicit opinion, the top $K$ explicit opinions with the highest cost are considered. The cost function is computed using the three different features as follows:

$$\mathrm{C}(i,j) = \mathrm{sim}(\boldsymbol{s}_\mathbf{i}, \boldsymbol{s}_\mathbf{j}) + \mathrm{Q}(i,j) + \mathrm{R}(i,j) \tag{1}$$

where:

- $\mathrm{sim}$ represents the similarity measure computed between two sentence embedding vectors $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$.
- $\mathrm{Q}$ represents the cost value by checking whether sentiment of the two sentences are same or not.
- $\mathrm{R}$ represents the cost value by checking whether target present in the two sentences are the same or not.

# Different sentence embedding representations

- Each word is initialised with pre-trained embedding vectors.
- Existing works by Arora et al. (2016) and Mu et al. (2017) are used to perform different steps on the initialised word embeddings to create sentence embedding vectors.
- Two post-processing steps are performed by Mu et al. (2017) on pre-trained word embedding vectors. The motivation of their work is to create better word embedding representations and hence do not focus on sentence representation.

**Diff** Let us assume that we are given a set $\mathcal{V}$ (vocabulary) of words $w$, which are represented by a pre-trained word embedding $\boldsymbol{w}_i \in \mathbb{R}^k$ in some $k$ dimensional vector space. The mean embedding vector, $\hat{\boldsymbol{w}}$, of all embeddings for the words in $\mathcal{V}$ is given by:

$$\hat{\boldsymbol{w}} = \frac{1}{|\mathcal{V}|} \sum_{w \in \mathcal{V}} \boldsymbol{w} \tag{2}$$

Using the steps in Mu et al. (2017), the mean is subtracted from each word embedding to create isotropic embeddings as follows:

$$\forall_{w \in \mathcal{V}} \quad \tilde{\boldsymbol{w}} = \boldsymbol{w} - \hat{\boldsymbol{w}} \tag{3}$$

**WordPCA** The mean-subtracted word embeddings given by (3) for all $w \in \mathcal{V}$ are arranged as columns in a matrix $\mathbf{A} \in \mathbb{R}^{k \times |\mathcal{V}|}$, and its $d$ principle component vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d$ are computed. Mu et al. (2017) observed that the normalised variance ratio decays until some top $l \leq d$ components, and remains constant after that, and proposed to remove the top $l$ principle components from the mean-subtracted embeddings as follows:

$$\boldsymbol{w}' = \tilde{\boldsymbol{w}} - \sum_{i=1}^{l} (\boldsymbol{u}_i \boldsymbol{w})\, \boldsymbol{u}_i \tag{4}$$

# Different sentence embedding representations

**AVG** One of the simplest, yet surprisingly accurate, method to represent a sentence is to compute the average of the embedding vectors of the words present in that sentence. Given a sentence $\mathcal{S}$, we first represent it using the set of words $\{w | w \in \mathcal{S}\}$. We then create its sentence embedding $\boldsymbol{s} \in \mathbb{R}^k$ as follows:

$$\boldsymbol{s} = \frac{1}{|\mathcal{S}|} \sum_{w \in \mathcal{S}} \boldsymbol{w} \tag{5}$$

Three different variants for sentence embeddings are possible depending on the pre-processing applied on the word embeddings used in (5): **AVG** (uses unprocessed word embeddings $\boldsymbol{w}$), **Diff+AVG** (uses $\tilde{\boldsymbol{w}}$) and **WordPCA+AVG** (uses $\boldsymbol{w}'$).

**WEmbed** Arora et al. (2016) Sentence embeddings as the weighted-average of the word embeddings for the words in a sentence. The weight $\psi(w)$ of a word $w$ is computed using its occurrence probability $p(w)$ estimated from a corpus as follows:

$$\psi(w) = \frac{a}{a + p(w)} \boldsymbol{w} \tag{6}$$

$$\boldsymbol{s} = \frac{1}{|\mathcal{S}|} \sum_{w \in \mathcal{S}} \psi(w) \boldsymbol{w} \tag{7}$$

**SentPCA** Given a set of sentences $\mathcal{T}$, apply PCA on the matrix that contains individual sentence embeddings as columns to compute the first principle component vector $\boldsymbol{v}$, which is subtracted from each sentence's embedding as follows:

$$\boldsymbol{s}' = \boldsymbol{s} - \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} \boldsymbol{s} \tag{8}$$

# Similarity score: Unsupervised approach

- Cosine similarity score between two sentence embeddings.
- Sentence embedding vectors computed as described in previous section.

# Similarity score: Supervised approach

- A pair of sentences is represented using two operators: $\boldsymbol{h}_{\times}$ and $\boldsymbol{h}_{-}$ and sentences are initialized using sentence embedding vectors (described in previous section).
- A neural network containing a sigmoid ($\sigma(\cdot)$) hidden layer and a softmax ($\phi(\cdot)$) output layer parametrised by a set $\theta = \{\mathbf{W}^{(\times)}, \mathbf{W}^{(-)}, \mathbf{W}^{(p)}, \boldsymbol{b}^{(h)}, \boldsymbol{b}^{(p)}\}$ as follows:

$$\boldsymbol{h}_{\times} = \boldsymbol{s}_i \odot \boldsymbol{s}_j$$

$$\boldsymbol{h}_s = \sigma\left(\mathbf{W}^{\times}\boldsymbol{h}_{\times} + \mathbf{W}^{(-)}\boldsymbol{h}_{-} + \boldsymbol{b}^{(h)}\right)$$

$$\boldsymbol{h}_{-} = |\boldsymbol{s}_i - \boldsymbol{s}_j|$$

$$\hat{\boldsymbol{p}}_{\theta} = \phi\left(\mathbf{W}^{(p)}\boldsymbol{h}_s + \boldsymbol{b}^{(p)}\right)$$

- Parameters $\theta$ of the model are found by minimising the KL-divergence between $\boldsymbol{p}$ and $\hat{\boldsymbol{p}}_{\theta}$ subjected to $\ell_2$ regularisation over the entire training dataset $\mathrm{D}$ of sentence pairs as follows:

$$J(\theta) = \sum_{(s_i, s_j) \in \mathrm{D}} \mathrm{KL}\left((p^{(k)} || \hat{p}_{\theta}^{(k)}\right) + \frac{\lambda}{2} ||\theta||_2^2 \qquad (9)$$

Here, $\lambda \in \mathbb{R}$ is the regularisation coefficient, set using validation data.

# Sentiment and Target scores

- If two sentences have the same sentiment, a predefined score is set, else 0.0
- If two sentences talk about the same target/aspect, a predefined score is set, else 0.0

# Experiments

- Pre-trained Glove 300 dimensional word vectors are used.
- Sentiment of an opinion and the targets present are manually annotated and a domain knowledge base related to the different aspects and aspect categories is used.
- Threshold values for both the sentiment and target functions were set as 0.5 (varied from 0 to 1 on development data) such that the cost function is not biased towards the sentiment and target information alone.
- SICK similarity dataset is used as a training set for computing similarity score in a supervised approach.

# Experiments - Datasets

## Implicit/Explicit opinions dataset

- Randomly selected 57 implicit opinions from implicit/explicit opinions dataset and manually annotated with three most appropriated explicit opinions.
- The implicit/explicit opinions dataset contains 1288 opinions manually annotated by two annotators with an inter-annotator agreement with a Cohen's kappa of 0.71.

## Citizen Dialogue corpus

- We also collected 64 argument pairs with rephrase relation from the Citizen Dialogue corpus for our experiments and manually annotated arguments and their corresponding simplified arguments.
- Example: *We're going to keep you informed* is a simplified argument representation of *During this construction phase, we're going to be doing everything we can to keep you informed and keep you safe and keep traffic moving safely.*.

# Evaluation measures

- Precision@K
- Averaged precision@K (Avg P@K)
- Mean Reciprocal Rank (MRR)
- Accuracy

# Results

For a given set 57 implicit opinions and 56 explicit opinions, we compute the cosine similarity between each pair of implicit and explicit opinions using each of the methods described and results are shown below.

| **Methods** | **P@10** | **P@15** | **P@20** | **Avg P@15** | **Avg P@20** |
|---|---|---|---|---|---|
| **UNSUPERVISED** | | | | | |
| AVG | 0.15 | 0.22 | 0.30 | 0.13 | 0.16 |
| Diff+AVG | 0.15 | 0.21 | 0.27 | 0.12 | 0.15 |
| WordPCA+AVG | **0.17** | **0.23** | **0.30** | **0.14** | **0.17** |
| WEmbed | 0.14 | 0.20 | 0.25 | 0.12 | 0.15 |
| SENTPCA | 0.14 | 0.20 | 0.27 | 0.12 | 0.21 |
| **SUPERVISED** | | | | | |
| AVG | 0.14 | 0.19 | 0.25 | 0.12 | 0.15 |
| Diff+AVG | 0.14 | 0.19 | 0.24 | 0.11 | 0.14 |
| WordPCA+AVG | 0.14 | 0.21 | 0.25 | 0.12 | 0.15 |
| WEmbed | 0.07 | 0.12 | 0.18 | 0.05 | 0.08 |
| SENTPCA | 0.10 | 0.14 | 0.22 | 0.08 | 0.11 |
| Sentiment | 0.08 | 0.14 | 0.17 | 0.06 | 0.13 |
| Target | 0.16 | 0.20 | 0.24 | 0.12 | 0.19 |
| Sentiment + target | 0.17 | 0.22 | 0.25 | 0.13 | 0.20 |
| WordPCA+AVG+sentiment+target | **0.28** | **0.34** | **0.39** | **0.21** | **0.26** |

# Results

The results below are reported based on the following: the information whether an opinion is implicit/explicit for the implicit/explicit dataset and the category to which an argument belongs to for the Citizen Dialogue corpus is given (With Information) or not given (Without Information).

| **Methods** | Without Information | | | | With Information | | | |
|---|---|---|---|---|---|---|---|---|
| | Citizen Dialogue | | Implicit/Explicit | | Citizen Dialogue | | Implicit/Explicit | |
| | **MRR** | **Acc** | **MRR** | **Acc** | **MRR** | **Acc** | **MRR** | **Acc** |
| **UNSUPERVISED** | | | | | | | | |
| AVG | 0.56 | 0.75 | 0.13 | 0.31 | 0.62 | 0.81 | 0.29 | 0.75 |
| Diff+AVG | 0.55 | 0.75 | 0.12 | 0.28 | 0.61 | 0.81 | 0.28 | 0.75 |
| WordPCA+AVG | 0.59 | 0.80 | 0.07 | 0.24 | 0.64 | 0.86 | 0.25 | 0.82 |
| WEmbed | 0.52 | 0.67 | 0.15 | 0.49 | 0.55 | 0.72 | 0.32 | 0.68 |
| SENTPCA | 0.51 | 0.67 | 0.16 | 0.47 | 0.55 | 0.72 | 0.35 | 0.65 |
| **SUPERVISED** | | | | | | | | |
| AVG | 0.56 | 0.78 | 0.10 | 0.31 | 0.63 | 0.83 | 0.27 | 0.68 |
| Diff+AVG | 0.54 | 0.78 | 0.10 | 0.30 | 0.61 | 0.83 | 0.25 | 0.68 |
| WordPCA+AVG | 0.57 | 0.76 | 0.06 | 0.24 | 0.63 | 0.80 | 0.26 | 0.74 |
| WEmbed | 0.004 | 0.03 | 0.08 | 0.23 | 0.04 | 0.16 | 0.23 | 0.70 |
| SENTPCA | 0.007 | 0.04 | 0.10 | 0.31 | 0.03 | 0.16 | 0.13 | 0.35 |

# Results

- **SENTPCA** does not perform better than the simple baseline **AVG**.
- **WordPCA+AVG** is the best sentence embedding representation useful for predicting the correct explicit opinions.
- Similarity scores obtained using unsupervised sentence embedding representation do better than the sentiment and target functions, and we get the best performance using all three types of features.

# Analysis of results

- In some cases, sentiment and target are not able to predict the answers correctly while in other cases, the similarity measure fails to capture the information that is explicitly provided by sentiment and target.
- An example where sentiment and target fails:

  Implicit opinion *"but the service is totally different with so many rooms for improvement it became not acceptable"*

  Simplified argument/Explicit opinion (similarity score, target and sentiment) *"we were extremely unimpressed by the quality of service we encountered"* (Correct)

  Simplified argument/Explicit opinion (target and sentiment) *"the rooms are not worth the money"* (Incorrect)

# Analysis of results

- An example where similarity score fails to capture:

  Implicit opinion *"the laundry came back promptly"*
  Simplified argument/Explicit opinion *"the service was great"*

- Reason 1: Sentences are quite short, and many of the words they contain — "came", "was", "back" and so on — are common words that are not good features for opinion matching.
- Reason 2: It is also possible that the embeddings of the words "laundry" and "service" were not available or were not present as close word pairs.
- As future work, we will investigate on this.

# Conclusion

- Unsupervised bipartite graph-based approach to automatically predict simplified arguments.
- Experimental results on two different datasets show that unsupervised sentence representations help in matching arguments with their corresponding simplified arguments.
- The implicit/explicit opinion classification improves the performance for predicting the relation among opinions.
- Weighted-averaged sentence embeddings, useful for similarity tasks, do not give the best performance. The best performance is achieved when sentences are represented using averaged word vectors, where the word vectors are post-processed using **WordPCA**.

Thank you!