A Pilot Study on Argument Simplification in Stance-based Opinions

Pavithra Rajendran[1], Danushka Bollegala[1], and Simon Parsons[2]

[1] University of Liverpool, United Kingdom
pavithra.rajendran@liverpool.ac.uk, danushka.bollegala@liverpool.ac.uk
[2] Kings College London, United Kingdom
simon.parsons@kcl.ac.uk

Abstract. Prior work has investigated the problem mining arguments from online reviews by classifying opinions based on the stance expressed explicitly or implicitly. An implicit opinion has the stance left unexpressed linguistically while an explicit opinion has the stance expressed explicitly. In this paper, we propose a bipartite graph-based approach to relate a given set of explicit opinions as simplified arguments for a given set of implicit opinions using three different features (a) sentence similarity, (b) sentiment and (c) target. Experiments are carried out on a manually annotated set of explicit-implicit opinions and show that unsupervised sentence representations can be used to accurately match arguments with their corresponding simplified versions.

Keywords: argument mining · argument simplification

1 Introduction

The rise of social media has allowed people to share their opinions, in the form of reviews and debates, on online portals. Argument mining [18], an emerging research area, aims to discover arguments that are present in such user-based content. This paper is a contribution to work on argument mining.

In prior work [20], opinions were extracted from a set of hotel reviews and manually annotated as explicit or implicit based on how the stance in the opinion is expressed. Stance in NLP research refers to the standpoint taken by the user, whether they are for or against the given topic. In linguistics, stance is defined as "*the expression of attitude, judgement of the user towards the standpoint taken in the content*". According to this definition, stance can be expressed either explicitly in a sentence or must be inferred from the context.

The question we explore here is that **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?**. We consider that explicit and implicit opinions are two different ways of expressing the same argument. The difference is that an explicitly stated opinion might be easier to spot and understood by a human than an implicitly stated one, since understanding an implicit opinion requires inference from the context. We

can see examples in Table 1, which gives pairs of implicit opinions and explicit opinions that express the same argument.

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

Table 1. Implicit opinions with corresponding explicit opinions as their simplified arguments.

Given a set of explicit and implicit opinions, we propose a bipartite graph-based approach to identify whether a given explicit opinion is a simplified argument representation of an implicit opinion. We perform experiments on a hotel review dataset with and without the implicit/explicit opinion classification using three different types of features: (a) sentence similarity using different sentence embedding representations, (b) sentiment and (c) target information. We also experiment with a dialogue-based argumentation dataset (Citizen's Dialogue), which contains speaker's arguments annotated using the rephrase relation [13]. Our results show that the semantic similarity scores obtained using the unsupervised sentence embeddings give good performance for both datasets reporting the best accuracies of 0.86 and 0.82 respectively on the hotel review and Citizen's Dialogue datasets.

2 Related work

Existing work on argument mining can be broadly classified into monological and dialogical texts depending on the user interaction. Work on monological texts deal with persuasive essays [21], articles [1] and online reviews [23, 25]. Work on dialogical texts deals with debates [11], Tweets [6], dialogues [24], and other forms of user interactions [10].

Ghosh et al. [10] annotate user comments in forums as target-callout pairs based on pragma-dialectic theory.annotate user comments in forums as target-callout pairs based on pragma-dialectic theory and also investigate on the difficulties faced in doing the annotation task. This work is useful for the research community to understand the difficulties of annotating arguments in social media

texts. Boltuzic et al. [4, 3] have done continuous assessment on identify premises and claim, in particular, how they are related in debates. Their definition of support and attack depends on whether the relation is explicit or not. The authors also have created a dataset consisting of 125 claim pairs containing annotated premises for filling the gap between a user claim and the main claim of a topic. An advantage of this work is the availability of the dataset that can be used for comparing whether the model proposed can be useful for other available datasets.

Habernal et al. [11] build a large corpus based on the extended Toulmin model from debate portals using a semi-supervised approach. The different components annotated to represent an argument are the following:- premise, claim, backing, rebuttal and refuta- tion. The semi-supervised approach automatically extracts features from an unlabelled corpus by clustering word embedding vectors for classifying whether a given sentence is an argument or not. An advantage of this work is the semi-supervised approach that has been evaluated in detail for in-domain and cross-domain data along with detailed error analysis. Differing from the rest, Duthie et al. [9] work on political based debate corpus to identify ethos, which is an important part of argumentation. Walker et al. [24] determine how persuasive arguments are from the audience perspective while Oraby et al. [17] classify a dialogue based on whether it is factual or emotional.

Not only does argument mining focus on annotating arguments and its components (see [14] for a detailed survey), recent work has also considered the problem of extracting relations that exist between arguments. Cabrio and Villata [7] extract abstract arguments from debates to form a bipolar argumentation framework, with the support and attack relation automatically identified using textual entailment. In this paper, the authors empirically demonstrate that, in most cases, support and attack relation satisfy entailment and contradiction relations respectively. Boltuzic et al. [3] relate arguments using implicit/explicit support and attack relations. Similarly, Bosc et al. [5] annotate the support and attack relation among arguments present in tweets. Among scientific articles, the support and attack relation were extracted by Kirschner et al. [12].

Instead of extracting relations, Carstens and Toni [8] investigate towards how relation information can help in identifying arguments. In their work, they show how in many cases, the objective statements often ignored can actually constitute an argument. Thus, they consider pairs of sentences that satisfy either the *support*, *attack* or *neither* relation to demonstrate the same.

Konat et al. [13] studied the rephrase relation between two arguments. According to that work, rephrased arguments must not be considered as an additional support/attack. In particular, they consider a particular dialogue corpus where the same argument is made multiple different people. We propose an unsupervised approach that uses rephrase relations for argument simplification.

3 Background

In our prior work [20], we considered a statement expressed by a sentence, which can be either positive or negative in sentiment and talks about a single target entity, to be a stance expressing an opinion. Further, opinions present in a set of hotel reviews were annotated as implicit or explicit depending on how the stance is expressed within the text. The following guidelines were given to the human-annotators:

Explicit opinion: Direct expression of approval/disapproval towards the hotel or its aspects. Certain words or clauses have a strong intensity of expression towards a particular target. For example, *worst staff!* has a stronger intensity against the target *staff* than *the staff were not helpful*.

Implicit opinion: Those words or clauses that do not have a strong intensity of expression towards a particular target. In the above example *staff were not helpful* is an implicit opinion. Moreover, personal facts such as *small room*, *carpets are dirty* etc. Some of them may also be in the form of justifications or describing an incident.

4 Bipartite graph-based Opinion Matching

Given a set of opinions classified as implicit/explicit, we formulate the problem of identifying simplified arguments as a maximum cost K ranked bipartite graph-matching problem. The bipartite graph is formed by mapping each implicit opinion with each of the given explicit opinions. For every implicit opinion, the top K explicit opinions with the smallest costs are considered. Three different features are explored in computing the cost function for every implicit-explicit mapping as described in the following sections.

4.1 Unsupervised Sentence Embedding

To measure sentence similarity we use both unsupervised and supervised sentence-embedding representations. First, each word is represented using pre-trained embedding vectors. Based on existing works [2, 16], we perform different steps on the pre-trained word embeddings to create sentence embeddings. Mu et al. [16] perform two post-processing steps on pre-trained word embeddings. The motivation of their work is to create better word embedding representations and hence they do not focus on sentence representation. They show that word embeddings are narrowly distributed in a cone and by subtracting the mean vector and applying Principal Component Analysis (PCA), it is possible to obtain an isotropic spherical distribution, which is better at recognising similar word pairs. The two post-processing steps are described next.

Diff: Assume we are given a set 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 V is given by:

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

Following [16], the mean is subtracted from each word embedding to create isotropic embeddings as follows:

$$\forall_{w \in \mathrm{V}} \quad \tilde{\boldsymbol{w}} = \boldsymbol{w} - \hat{\boldsymbol{w}} \tag{2}$$

WordPCA: The mean-subtracted word embeddings given by (2) for all $w \in \mathrm{V}$ are arranged as columns in a matrix $\mathbf{A} \in \mathbb{R}^{k \times |\mathrm{V}|}$, and its d principle component vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d$ are computed. Mu et al. [16] proposed an embedding which removes the l most important principle components:

$$\boldsymbol{w}' = \tilde{\boldsymbol{w}} - \sum_{i=1}^{l} (\boldsymbol{u}_i \boldsymbol{w})\, \boldsymbol{u}_i \tag{3}$$

We use these word embeddings to create sentence embeddings:

AVG: A simple, 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 $\{w | w \in \mathrm{S}\}$. We then create its sentence embedding $\boldsymbol{s} \in \mathbb{R}^k$ as follows:

$$\boldsymbol{s} = \frac{1}{|\mathrm{S}|} \sum_{w \in \mathrm{S}} \boldsymbol{w} \tag{4}$$

Depending on the pre-processing applied on the word embeddings used in (4), three different variants for sentence embeddings are possible: **AVG** (uses unprocessed word embeddings $\boldsymbol{w}$), **Diff+AVG** (uses $\tilde{\boldsymbol{w}}$) and **WordPCA+AVG** (uses $\boldsymbol{w}'$).

WEmbed: Arora et al. [2] proposed a method to create 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:

$$\psi(w) = \frac{a}{a + p(w)} \boldsymbol{w} \tag{5}$$

$$\boldsymbol{s} = \frac{1}{|\mathrm{S}|} \sum_{w \in \mathrm{S}} \psi(w) \boldsymbol{w} \tag{6}$$

Here, a is a small constant[3]. Intuitively, frequent words such as stop words will have a smaller weight assigned to them, effectively ignoring their word embeddings when computing the sentence embeddings.

[3] Set to 0.001 in our experiments

SentPCA: Given a set of sentences, Arora et al. [2] applied PCA on the matrix that contains individual sentence embeddings as columns to compute the first principal component vector $\boldsymbol{v}$, which is subtracted from each sentence's embedding as follows: In total we have five sentence embedding methods (**AVG**, **Diff+AVG**, **WordPCA+AVG**, **WEmbed** and **SentPCA**). In the unsupervised approach, we measure the similarity between an implicit and an explicit opinion as the cosine similarity between their corresponding sentence embeddings.

4.2 Supervised Sentence Similarity

We propose a supervised method to compute the similarity between two sentences using their sentence embeddings, created from pre-trained word embeddings as described in Section 4.1 using a training dataset, where each pair of sentences is manually rated for the degree of their semantic similarity. Specifically, given two sentences s_i, s_j, we first compute their sentence embeddings, respectively $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, using one of the unsupervised sentence embedding methods described in Section 4.1. Next, we represent a pair of sentences using two operators: $\boldsymbol{h}_\times$ (elementwise multiplication) and $\boldsymbol{h}_-$ (elementwise absolute value of the difference).[4] Intuitively, $\boldsymbol{h}_\times$ captures common attributes in the two sentences, whereas $\boldsymbol{h}_-$ captures attributes unique to one of the two sentences. We then feed $\boldsymbol{h}_\times$ and $\boldsymbol{h}_-$ to 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:

$$\begin{aligned}
\boldsymbol{h}_\times &= \boldsymbol{s}_i \odot \boldsymbol{s}_j \\
\boldsymbol{h}_s &= \sigma\left(\mathbf{W}^{\times}\boldsymbol{h}_\times + \mathbf{W}^{(-)}\boldsymbol{h}_- + \boldsymbol{b}^{(h)}\right) \\
\boldsymbol{h}_- &= |\boldsymbol{s}_i - \boldsymbol{s}_j| \\
\hat{\boldsymbol{p}}_\theta &= \phi\left(\mathbf{W}^{(p)}\boldsymbol{h}_s + \boldsymbol{b}^{(p)}\right)
\end{aligned}$$

We use the SICK [15] sentence similarity dataset that consists of pairs of sentences manually rated in an ordinal range from 1 to 5, where 1 represents the lowest and 5 represents the highest similarity. We denote this gold standard rating for s_i and s_j by $y(s_i, s_j) \in [1, K]$, where $K = 5$ for the SICK dataset. We use the class probability distribution, $\hat{\boldsymbol{p}}_\theta$ to compute the expected similarity rating $\hat{y}(s_i, s_j)$ between s_i and s_j as follows:

$$\hat{y}(s_i, s_j) = \boldsymbol{r}\hat{\boldsymbol{p}}_\theta \tag{7}$$

Here, the rating vector is $\boldsymbol{r} = (1, 2, \ldots, K)$. We would like the expected rating to be close to the gold standard rating. Following Tai et al. [22], we define a sparse

[4] We drop the arguments of the operators to simplify the notation.

target distribution $\boldsymbol{p}$ that satisfies $y = \boldsymbol{rp}$:

$$p_i = \begin{cases} y - \lfloor y \rfloor & \text{if } i = \lfloor y \rfloor + 1 \\ y - \lfloor y \rfloor + 1 & \text{if } i = \lfloor y \rfloor \\ 0 & \text{otherwise} \end{cases}$$

The parameters θ of the model are found by minimising the KL-divergence between $\boldsymbol{p}$ and $\hat{\boldsymbol{p}}_\theta$ subjected to ℓ_2 regularisation over the entire training dataset D of sentence pairs as follows:

$$J(\theta) = \sum_{(s_i, s_j) \in \mathrm{D}} \mathrm{KL}\left((p^{(k)} || \hat{p}_\theta^{(k)}\right) + \frac{\lambda}{2} ||\theta||_2^2 \tag{8}$$

Here, $\lambda \in \mathbb{R}$ is the regularisation coefficient, set using validation data. The cost function of the bipartite matching problem using sentence similarity can then be defined as follows.

$$\mathrm{C}(i, j) = \mathrm{sim}(\boldsymbol{w_i}, \boldsymbol{w_j}) \tag{9}$$

Here, sim is the cosine similarity between sentence embeddings for the unsupervised approach and the predicted similarity rating $\hat{y}$ for the supervised approach.

4.3 Sentiment and Target

Sentiment and target play an important role among these stance-bearing opinions, and we can maximise the cost function by considering these two features. For this purpose, we define the cost function as follows:

$$\mathrm{C}(i, j) = \mathrm{sim}(\boldsymbol{s_i}, \boldsymbol{s_j}) + \mathrm{Q}(i, j) + \mathrm{R}(i, j) \tag{10}$$

Q and R output a threshold value if S_i and S_j have the same sentiment and target.

We focus on whether implicit/explicit classification along with linguistic structure can help in identifying an simplified argument for a given argument, without the sentiment and target information. In many cases, the target may be stated explicitly in the opinion or may mention the target implicitly. Hence, for our experiments, we make use of the dataset where the sentiment and target are manually annotated.

5 Experiments and Results

For our experiments, we use pre-trained Glove embeddings [19] with 300 dimensions[5]. For *WordPCA*, $l = 2$ is used [16]. Sentiment of an opinion and the targets present are manually annotated. Here, a domain knowledge base related to the different aspects and aspect categories is used. The threshold values for both the sentiment and target functions (given in (10)) were varied from 0 to 1 on development data and we found that 0.5 is appropriate such that the cost function is not biased towards the sentiment and target information alone.

[5] `https://nlp.stanford.edu/projects/glove/`

5.1 Evaluation measures

The evaluation measures used in our experiments were:

Precision@K (P@K) For every implicit opinion, the top K explicit opinions are obtained. The number of correct explicit opinions among the top K opinions are summed and divided by the total number of implicit opinions present. Thus:

$$\text{P@K} = \frac{1}{m} \sum_{i=1}^{m} \frac{n_i}{K} \tag{11}$$

where m is the total number of implicit opinions, and n_i is the number of correct explicit opinions for the corresponding *i-th* implicit opinion.

Averaged precision@K (Avg P@K)

$$\text{Avg P@K} = \frac{1}{K} \sum_{i=1}^{K} P@i \tag{12}$$

Here, K is the number of top explicit opinions that are considered and $P@i$ represents the precision@i score.

Mean reciprocal rank (MRR)

$$\text{MRR} = \frac{1}{m} = \sum_{i=1}^{i=m} \frac{1}{R_i} \tag{13}$$

where m is the total number of implicit opinions and R_i is the rank of the first correct explicit opinion for the i-th implicit opinion.

Accuracy (Acc)

$$\text{Acc} = \frac{1}{m} \sum_{i=1}^{m} l \tag{14}$$

where $l = 1$ if at least one of the correct explicit opinions is present within the top 10 explicit opinions; otherwise 0. This is because in the case of the Citizen Dialogue corpus, exactly one argument is matched as a simplified argument for another.

We randomly selected 57 implicit opinions from the implicit/explicit opinions dataset[6]. Each implicit opinion is manually tested with the three most appropriate explicit opinions that are the corresponding simplified arguments from the dataset. In total, we have 56 explicit opinions. Again, each implicit opinion was manually verified with the 56 explicit ones and any of those that represent as simplified arguments for the implicit opinion was updated. The number of explicit opinions that are simplified arguments of the implicit opinions ranged

[6] This dataset contains 1288 opinions manually annotated by two annotators with an inter-annotator agreement with a Cohen's kappa of 0.71

from a minimum of 1 to a maximum of 13. On average, the number of explicit opinions as simplified arguments for an implicit opinion was 6.

A bipartite graph with the implicit and explicit opinions as nodes and edges drawn from every implicit opinion to each of the explicit opinion is considered. For each implicit opinion, the top K explicit opinions with the cost function score ranging from highest to lowest are considered as correctly predicted simplified arguments. The cost function is computed using different features as described in Section. 4. These top K explicit opinions were then compared against the manually identified explicit opinions.

In Table. 2, we report the P@K for values of $K = 10, 15$ and 20 and the Avg P@K for values of $K = 15$ and 20. We observe that using **SENTPCA** does not perform better than the simple baseline **AVG**. The results also show that **WordPCA+AVG** is the best sentence embedding representation useful for predicting the correct explicit opinions. The similarity scores obtained using this unsupervised sentence embedding representation do better than the sentiment and target functions, and we get the best performance using all three types of features.

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**

Table 2. 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 in Section 4. Moreover, sentiment and target functions are computed. Precision@K with K = 10,15,20 are computed and the results are present. In addition, average Precision@K with K = 15 and 20 are computed and the results are shown.

The Citizen's Dialogue corpus contains the rephrase relation identified among premises present in the same argument structure present within the same dia-

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

Table 3. We compute the sentence similarity based on the methods described in Section 4. Mean reciprocal rank (MRR) and accuracy (Acc) is computed. The results 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).

logue. As the related premises belong to the same argument structure, a premise with additional information rephrases a premise with less information but which has a similar meaning. We collected 64 argument pairs with rephrase relation from this corpus for our experiments. Firstly, we are interested to know how the implicit/explicit opinion classification helps in identifying simplified arguments for a given set of arguments. To make a fair comparison against the Citizen Dialogue corpus and to assess the adaptability of our method, we assume that there is a classification system that is able to classify a premise as being a simplified argument or not. For instance, the length of the premise could be considered as one feature. An example from the corpus is given below:

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.*.

We experimented on two different settings — one where the information about whether an argument is simplified or not is given and the other where the information is not given.

The results are reported in Table 3. We observe that, for both datasets, the best performances for supervised and unsupervised approach are obtained using **WordPCA+AVG**. The implicit/explicit classification significantly improves the performance for the implicit/explicit dataset. Overall, performing the two post-processing steps on pre-trained embeddings gives the best sentence embedding representation using the simple average based embedding method.

5.2 Analysis of Results

The results in the previous section provide quantitative measures of performance in identifying simplified arguments for a given set of arguments. In this section, we look in some detail at the performance of similarity measure, sentiment and target in predicting the correct answers. First, consider the results when the cost function uses all three functions — $\mathrm{sim}, \mathrm{Q}, \mathrm{R}$ (Eq. 10) — for computing the cost. These are compared with the results when the cost function uses only the sentiment and target function (Q, R in Eq. 10). We use *WordPCA+AVG* for computing the similarity measure.

We find that, 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. Given the implicit opinion *"but the service is totally different with so many rooms for improvement it became not acceptable"*, the first ranked predicted explicit opinion when using all three functions ($\mathrm{Sim} + \mathrm{Q} + \mathrm{R}$) for computing the cost was *"we were extremely unimpressed by the quality of service we encountered"*. Both the implicit and explicit opinion express the same argument about the aspect "service" and hence, the answer is correct.

For the same example, the first ranked predicted explicit opinion using the sentiment and target functions ($\mathrm{Q} + \mathrm{R}$) for computing the cost is *"the rooms are not worth the money"*. We can see that the word "rooms" in the implicit opinion has been wrongly considered to refer to hotel rooms, and this mismatch cannot be captured using the sentiment and target information alone. This mismatch means that the prediction is incorrect. The sentiment and target functions, unlike the similarity measure, do not capture any contextual information and might predict answers randomly based on the sentiment and target information.

A second example starts with the implicit opinion *"this hotel could easily be 5 star, the facilities are fantastic, the rooms are beautifully furnished and equipped with all the latest technology"*. Here the top-ranked explicit opinion using $\mathrm{Sim} + \mathrm{Q} + \mathrm{R}$ is *"the hotel rooms are nice"* is a correct match for the implicit opinion, while the top-ranked opinion using $\mathrm{Q} + \mathrm{R}$ is *"the rooms are not worth the money"* which, while the aspect has been correctly determined to be hotel rooms, is completely wrong.

Both the previous examples show the similarity measure are working well. It is the elements of the cost function that makes it possible to find a good match. However there are some cases where the contextual information captured by the similarity measure is not sufficient to detect a good match. This is where the domain knowledge information that identifies different aspects as the same target is not captured by the similarity measure. For example the implicit opinion *"the laundry came back promptly"* is correctly matched with the explicit opinion *"the service was great"* by the sentiment and target functions, but the similarity measure does not recognise these opinions as being similar. This might be because both 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. It is also possible that the embeddings of the words

"laundry" and "service" were not available or were not present as close word pairs. Understanding the performance of the similarity measure is something we will investigate more.

6 Conclusion

We proposed an unsupervised bipartite graph-based approach to automatically predict among opinions, where one opinion can be represented as a simplfiied argument of the another without changing the context. Three different features: sentence similarity, sentiment and target are used for computing the cost function. Our experimental results on two different datasets show that unsupervised sentence representations help in matching arguments with their corresponding simplified arguments. Moreover, we observe that the 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**. This, in combination with sentiment and target gives a precision@10 of 0.28.

References

1. Al-Khatib, K., Wachsmuth, H., Kiesel, J., Hagen, M., Stein, B.: A news editorial corpus for mining argumentation strategies. In: CicLing. pp. 3433–3443 (2016)
2. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings (2017)
3. Boltužić, F., Šnajder, J.: Back up your stance: Recognizing arguments in online discussions. In: ACL. pp. 49–58 (2014)
4. Boltuzic, F., Šnajder, J.: Fill the gap! analyzing implicit premises between claims from online debates. In: ArgMining@ACL. pp. 124–133 (2016)
5. Bosc, T., Cabrio, E., Villata, S.: Dart: a dataset of arguments and their relations on twitter. In: LREC. pp. 1258–1263 (2016)
6. Bosc, T., Cabrio, E., Villata, S.: Tweeties squabbling: Positive and negative results in applying argument mining on social media. In: COMMA. pp. 21–32 (2016)
7. Cabrio, E., Villata, S.: Combining textual entailment and argumentation theory for supporting online debates interactions. In: ACL. pp. 208–212 (2012)
8. Carstens, L., Toni, F.: Towards relation based argumentation mining. ArgMining@ACL pp. 29–34 (2015)
9. Duthie, R., Budzynska, K., Reed, C.: Mining ethos in political debate. In: COMMA. pp. 299–310 (2016)
10. Ghosh, D., Muresan, S., Wacholder, N., Aakhus, M., Mitsui, M.: Analyzing argumentative discourse units in online interactions. In: ACL. pp. 39–48 (2014)
11. Habernal, I., Gurevych, I.: Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In: EMNLP. pp. 2127–2137 (2015)
12. Kirschner, C., Eckle-Kohler, J., Gurevych, I.: Linking the thoughts: Analysis of argumentation structures in scientific publications. In: ArgMining@EMNLP. pp. 1–11 (2015)
13. Konat, B., Budzynska, K., Saint-Dizier, P.: Rephrase in argument structure. FLA Workshop@COMMA'16 pp. 32–39 (2016)

14. Lippi, M., Torroni, P.: Argument mining: A machine learning perspective. In: TAFA. pp. 163–176 (2015)
15. Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., Zamparelli, R.: Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In: SemEval@ COLING. pp. 1–8 (2014)
16. Mu, J., Bhat, S., Viswanath, P.: All-but-the-top: Simple and effective postprocessing for word representations. CoRR **abs/1702.01417** (2017)
17. Oraby, S., Reed, L., Compton, R., Riloff, E., Walker, M.A., Whittaker, S.: And that's A fact: Distinguishing factual and emotional argumentation in online dialogue. In: ArgMining@HLT-NAACL. pp. 116–126 (2015)
18. Palau, R.M., Moens, M.F.: Argumentation mining: the detection, classification and structure of arguments in text. In: ICAIL. pp. 98–107 (2009)
19. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
20. Rajendran, P., Bollegala, D., Parsons, S.: Contextual stance classification of opinions: A step towards enthymeme reconstruction in online reviews. In: ArgMining@ACL'16 (2016)
21. Stab, C., Gurevych, I.: Parsing argumentation structures in persuasive essays. Computational Linguistics **43**(3), 619–659 (2017)
22. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: ACL. pp. 1556–1566 (2015)
23. Villalba, M.P.G., Saint-Dizier, P.: A framework to extract arguments in opinion texts. IJCINI **6**(3), 62–87 (2012)
24. Walker, M.A., Anand, P., Lukin, S.M., Whittaker, S.: Argument strength is in the eye of the beholder: Audience effects in persuasion. In: EACL. pp. 742–753 (2017)
25. Wyner, A., Schneider, J., Atkinson, K., Bench-Capon, T.J.M.: Semi-automated argumentative analysis of online product reviews. In: COMMA. pp. 43–50 (2012)