Octa: Omissions and Conflicts in Target-Aspect Sentiment Analysis

Zhe Zhang [*†], **Chung-Wei Hang** [*†], and **Munindar P. Singh**[‡]

[†]IBM Corporation
[‡]Department of Computer Science, North Carolina State University
{zhangzhe,hangc}@us.ibm.com, singh@ncsu.edu

Abstract

Sentiments in opinionated text are often determined by both aspects and target words (or *targets*). We observe that targets and aspects interrelate in subtle ways, often yielding conflicting sentiments. Thus, a naive aggregation of sentiments from aspects and targets treated separately, as in existing sentiment analysis models, impairs performance.

We propose Octa,[1] an approach that jointly considers aspects and targets when inferring sentiments. To capture and quantify relationships between targets and context words, Octa uses a selective self-attention mechanism that handles implicit or missing targets. Specifically, Octa involves two layers of attention mechanisms for, respectively, selective attention between targets and context words and attention over words based on aspects. On benchmark datasets, Octa outperforms leading models by a large margin, yielding (absolute) gains in accuracy of 1.6% to 4.3%.

1 Introduction

People share their opinions about almost anything: tourist attractions, restaurants, car dealerships, and products. Such opinionated texts do not merely help people make decisions in their daily life, but also help businesses measure consumer satisfaction to improve their offerings.

Sentiment analysis involves many aspects of Natural Language Processing, e.g., negation handling (Zhu et al., 2014), entity recognition (Mitchell et al., 2013), topic modeling (Zhang and Singh, 2018, 2019). Importantly, opinionated texts often convey conflicting sentiments. Distinct sentiments may refer to distinct *aspects* of the domain in question—e.g., food quality of a restaurant or battery life of a smartphone. These predefined domain aspects may or may not appear in the texts. **Aspect-Based Sentiment Analysis (ABSA)** approaches (Wang et al., 2016; Xue and Li, 2018; Liang et al., 2019) predict sentiments from text about a given aspect. And, **Target-Based Sentiment Analysis (TBSA)** approaches (Chen et al., 2017; Fan et al., 2018; Li et al., 2018; Du et al., 2019; Zhang et al., 2019) predict sentiments of *targets* that appear in an opinionated text. Targets are usually entities in a review: e.g., a dish for a restaurant and a salesperson for a car dealership.

We posit that aspects and targets provide subtle, sometimes contradictory, information about sentiment and should therefore be modeled, not in isolation, but jointly. Considering them separately, as ABSA and TBSA approaches do, impairs performance. Take this review sentence from SemEval-15 as an example:

Conflicting Sentiments on Aspect

*We both had the **filet**, very good, didn't much like the **frites** that came with.*

If we ask about aspect *Food#Quality*, by disregarding targets during training, ABSA models fail to address the contradiction in sentiment about *filet* and *frites*, as do TBSA models, which focus on targets and disregard aspects. In the following review sentence from SemEval-16, the target *fish* is associated with opposite sentiments: positive for *Food#Quality* and negative for *Food#Style_options*.

Conflicting Sentiments on Target

*The **fish** was fresh , though it was cut very thin.*

Opinionated text is often not structured. Users may not always mention targets explicitly. In some cases, the entities in a sentence are not the targets associated with the sentiment. In other cases, users mention multiple targets with sentiments in a sen-

[*]Equal contribution.

[1]The data and source code of Octa can be found at https://github.com/chungweihang/octa

tence, but we need the overall sentiment. Consider the following two sentences from SemEval-16:

Implicit or Missing Target

(1) *You are bound to have a very charming time.*
(2) *Endless fun, awesome music, great staff!!!*

Here, (1) contains entity *You* and positive sentiment toward aspect *Restaurant#General* but omits mention of the target *restaurant*. And, (2) contains positive sentiment toward aspects *Ambience* and *Service*. It expresses a positive sentiment toward aspect *Restaurant#General* albeit with no target. How can we extract sentiments given an aspect with or without a target?

Contributions We propose Octa, an approach that jointly considers aspects and targets. Octa uses a selective attention mechanism to capture subtle Target-Context and Target-Target relationships that reduce noisy information from irrelevant relations. Octa uses (1) aspect embeddings with attention to incorporate aspect dependencies and (2) a surrogate target with BERT sequence embeddings to handle implicit or missing targets. Octa can classify different types of conflicting sentiments with aspects only, targets only, both, or none.

Octa yields strong results on six benchmark datasets including SentiHood and four SemEval datasets, i.e., 2014 (target and aspect), 2015, and 2016. Octa outperforms 16 state-of-the-art baselines by absolute gains in accuracy from 1.6% to 4.3%.

Sample Results of Octa We explain the benefit of Octa via a few examples from the SemEval-16 test set in Table 1. In case (a), the **same target** is paired with **different aspects**. Octa detects positive sentiment toward aspect *Food#Quality* based on target *fish* and context *fresh*. By attending to different context *cut very thin* but the same target, Octa detects negative sentiment toward aspect *Food#Style_options*. In case (b) where **different aspects** paired with the **same or different targets**, Octa correctly detects neutral sentiment toward target *food* for aspect *Food#Quality*. For target *restaurant*, Octa successfully detects conflicting sentiments toward different aspects by locating different context words. In case (c), the **same aspects** are paired with **different targets**. Octa correctly detects the conflicting sentiments toward the same aspect *Ambience#General*. Case (d) has **aspect with implicit target** and case (e) has **different aspects with or without target**. Octa successfully detects the sentiment toward implicit or missing target.

2 Problem Definition

The input of our sentiment analysis task is a sequence of words, with an aspect, or a target, or both. Our goal is to identify the sentiment polarity associated with the aspect and the target. Formally, Octa has three inputs,

- Sequence of words: $W = \{w_1, \ldots, w_N\}$,
- Target $T_i = \{t_1, \ldots, t_M\}$ where $t_i \in W$, and
- Aspect $a_i \in A = \{a_1, \ldots, a_{|A|}\}$ where A is a set of aspects.

The remaining words that are not part of the target are context words $C = \{c_1, \ldots, c_{N-M}\}$.

3 Octa Model Overview

Figure 1 shows the Octa architecture. To infer the sentiment for an aspect and a target composed of words from the sequence, first, Octa uses BERT to generate word embeddings. Second, Octa uses a selective attention mechanism to compute context word and target attention weights and applies them to word embeddings to generate targeted contextual embeddings. Third, Octa constructs aspect embeddings and uses the embeddings to compute aspect attention over target and context words.

Octa uses a multihead architecture to learn attention in diverse embedding subspaces. It fuses and normalizes embeddings from each head and uses a linear classification layer with a softmax activation for sentiment classification. To introduce nonlinearity, Octa uses feed-forward networks (shown in grey), each comprising two fully connected layers followed by a nonlinear activation.

3.1 BERT Embeddings

Octa uses Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) to generate word embeddings. BERT is a contextualized language representation model, pretrained on large corpora and fine-tuned on downstream tasks, including token-level classification (named entity recognition and reading comprehension) and sequence-level classification (semantic similarity and sentiment analysis). Despite its success on various benchmarks, BERT ignores the relationships among target words, context words, and aspects, which are crucial for sentiment analysis.

	Sentence	Aspect	Target	Sent.
(a)	*The **fish** was fresh , though it was cut very thin.*	Food#Quality Food#Style_options	*fish* *fish*	POS NEG
(b)	***Food** wise, it's ok but a bit pricey for what you get considering the **restaurant** isn't a fancy place.*	Food#Quality Restaurant#Prices Ambience#General	*Food* *restaurant* *restaurant*	NEU NEG NEU
(c)	*The **music** playing was very hip, 20-30 something pop music, but the **subwoofer to the sound system** was located under my seat, which became annoying midway through dinner.*	Ambience#General Ambience#General	*music* *subwoofer to the sound system*	POS NEG
(d)	*As part of a small party of four, our food was dropped off without comment*	Service#General	—	NEG
(e)	*Endless fun, awesome **music**, great **staff***	Ambience#General Service#General Restaurant#General	*music* *staff* —	POS POS POS

Table 1: Sample results of Octa.

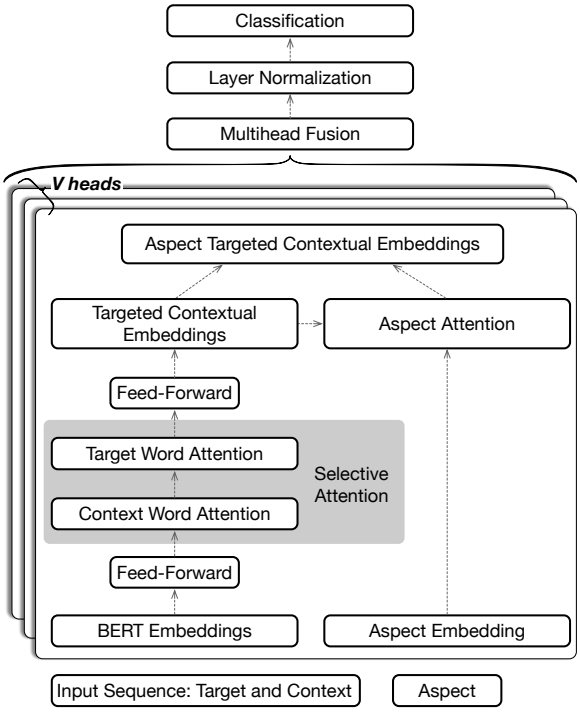

Figure 1: Architecture of Octa.

3.2 Selective Self-Attention Mechanism

Words connect with one another to form semantic relations and create meanings in different contexts. Self-attention (Vaswani et al., 2017) seeks to quantify this process. To capture relationships between words, it learns to represent each word using itself and the other words in the same sentence. The flexible structure of self-attention provides benefits in capturing different relations without range restriction. An ideal self-attention layer should attend to relations differently to create contexts for different goals. In practice, such flexibility may introduce noisy relations that lead to less-focused attention and confuse the decision layer.

In opinionated texts, context words carry sentiment. A context word can be associated with one or more targets. Thus, capturing Target-Context relationships is pivotal. We posit that capturing Target-Target relationships is important when targets contain multiple words. Context words can carry different sentiment when the same target word paired with other target words. For example, in the sentences *The wine list is long* and *The waiting list is long*, context word, *long*, is positive for target *wine list* but negative for target *waiting list*.

Octa uses a selective self-attention encoder to capture the subtle Target-Context and Target-Target relationships. Figure 2 shows the encoding process.

Formally, given a sentence containing one target t that consists of M target words and context c that consists of N context words, let $B_t = [b_{t_1}, \ldots, b_{t_M}] \in \mathbb{R}^{M \times d_B}$, $B_c = [b_{c_1}, \ldots, b_{c_N}] \in \mathbb{R}^{N \times d_B}$ denote the BERT embedding matrices of targets and context words, respectively, where d_B is the dimension of BERT embeddings. We use BERT's `[CLS]` token as either a target (when no target is provided) or a context word.

Feed-Forward Networks. Octa adopts a key-query-value attention structure (Vaswani et al., 2017) where keys, queries, and values are projected vectors. The structure first combines each query with all of keys through a compatibility function to generate attention weights. Then, it uses the weights to combine corresponding values to generate the output. Octa uses five feed-forward networks to construct keys, queries, and values for target and context words. Each feed-forward network comprises two fully connected linear layers connected by a GELU (Hendrycks and Gimpel, 2016)

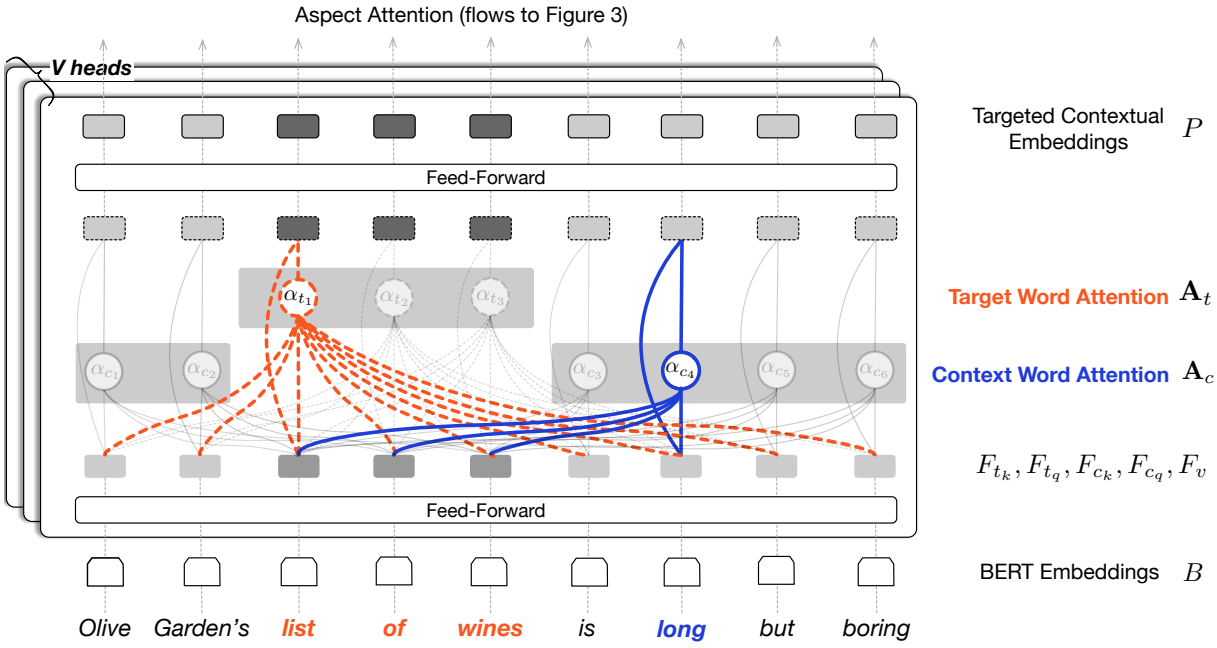

Figure 2: An illustration of target attention.

activation for element-wise nonlinear projection.

$$F_{t_k} = W_{t_{k1}} \cdot (\text{GELU}(W_{t_{k2}} \cdot B_t)), \quad (1)$$
$$F_{t_q} = W_{t_{q1}} \cdot (\text{GELU}(W_{t_{q2}} \cdot B_t)), \quad (2)$$
$$F_{c_k} = W_{c_{k1}} \cdot (\text{GELU}(W_{c_{k2}} \cdot B_c)), \quad (3)$$
$$F_{c_q} = W_{c_{q1}} \cdot (\text{GELU}(W_{c_{q2}} \cdot B_c)), \quad (4)$$
$$F_v = W_{v_1} \cdot (\text{GELU}(W_{v_2} \cdot [B_t \oplus B_c])), \quad (5)$$

where $F_{t_k}, F_{t_q} \in \mathbb{R}^{M \times d_F}$ are keys and queries of targets, $F_{c_k}, F_{c_q} \in \mathbb{R}^{N \times d_F}$ are keys and queries of context words, $F_v \in \mathbb{R}^{(M+N) \times d_F}$ are values for both kinds of words, $\oplus$ means matrix vertical concatenation, $W_{(\cdot)}$ are parameters to learn, and we omit the bias for simplicity.

Target Word Attention. Octa constructs an affinity matrix $\mathbf{A}_t = \{\alpha_{t_1}, \ldots, \alpha_{t_M}\} \in \mathbb{R}^{M \times (M+N)}$ by computing dot products of each target with each word in the sentence.

$$\mathbf{A}_t = \text{softmax}(F_{t_q} \cdot [F_{t_k} \oplus F_{c_k}]^T). \quad (6)$$

$\mathbf{A}_t$ is normalized row-wise to generate a list of attention weights for each target. These attention weights quantify relations between words and describe the amount of focus the encoder should place on other words when encoding a target. For sentences with no target, Octa uses BERT's `[CLS]` token as a surrogate target to leverage the aggregated sentence information.

Context Word Attention. Octa creates a mask matrix $\mathbf{K}_c = \{k_{c_1}, \ldots, k_{c_N}\} \in \mathbb{R}^{N \times (M+N)}$. Here, k_{c_i} equals 1.0 if the corresponding position is context word c_i or a target and zero otherwise. Octa constructs the affinity matrix $\mathbf{A}_c = \{\alpha_{c_1}, \ldots, \alpha_{c_N}\} \in \mathbb{R}^{N \times (M+N)}$ by computing the dot products of each context word with itself and each target in the sentence masked by $\mathbf{K}_c$, where $\circ$ denotes Hadamard product.

$$\mathbf{A}_c = \text{softmax}(F_{c_q} \cdot [F_{t_k} \oplus F_{c_k}]^T \circ \mathbf{K}_c). \quad (7)$$

$\mathbf{A}_c$ is normalized row-wise to generate a list of attention weights for each context word. These attention weights quantify dependencies between each context word and each target. Our mask removes noisy dependencies between the context words.

Targeted Contextual Embeddings. Given target attention $\mathbf{A}_t$ and context word attention $\mathbf{A}_c$, Octa computes targeted contextual embeddings $P \in \mathbb{R}^{(M+N) \times d_F}$ as follows.

$$P = [\mathbf{A}_t \oplus \mathbf{A}_c] \cdot F_v. \quad (8)$$

3.3 Aspect Attention

How the aspects and words in a sentence relate is vital in inferring sentiments. As the second review sentence in Section 1 shows, one target can associate with different sentiments for different aspects.

To incorporate aspect information, given L aspects, $A = \{a_1, \ldots, a_L\}$, Octa learns a list of aspect embeddings $F_A = \{f_{a_1}, \ldots, f_{a_L}\} \in \mathbb{R}^{L \times d_E}$ as follows,

$$F_A = W_{A_1} \cdot (\text{GELU}(W_{A_2} \cdot E)), \quad (9)$$

where $E = \{e_{a_1}, \ldots, e_{a_L}\}$, $e_{a_i} \in \mathbb{R}^{d_E}$ are a list of randomly initialized aspect keys, W_{A_1} and W_{A_2} are weights to learn, and bias is omitted for simplicity. To capture the relationships, Octa builds the affinity matrix $\mathbf{A}_a \in \mathbb{R}^{M+N}$ between aspect embeddings f_{a_i} and targeted contextual embeddings P of the sentence.

An illustrative example of aspect attention is shown in Figure 3.

$$\mathbf{A}_{a_i} = \text{softmax}(f_{a_i} \cdot P^T). \tag{10}$$

The aspect and targeted contextual embeddings Q_{a_i} for aspect a_i, $Q_{a_i} \in \mathbb{R}^{d_E+d_F}$, are computed as

$$Q_{a_i} = [\mathbf{A}_{a_i} \cdot P] \odot f_{a_i}, \tag{11}$$

where $\odot$ denotes horizontal matrix concatenation.

3.4 Multihead Fusion

To attend in parallel to relation information from different dimensional subspaces, Octa uses a multihead architecture with V heads. The final aspect and targeted contextual embeddings $H_{a_i} \in \mathbb{R}^{V*(d_E+d_F)}$ for aspect a_i is the fusion of all heads.

$$H_{a_i} = [Q_{a_i}^{h_1} \odot, \ldots, \odot\, Q_{a_i}^{h_V}]. \tag{12}$$

3.5 Sentiment Classification

For sentiment classification, Octa first applies layer normalization (Ba et al., 2016) on the multihead fusion. Then, it uses a fully connected linear layer followed by a softmax activation to to project H_{a_i} to $y \in \mathbb{R}^S$, the posterior probability over S sentiment polarities, is y (omitting the bias):

$$y = \text{softmax}(W_y \cdot H_{a_i}), \tag{13}$$

where W_y is parameter to learn. We train Octa with cross-entropy loss.

4 Empirical Evaluation

4.1 Data

We train and evaluate Octa on six benchmark datasets, described in Table 2, from three domains.

4.2 Parameter Settings

We set the dimension of aspect embeddings d^E to 1,024. For all feed-forward networks, we use 1,024 as the dimension of both inner and outer states d^F. We train Octa with 16 attention heads and freeze aspect embeddings during training.

We follow the literature in that we do not further split SemEval training sets into training and validation sets due to their size. Instead, we use SentiHood-dev for parameter tuning. For regularization, we add dropouts with a rate of 0.1 between the two fully connected layers in each nonlinear feed-forward network. For optimization, we use Adam (Kingma and Ba, 2015) and set $\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay = 0.01, and the learning rate = 1e-5, with a warmup over 0.1% of training.

For all experiments, we train Octa for 10 epochs on mini-batches of 32 randomly sampled sequences of 128 tokens. We repeat the training and testing cycle five times using different random seeds. Our evaluation metrics include accuracy and macro F_1 score. We perform the two-sampled t-test on the improvement of Octa over BERT. As reported in (Devlin et al., 2019), we observe unstable performance for both Octa and BERT. We perform several restarts and select best performed models. For model size, Octa introduces 2.5% more parameters (343M) compared with BERT sequence classification (335M, whole word masking). Training on SemEval-16 with single NVIDIA Tesla V100 takes 69 seconds/epoch for Octa and 65 seconds/epoch for BERT.

4.3 Baselines

We compare the performance of Octa against the following published models.

Feature based Baselines: **NRC-Canada**, **DCU**, **Sentiue**, and **XRCE** require feature engineering based on linguistic tools and external resources. Of these, NRC-Canada and DCU achieve the best performance on SemEval 2014 sentiment classification for aspect category and aspect term, respectively. Sentiue and XRCE are the best performing for SemEval 2015 and 2016, respectively.

TBSA Baselines: **RAM** (Chen et al., 2017) builds position-weighted memory using two stacked BiLSTMs and the relative distance of each word to the left or right boundary of each target. It uses a GRU with multiple attention computed using the memory. **TNet-AS** (Li et al., 2018) dynamically associates targets with sentence words to generate target specific word representation and uses adaptive scaling to preserve context information. **MGAN** (Fan et al., 2018) is an attention network based on BiLSTM that computes coarse-grained attention using averaged target embeddings and context words and leverages word

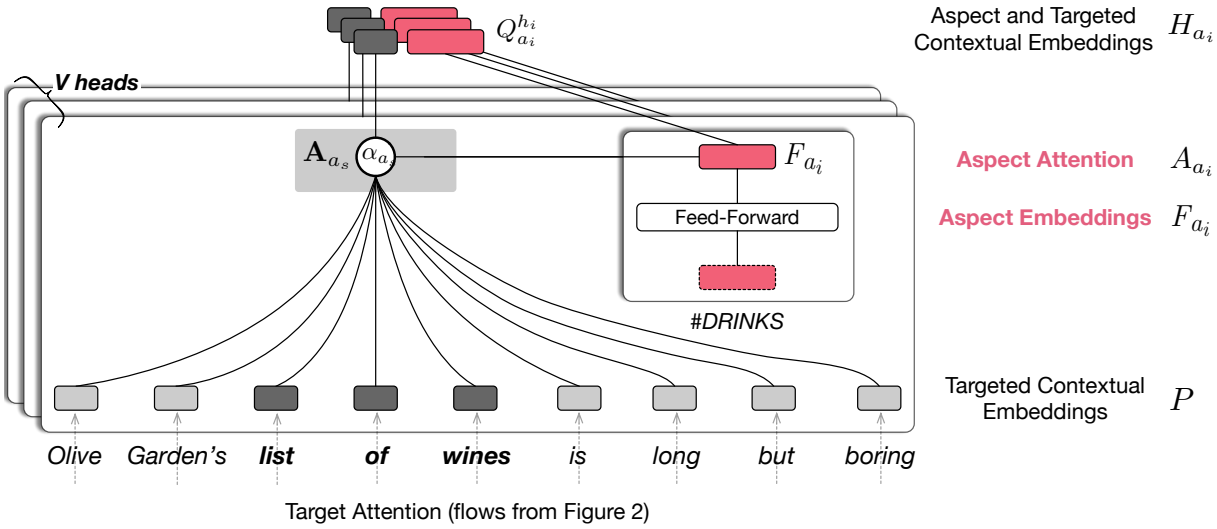

Figure 3: An illustration of aspect attention.

Table 2: Datasets. SemEval has restaurant review sentences. SentiHood has sentences about urban neighborhoods. SemEval-14-T has sentiments for targets without aspects and SemEval-14-A for aspects without targets. SemEval-15, SemEval-16, and SentiHood have targets and aspects.

Dataset			Positive		Neutral		Negative	
	Aspects	Labels	Train	Test	Train	Test	Train	Test
SemEval-14A	5	3	2,179	657	695	146	839	222
SemEval-14T	5	3	2,164	728	724	210	805	196
SemEval-15	13	3	1,198	454	53	45	403	346
SemEval-16	12	3	1,657	611	101	44	749	204
SentiHood-dev	12	2	2,480	616	–	–	921	224
SentiHood-test	12	2	2,480	1,217	–	–	921	462

similarity to build fine-grained attention. **IACapsNet** (Du et al., 2019) leverages capsule network to construct vector-based feature representation. It uses interactive attention EM-based capsule routing mechanism to learn the semantic relationship between targets and context words. **TNet-ATT** (Tang et al., 2019) leverages the relation between context words and model's prediction as supervision information to progressively refine its attention module for aspect based sentiment classification. **ASGCN-DG** (Zhang et al., 2019) builds Graph Convolutional Networks over dependency trees and uses masking and attention mechanisms to generate aspect-oriented sentence representations. **TD-GAT-BERT** (Huang and Carley, 2019) uses a Graph Attention Network to capture dependency relationship among words and an LSTM to model target related information.

ABSA Baselines: **ATAE-LSTM** (Wang et al., 2016) is based on LSTM. It uses aspect embeddings to learn attention weights. **GCAE** (Xue and Li, 2018) is a CNN with two convolutional layers that use different nonlinear gating units to extract aspect-specific information. **AGDT** (Liang et al., 2019) contains an aspect-guided encoder which consists of an aspect-guided GRU and a deep transition GRU to extract aspect-specific sentence representation. Note that GCAE and AGDT can be extended for TBSA. However, neither of them jointly considers both aspects and targets and therefore fails to handle conflicting sentiments.

Other Baselines: **Sentic LSTM** (Ma et al., 2018) uses an LSTM with a hierarchical attention mechanism to model both target and aspect attention. It incorporates commonsense knowledge into sentence embeddings. **BERT** does not consider aspects and targets. We compare with BERT to evaluate the performance gain from selective attention. We use the whole world masking pretrained BERT in our experiments. Additional results using BERT base and large models are in Appendix A.

4.4 Results

Table 3 compares Octa with baselines on SemEval datasets. For SemEval-14-A, AGDT outperforms GCAE, demonstrating the benefits of aspect-guided

	Model	SemEval-14A		SemEval-14T		SemEval-15		SemEval-16	
		Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Feature Based	NRC-Canada♯	82.92	–	80.05	–	–	–	–	–
	DCU♯	–	–	80.95	–	–	–	–	–
	Sentiue♯	–	–	–	–	78.69	–	–	–
	XRCE♯	–	–	–	–	–	–	88.13	–
Deep-Learning Based	ATAE-LSTM♯	77.20	–	–	–	–	–	–	–
	GCAE♯	79.35	–	77.28	–	–	–	–	–
	AGDT♯	81.78	–	78.85	–	–	–	–	–
	RAM♮	–	–	79.79	68.86	–	–	–	–
	MGAN♯	–	–	81.25	71.94	–	–	–	–
	TNet-AS♯	–	–	80.69	71.27	–	–	–	–
	IACapsNet♯	–	–	81.79	73.40	–	–	–	–
	TNet-ATT♯	–	–	81.53	72.90	–	–	–	–
	ASGCN-DG♯	–	–	80.77	72.02	79.89	61.89	88.99	67.48
	Sentic LSTM♯	–	–	–	–	76.47	–	–	–
	TD-GAT-BERT♯	–	–	83.00	–	–	–	–	–
	BERT	**86.15**	78.70	80.39	69.00	83.72	65.63	88.52	74.68
	Octa	86.03	**78.88**	**84.90**†	**77.57**†	**86.27**†	**67.17**	**90.10**†	**76.51**
	p-value vs. BERT	0.64	0.72	1.16e-6	1.27e-6	9.21e-4	8.14e-2	1.64e-4	6.12e-2

Table 3: Comparing accuracy and F_1 on SemEval tasks. Note that only Sentic LSTM and Octa can jointly consider aspects and targets. Results with ♯ are obtained from the original papers. Results with ♮ are obtained from (Li et al., 2018). Throughout, $*$ and † indicate if performance of BERT is significantly different from that of Octa at the levels of 0.05 and 0.001, respectively, measured by the two-sample t-test (p-values for the comparison with BERT are listed at the last row). See Appendix A for additional significance test results.

sentence representation. Octa outperforms AGDT and NRC-Canada with accuracy gains of 4.3% and 3.1%, respectively. Since SemEval-14-A lacks target information, Octa uses the BERT `[CLS]` token as the target. The result shows the benefit of selective attention to capture implicit target information. Octa and BERT yield comparable performance. We find that SemEval-14-A contains sentences with conflicting sentiments toward the same aspect. In the testing split, of 146 sentences labeled NEU, 52 sentences show conflicting sentiments—e.g., "the falafal was rather over cooked and dried but the chicken was fine" is labeled NEU for aspect *food* but contains positive sentiment toward target *chicken* and negative sentiment toward target *falafal*. We conjecture that such data defects undermine the benefit of selective attention.

SemEval-14-T lacks aspect labels so Octa treats it as one aspect. Octa outperforms all baselines with an accuracy gain of 1.9% compared with the best performing baseline, TD-GAT-BERT, of 4.0% over the feature-based baseline DCU.

SemEval-15 and SemEval-16 associate sentiment with both aspect and targets. Octa outperforms all baselines. Specifically, Octa obtains a 2.6% and 1.6% accuracy improvement over BERT on SemEval-15 and SemEval-16, respectively. The F_1 improvements over BERT are 1.5% and 1.8%. Also, Octa outperforms the top feature-based models, Sentiue and XRCE. The results demonstrate the benefit of jointly considering aspects and targets.

Model	SentiHood-D		SentiHood-T	
	Acc.	F_1	Acc.	F_1
Sentic LSTM	88.80	–	89.32	–
BERT	87.60	83.76	87.09	83.02
Octa	**92.17**†	**89.86**†	**91.34**†	**89.00**†
p-value	8.32e-9	1.46e-8	5.08e-9	1.16e-8

Table 4: Comparing performance on SentiHood data.

Table 4 shows the results on SentiHood. Octa outperforms the state-of-the-art Sentic LSTM with accuracy gains of 3.3% and 2.0% on dev and test, respectively. Sentic LSTM jointly considers both aspects and targets through a hierarchical attention mechanism. We attribute Octa's performance to its nonrecurrent architecture, which alleviates the dependency range restriction in LSTM, and to its selective attention mechanism, which reduces noisy dependency information from irrelevant relations.

To further evaluate Octa's capability of han-

dling sentences with conflicting sentiments, we apply trained BERT and Octa only on the conflicting samples from SemEval-15, SemEval-16, and SentiHood-test. There are 152, 96, 343 conflicting samples in SemEval-15, SemEval-16, and SentiHood-test, respectively. Table 5 shows the results. We see that for all datasets, Octa outperforms BERT with a large margin. The accuracy gains are 15.3%, 11.5%, and 19.8%, respectively.

Model	SemEval-15		SemEval-16		SentiHood-T	
	Acc.	F_1	Acc.	F_1	Acc.	F_1
BERT	52.55	39.26	52.50	43.88	53.24	51.14
Octa	**67.84**†	**51.52**†	**63.96***	**54.71**	**73.00**†	**72.68**†
p-value	1.84e-8	7.74e-7	9.97e-3	6.12e-2	1.52e-7	2.47e-7

Table 5: Comparing performance on conflicts.

4.5 Ablation Study

We evaluate variants of Octa on SemEval-15 to understand the contribution of aspects, targets, and selective attention. The same conclusion holds for the other datasets. As Table 6 shows, using target selective attention (Octa-Sel) yields 1.1% better accuracy but similar F_1 as using aspect attention (Octa-Asp). Combining aspect attention with target self-attention (Octa-Asp-Full) hurts performance and stability, as seen in the lower accuracy and F_1, indicating that simply applying self-attention on targets and context words introduces noisy information. Replacing self-attention with selective attention (Octa) yields gains in accuracy and F_1 of 4.3% and 6.1% respectively, indicating that selective attention is effective in combating noise.

Model	Aspect	Target	Acc.	F_1
Octa-Asp	Yes	–	84.09†	65.20*
Octa-Sel	–	Selective	85.16	65.21
Octa-Asp-Full	Yes	Self	82.01*	61.08
Octa	**Yes**	**Selective**	**86.27**	**67.17**

Table 6: Comparing model variants on SemEval-15.

5 Related Work

Sentiment analysis has received substantial attention over the last few years. We highlight here only the works most relevant to Octa.

5.1 Aspect-Based Sentiment Analysis (ABSA)

For the ABSA task, Wang et al. (2016) concatenate aspect embeddings with LSTM hidden states and apply attention mechanism to focus on different parts of a sentence given different aspects. Xue and Li (2018) extracts features from text using a convolutional layer and propagates the features to a max pooling layer based on either aspects or targets. Liang et al. (2019) uses an aspect-guided encoder with an aspect-reconstruction step to generate either aspect- or target-specific sentence representation. The above models do not jointly consider aspects and targets and suffer when a target has conflicting sentiments toward different aspects.

5.2 Target-Based Sentiment Analysis (TBSA)

For TBSA task, Tang et al. (2016) concatenate target and context word embeddings and use two LSTM models to capture a target's preceding and following contexts. Chen et al. (2017) builds position-weighted memory using two stacked BiLSTMs and the relative distance of each word to the left or right boundary of each target. Li et al. (2018) dynamically associates targets with sentence words to generate target specific word representation and uses adaptive scaling to preserve context information. Majumder et al. (2018) uses a GRU with attention to generate an aspect-aware sentence representation and a multihop memory network to capture aspect dependencies. Fan et al. (2018) uses BiLSTM with attention mechanism to computes coarse-grained attention using averaged target embeddings and context words. It leverages word similarity to build fine-grained attention. Xu et al. (2019) prepend target tokens to a given text sequence, and predict sentiment based on BERT sequence embeddings. Du et al. (2019) leverages capsule network and uses interactive attention capsule routing mechanism to learn the relationship between targets and context words.

6 Conclusion

The main innovation of Octa is to jointly consider aspects and targets. It uses selective attention to model the relationships between target and context words, and aspects to attend to targeted contexts to predict sentiments. Users can "query" Octa about sentiment of a particular aspect or target, or both. Our evaluation shows that Octa outperforms state-of-the-art models on SemEval, SentiHood, and conflicting sentiment datasets. Our ablation study shows that jointly modeling aspects and targets with selective attention is superior to selective attention only, aspect attention only, and aspect with self-attention.

References

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 22^{nd} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 452–461, Copenhagen.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 17^{th} Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, Minneapolis.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, Jianxin Liao, Tong Xu, and Ming Liu. 2019. Capsule network with interactive attention for aspect-level sentiment classification. In *Proceedings of the 24^{th} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5492–5501, Hong Kong.

Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of the 23^{rd} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3433–3442, Brussels.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*.

Binxuan Huang and Kathleen Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. In *Proceedings of the 24^{th} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5469–5477, Hong Kong.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3^{rd} International Conference on Learning Representations (ICLR)*, San Diego, CA.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56^{th} Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 946–956, Melbourne.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. A novel aspect-guided deep transition model for aspect based sentiment analysis. In *Proceedings of the 24^{th} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5572–5584, Hong Kong.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Proceedings of the 30^{th} AAAI Conference on Artificial Intelligence (AAAI)*, pages 5876–5883, New Orleans.

Navonil Majumder, Soujanya Poria, Alexander F. Gelbukh, Md. Shad Akhtar, Erik Cambria, and Asif Ekbal. 2018. IARM: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis. In *Proceedings of the 23^{rd} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3402–3411, Brussels.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of the 18^{th} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1643–1654, Seattle.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for target-dependent sentiment classification. In *Proceedings of the 26^{th} International Conference on Computational Linguistics (COLING)*, pages 3298–3307, Osaka.

Jialong Tang, Ziyao Lu, Jinsong Su, Yubin Ge, Linfeng Song, Le Sun, and Jiebo Luo. 2019. Progressive self-supervised attention learning for aspect-level sentiment analysis. In *Proceedings of the 57^{th} Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 557–566, Florence.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 30^{th} Annual Conference on Neural Information Processing Systems (NIPS)*, pages 6000–6010, Long Beach.

Yequan Wang, Minlie Huang, Li Zhao, and Xiaoyan Zhu. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 21^{st} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 606–615, Austin.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 17^{th} Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1–12, Minneapolis.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56^{th} Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2514–2523, Melbourne.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the 24^{th} Conference on Empirical Methods*

in Natural Language Processing (EMNLP), pages 4560–4570, Hong Kong.

Zhe Zhang and Munindar Singh. 2018. Limbic: Author-based sentiment aspect modeling regularized with word embeddings and discourse relations. In *Proceedings of the 23^{rd} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3412–3422, Brussels.

Zhe Zhang and Munindar Singh. 2019. Leveraging structural and semantic correspondence for attribute-oriented aspect sentiment discovery. In *Proceedings of the 24^{th} Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong.

Xiaodan Zhu, Hongyu Guo, Saif Mohammad, and Svetlana Kiritchenko. 2014. An empirical study on the effect of negation words on sentiment. In *Proceedings of the 52^{nd} Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 304–313, Baltimore.

A Appendices

We present additional results here.

Table 7: Comparing performance of Octa, $\text{BERT}_{\text{BASE}}$, and $\text{BERT}_{\text{LARGE}}$ on all tasks. Each experiment is repeated five times with different random seeds. Here, "sd" indicates one standard deviation.

Model	SemEval-14-A		SemEval-14-T	
	Accuracy	F_1	Accuracy	F_1
$\text{BERT}_{\text{BASE}}$	84.16, sd 0.53	76.37, sd 0.84	79.19, sd 0.62	67.88, sd 1.39
Octa-$\text{BERT}_{\text{BASE}}$	**84.53, sd 0.13**	**77.15, sd 0.48**	**82.93, sd 0.57**	**74.96, sd 1.02**
p-value	0.16	0.11	9.04e-6	1.63e-05
$\text{BERT}_{\text{LARGE}}$	85.17, sd 0.55	77.41, sd 0.80	79.82, sd 0.34	68.31, sd 0.84
Octa-$\text{BERT}_{\text{LARGE}}$	85.17, sd 0.52	**77.46, sd 0.77**	**83.30, sd 0.25**	**74.96, sd 0.25**
p-value	1.00	0.91	8.21e-8	1.44e-7

Model	SemEval-15		SemEval-16	
	Accuracy	F_1	Accuracy	F_1
$\text{BERT}_{\text{BASE}}$	79.10, sd 1.03	61.09, sd 1.82	86.52, sd 0.49	70.64, sd 0.99
Octa-$\text{BERT}_{\text{BASE}}$	**83.15, sd 1.04**	**64.83, sd 2.49**	**89.31, sd 0.56**	**75.33, sd 1.46**
p-value	2.63e-04	2.67e-02	3.04e-05	3.47e-4
$\text{BERT}_{\text{LARGE}}$	83.81, sd 0.64	**65.41 , sd 0.91**	88.85, sd 0.44	74.25, sd 1.27
Octa-$\text{BERT}_{\text{LARGE}}$	**84.85, sd 0.46**	64.96, sd 1.19	**90.45, sd 0.63**	**74.61, sd 2.37**
p-value	0.02	0.52	1.66e-3	0.77

Model	SentiHood-dev		SentiHood-test	
	Accuracy	F_1	Accuracy	F_1
$\text{BERT}_{\text{BASE}}$	86.52, sd 0.60	82.33, sd 0.83	86.54, sd 0.47	82.63, sd 0.93
Octa-$\text{BERT}_{\text{BASE}}$	**91.55, sd 0.79**	**89.05, sd 1.02**	**91.10, sd 0.37**	**88.73, sd 0.45**
p-value	3.39e-06	3.08e-6	1.43e-7	1.08e-6
$\text{BERT}_{\text{LARGE}}$	87.38, sd 0.35	83.32, sd 0.38	87.03, sd 0.30	83.09, sd 0.41
Octa-$\text{BERT}_{\text{LARGE}}$	**88.48, sd 0.36**	**84.91, sd 4.68**	**91.34, sd 0.59**	**89.03, sd 0.79**
p-value	0.51	0.47	5.03e-7	3.94e-7

Table 8: Comparing accuracy of Octa and BERT on all tasks. Each experiment is repeated five times with different random seeds. ", sd " indicates one standard deviation. The p-value row indicates if the accuracy of BERT is significantly different from Octa, measured by two sample t-test. We compare each of the 25 combination of experiments between BERT and Octa. The "BERT $\geq$ Octa" row counts the number of combinations where BERT is no worse than Octa, and how many of them are significant, measured by McNemar test. Similarly, "BERT $<$ Octa" counts the number of combinations where BERT is worse than Octa. For example, on SemEval-14-A, BERT is no worse than Octa in 12 combinations, none of which are significant. BERT performs worse than Octa in the other 13 combinations, none of which are significant either.

Model	SemEval-14-A	SemEval-14-T	SemEval-15	SemEval-16	SentiHood-dev	SentiHood-test
BERT	**86.15, sd 0.52**	80.39, sd 0.63	83.72, sd 0.96	88.52, sd 0.49	87.60, sd 0.17	87.09, sd 0.20
Octa	86.03, sd 0.16	**84.90, sd 0.45**	**86.27, sd 0.57**	**90.10, sd 0.22**	**92.17, sd 0.37**	**91.34, sd 0.31**
p-value	0.64	1.16e-6	9.22e-4	1.64e-4	8.32e-9	5.08e-9
BERT $\geq$ Octa	12 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
BERT $<$ Octa	13 (0)	25 (25)	25 (17)	25 (8)	25 (25)	25 (25)

Table 9: Comparing accuracy of Octa model variants on SemEval-15. Each experiment is repeated five times with different random seeds. ", sd " indicates one standard deviation. The p-value column indicates if the accuracy of the variant is significantly different from Octa, measured by two sample t-test. We compare each of the 25 combination of experiments between the variants and Octa. The "variant $\geq$ Octa" column counts the number of combinations where the variant is better than Octa, and how many of them are significant, measured by McNemar test. Similarly, "variant $<$ Octa" counts the number of combinations where the variant is worse than Octa. For example, Octa-Sel is better than Octa in eight combinations but none of them are significant. Octa is better than Octa-Sel in 17 combinations where nine of them are significant.

					McNemar significance test	
Model	Aspect	Target	Accuracy	p-value	variant $\geq$ Octa	variant $<$ Octa
Octa-Asp	Yes	–	84.09, sd 0.68	2.23e-8	0 (0)	25 (14)
Octa-Sel	–	Selective	85.16, sd 1.24	4.65e-6	8 (0)	17 (9)
Octa-Asp-Full	Yes	Self	82.01, sd 1.95	5.72e-6	0 (0)	25 (14)
Octa	**Yes**	**Selective**	**86.27, sd 0.57**	–	–	–