Enhancing Aspect-level Sentiment Analysis with Word Dependencies

Yuanhe Tian[♥*], **Guimin Chen**[♡*], **Yan Song**[♠♡†]
[♥]University of Washington [♡]Shenzhen Research Institute of Big Data
[♠]The Chinese University of Hong Kong (Shenzhen)
[♥]yhtian@uw.edu [♡]chenguimin@sribd.cn [♠]songyan@cuhk.edu.cn

Abstract

Aspect-level sentiment analysis (ASA) has received much attention in recent years. Most existing approaches tried to leverage syntactic information, such as the dependency parsing results of the input text, to improve sentiment analysis on different aspects. Although these approaches achieved satisfying results, their main focus is to leverage the dependency arcs among words where the dependency type information is omitted; and they model different dependencies equally where the noisy dependency results may hurt model performance. In this paper, we propose an approach to enhance aspect-level sentiment analysis with word dependencies, where the type information is modeled by key-value memory networks and different dependency results are selectively leveraged. Experimental results on five benchmark datasets demonstrate the effectiveness of our approach, where it outperforms baseline models on all datasets and achieves state-of-the-art performance on three of them.[1]

1 Introduction

Aspect-level sentiment analysis (ASA) determines the sentiment polarity of a given input text on the fine-grained level, where the sentiment towards a particular aspect in the text is predicted instead of the entire input. E.g., the sentiment of an aspect "*bar service*" in the sentence "*Total environment is fantastic although bar service is poor.*" is negative, although the text as a whole conveys a positive sentiment polarity. Due to its high practical value in many scenarios, e.g., product review analysis, social media tracking, etc., ASA attracts much attention in the natural language processing (NLP) community for years (Tang et al., 2016a,b; He et al., 2018a; Sun et al., 2019; Zhang et al., 2019; Song et al., 2019; Huang and Carley, 2019).

In recent studies, neural networks, especially recurrent models with attention mechanism, are widely applied in this task, where many of them (Wang et al., 2016; Tang et al., 2016a; Chen et al., 2017; Ma et al., 2017; Fan et al., 2018; Liang et al., 2019; Tang et al., 2020) model semantic relatedness between context and aspect words to facilitate sentiment analysis on aspects. There are other approaches using additional inputs such as word position (Gu et al., 2018), document information (He et al., 2018b; Li et al., 2018a), commonsense knowledge (Ma et al., 2018). Among all such inputs, dependency results of the input text are proved to be a kind of useful information (He et al., 2018a; Sun et al., 2019; Huang and Carley, 2019; Zhang et al., 2019; Wang et al., 2020; Tang et al., 2020), because they can help the model locate important content that modifies the aspect words and thus further suggests the sentiment towards the aspect words. Previous approaches with attention mechanism (He et al., 2018a; Wang et al., 2020), graph neural networks (GNN) (Sun et al., 2019; Huang and Carley, 2019; Zhang et al., 2019; Wang et al., 2020) and transformer (Tang et al., 2020) are applied in leveraging such information. However, most of them mainly focus on using the dependencies among words and omit to leverage other information such as relation types, which could provide useful cues to predict the sentiment. Also, they model all dependency information instances equally without weighting them according to their contribution to the task, where noisy information from the auto-generated dependency tree may hurt model performance. Therefore, improved methods are expected to comprehensively and efficiently learn dependencies among words to enhance ASA.

To address the aforementioned limitations, in this paper we propose an effective and efficient neural approach to ASA with incorporating word dependencies, which is acquired from off-the-shelf toolkits and modeled by key-value memory net-

[*]Equal contribution.
[†]Corresponding author.
[1]The code and different models are released at https://github.com/cuhksz-nlp/ASA-WD.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 3726–3739
April 19 - 23, 2021.

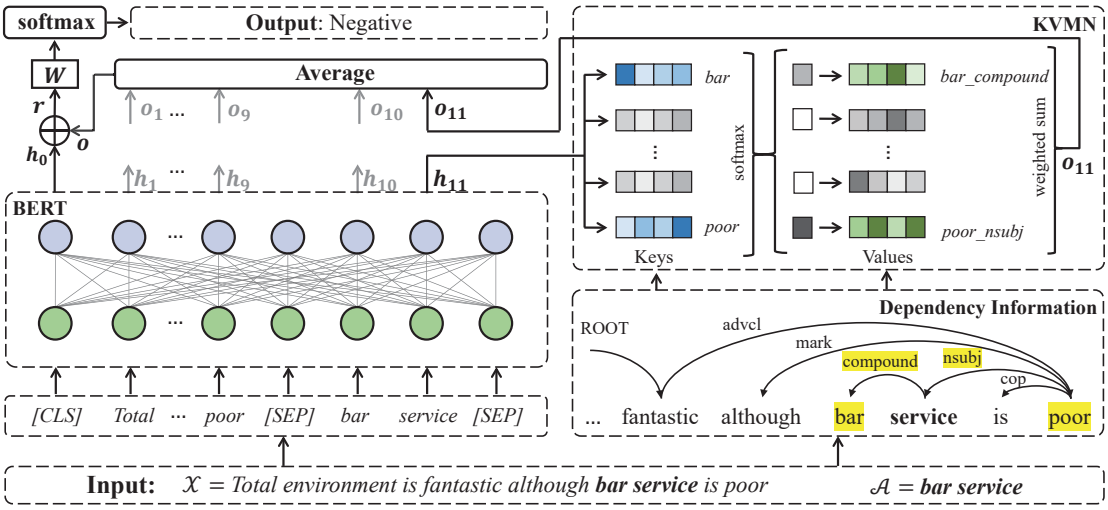

Figure 1: The overall architecture of the proposed model. The left part illustrates the backbone encoder (BERT) and decoder for ASA; the right part demonstrates the key-value memory networks (KVMN) for dependency information incorporation, where we use example word dependencies and their types (highlighted in yellow) of the aspect term "*service*" to show that how they are extracted, weighted and then fed into the left part for ASA.

works (KVMN) (Miller et al., 2016). In detail, for each input text parsed by a dependency parser, we extract its dependency relations and feed them into the KVMN, in which word-word associations and their corresponding dependency types are mapped to keys and values, respectively. Then the KVMN learns and weights different dependency knowledge according to the contribution of their corresponding keys to the ASA task, and provides the resulted representations to a regular ASA model, i.e., a BERT-based classifier, for final aspect-level sentiment predictions. In doing so, the proposed approach not only comprehensively leverages both word relations and their dependency types, but also effectively weights them through the memory mechanism according to their contributions to the ASA task. We evaluate the proposed approach on five benchmark datasets, where our approach outperforms the baselines on all datasets and achieves state-of-the-art on three of them.

2 The Approach

The task of ASA aims to analyze the sentiment of a text towards a specific aspect, which is formalized as a classification task performing on sentence-aspect pairs (Tang et al., 2016b; Ma et al., 2017; Xue and Li, 2018; Hazarika et al., 2018; Fan et al., 2018; Huang and Carley, 2018; Tang et al., 2019; Chen and Qian, 2019; Tan et al., 2019). In detail, each input sentence and the aspect terms in it are denoted by $\mathcal{X} = x_1, x_2, \cdots, x_n$ and $\mathcal{A} = a_1, a_2 \cdots, a_m$, respectively, where $\mathcal{A}$ is the sub-string of $\mathcal{X}$ ($\mathcal{A} \subset \mathcal{X}$), n and m refer to the word-based length of $\mathcal{X}$ and $\mathcal{A}$. Following this paradigm, we design the architecture of our approach in Figure 1, with a BERT-based (Devlin et al., 2019) encoder illustrated on the left to compute the sentence-aspect pair representation $\mathbf{r}$, and enhanced by the word dependency information obtained from the KVMN module on the right, then the result is fed into a softmax decoder to predict the text sentiment towards the aspect. Therefore, ASA through our approach can be formalized as

$$\hat{y} = \arg\max_{y \in \mathcal{T}} p(y|\mathcal{X}, \mathcal{A}, \text{KVMN}(\mathcal{X}, \mathcal{A})) \quad (1)$$

where $\mathcal{T}$ denotes the set of sentiment polarities for y and p computes the probability of predicting $y \in \mathcal{T}$ given $\mathcal{X}$ and $\mathcal{A}$. $\hat{y}$ refers to the predicted sentiment polarity type for $\mathcal{A}$ in the context of $\mathcal{X}$. In the rest of this section, we firstly describe KVMN for leveraging word dependencies, then explain how the resulted representations are integrated into the backbone sentiment classifier.

2.1 KVMN for Word Dependencies

High quality text representations always play a crucial role to obtain good model performance for different NLP tasks (Song et al., 2017; Seyler et al., 2018; Song and Shi, 2018; Song et al., 2018; Babanejad et al., 2020), where contextual features, including n-grams and syntactic information, have been demonstrated to be effective in enhancing text representation and thus leads to improvements on different models (Song et al., 2006, 2009; Song

and Xia, 2012; Song et al., 2012; Song and Xia, 2013; Dong et al., 2014; Miller et al., 2016; Seyler et al., 2018; Diao et al., 2019; Sun et al., 2019; Zhang et al., 2019; Huang and Carley, 2019; Tian et al., 2020b,c,d,e; Chen et al., 2020). Among all these features, dependency ones have been widely used, especially for ASA. To incorporate word dependencies into ASA task, there are many options, including attention mechanism (He et al., 2018a) where the information of dependency types among word pairs are omitted, and GNN and Transformer-based methods (Sun et al., 2019; Zhang et al., 2019; Wang et al., 2020; Tang et al., 2020) that require complicated architectures to model the entire dependency structure of an input text. Compared to these options, KVMN, whose variants have been demonstrated to be effective in incorporating contextual features (Miller et al., 2016; Guan et al., 2019; Song et al., 2020; Tian et al., 2020a,f; Nie et al., 2020), not only provides an appropriate way to leverage both word-word relations as well as their corresponding dependency types, but also weights different dependency information according to their contribution to the ASA task.

In detail, to build the KVMN, we firstly collect all word-word relations extracted from the parse results of a corpus via an off-the-shelf toolkit and use them to form the key set, and map their corresponding dependency types to the value set. Then, two embedding matrices, $\mathbf{K}$ and $\mathbf{V}$ are applied to the key and value sets with each vector representing a key or a value in the sets. At training or prediction stage, given an input text, our model obtains its dependency parsing result, i.e., for each w_i in a sentence-aspect pair, where w_i comes from $\mathcal{X}$, $\mathcal{A}$, or both $\mathcal{X}$ and $\mathcal{A}$, we extract words associated with w_i and their corresponding dependency types from the parse results. Note that, for each word, we use its inbound and outbound dependency types to represent its governor and dependent word, respectively. Therefore, for example, as illustrated in Figure 1, the words associated to the aspect word "*service*" are "*poor*" (governor) and "*bar*" (dependent); their corresponding dependency types are thus "*nsubj*" and "*compound*", respectively. Afterwards, we map the associated words and their corresponding dependency types to keys $\mathcal{K}_i = \{k_{i,1}, k_{i,2}, \cdots, k_{i,j}, \cdots, k_{i,q}\}$ and values $\mathcal{V}_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,j}, \cdots, v_{i,q}\}$ from $\mathbf{K}$ and $\mathbf{V}$ in the KVMN, where each item in $\mathcal{K}_i$ and $\mathcal{V}_i$ has its embedding denoted by $\mathbf{e}^k_{i,j}$ and $\mathbf{e}^v_{i,j}$, respectively. Once the keys and values are placed, we take the hidden vector $\mathbf{h}_i$ for w_i from the encoder (i.e., BERT), and compute the weight assigning to each value $v_{i,j}$ by

$$p_{i,j} = \frac{exp(\mathbf{h_i} \cdot \mathbf{e}^k_{i,j})}{\sum_{j=1}^{q} exp(\mathbf{h_i} \cdot \mathbf{e}^k_{i,j})} \quad (2)$$

We thus use $p_{i,j}$ to activate the corresponding values $v_{i,j}$ and compute the weighted sum by

$$\mathbf{o}_i = \sum_{j=1}^{q} p_{i,j} \mathbf{e}^v_{i,j} \quad (3)$$

where $\mathbf{o}_i$ refers to the output of the KVMN model for w_i and carries its word dependency information.

2.2 Word Dependency Integration for ASA

As shown in Figure 1, the entire model starts from encoding the input text. For the aforementioned sentence-aspect pair for ASA, it is normally organized by concatenating $\mathcal{X}$ and $\mathcal{A}$ to form a special sequence of $[[CLS], \mathcal{X}, [SEP], \mathcal{A}, [SEP]]$, and then feed it into an encoder, i.e., BERT, to obtain the hidden vectors by

$$[\mathbf{h}_0, \mathbf{H}^{\mathcal{X}}, \mathbf{H}^{\mathcal{A}}] = BERT(\mathcal{X}, \mathcal{A}) \quad (4)$$

where $\mathbf{h}_0$ denotes the hidden vector for the text-initial symbol $[CLS]$, and $\mathbf{H}^{\mathcal{X}}$, $\mathbf{H}^{\mathcal{A}}$ the embedding matrices of words in $\mathcal{X}$ and $\mathcal{A}$, respectively.

Upon the modeling of word dependencies for each w_i, different $\mathbf{o}_i$ are obtained and averaged, then concatenated with $\mathbf{h}_0$ by

$$\mathbf{r} = \mathbf{h}_0 \oplus \frac{1}{l} \cdot \sum_{i=1}^{l} \mathbf{o}_i \quad (5)$$

where $\mathbf{r}$ is the representation for the input sentence-aspect pair enhanced by word dependencies, and the value of l equals to n, m, or $n+m$ if all w_i come from $\mathcal{X}$ only, $\mathcal{A}$ only, or $\mathcal{X}+\mathcal{A}$, respectively.[2] Then, we use a dense layer with a trainable matrix $\mathbf{W}$ and vector $\mathbf{b}$ to align $\mathbf{r}$'s dimension to the output space by $\mathbf{u} = \mathbf{W} \cdot \mathbf{r} + \mathbf{b}$, with each dimension of $\mathbf{u}$ corresponding to a sentiment type. Finally, a softmax function is applied to $\mathbf{u}$ to predict the output sentiment $\hat{y}$ for the aspect $\mathcal{A}$ in $\mathcal{X}$:

$$\hat{y} = \arg\max \frac{exp(u^t)}{\sum_{t=1}^{|\mathcal{T}|} exp(u^t)} \quad (6)$$

where u^t is the value at dimension t in $\mathbf{u}$.

[2]Figure 1 illustrate the case that w_i comes from $\mathcal{X}+\mathcal{A}$, where $i \in [1, 11]$ for all $\mathbf{h}_i$.

[3]For all datasets, the sum of aspect samples under three sentiment polarities is larger than the total sentence numbers,

	LAP14		REST14		REST15		REST16		TWITTER	
	TRAIN	TEST	TRAIN	TEST	TRAIN	TEST	TRAIN	TEST	TRAIN	TEST
POSITIVE #	994	341	2,164	728	907	326	1,229	469	1,561	173
NEUTRAL #	464	169	637	196	36	34	69	30	3,127	346
NEGATIVE #	870	128	807	182	254	207	437	114	1,560	173
SENTENCE #	1,572	430	2,054	625	863	408	1,271	432	6,242	692
DIFF. #	147	36	301	76	39	34	72	18	0	0
DIFF. %	9.35	8.37	14.65	12.16	4.52	8.33	5.66	4.17	0	0

Table 1: The statistics of the five benchmark datasets, where the number of aspects on three sentiment polarities and sentences are reported.[3] We also report the number and percentage of the contrastive cases (DIFF.) where in a sentence the sentiments on aspect(s) are different from the entire sentence.

3 Experimental Settings

3.1 Datasets

Five benchmark datasets, i.e., LAP14 and REST14 (Pontiki et al., 2014), REST15 (Pontiki et al., 2015), REST16 (Pontiki et al., 2016), TWITTER (Dong et al., 2014), are used in our experiments. Specifically, LAP14 is a dataset consists of laptop computer reviews; REST14, REST15, and REST16 consist of restaurant reviews from online users; TWITTER includes tweets collected by querying the Twitter API. For all datasets, we use their official train/test splits and follow Tang et al. (2016b) to clean them by filtering out the aspects with the conflict label[4] as well as the sentences without an aspect. The statistics of the processed five datasets are reported in Table 1, where the numbers of aspects with positive, neutral, and negative polarities are reported. Note that in some datasets, e.g., LAP14 and REST14, there are rather high percentages of sentences (e.g., the sentence in Figure 1) that contain different sentiments towards aspects, as shown in the DIFF. rows in Table 1, which indicates a bigger challenge on ASA comparing to sentiment analysis on an entire sentence.

3.2 Word Dependency Extraction

Similar to previous studies (Wang et al., 2020; Tang et al., 2020) that also require dependency information, we employ the English version of SAPar[5] (Tian et al., 2020e), which is the most effective constituency parser trained on English Pen TreeBank (PTB) (Marcus et al., 1993), to obtain the constituency trees of the input text and then convert them into dependency trees by Stanford converter[6]. Therefore, when a dependency tree is built on the entire input text, for each word in the text, one can find its dependent words and types according to the dependency paths on the tree. Consequently, the dependency relations of each word to others can be extended along with the dependency paths and it is not restricted that only one-hop (first-order) relations can be considered in our model. One could easily extend the coverage of word dependencies with two- or three-hop relations from a given word, which are known as second- and third-order dependencies, e.g., "*poor* $\rightarrow$ *service* $\rightarrow$ *bar*" in Figure 1 is a second-order dependency relation.

As described in §2.1, extracting first-order word dependencies is straightforward; to extend it with higher order ones, we follow the same principle to extract word dependencies and assign dependency types as follows: (1) for the governor w_g of the target word w, we collect all its governor and dependents (except for w) associated with w_g's inbound and outbound dependency types, respectively; (2) for each dependent w_d of w, we find all dependents of w_d and use outbound dependency types to represent w_d's dependent words; (3) we include all context words and their corresponding dependency types collected in (1) and (2) as the input to KVMN for w and repeat the process for further higher order word dependencies.

For example, in the input text in Figure 1, the second-order word dependencies and types for "*service*" are started from its governor "*poor*" and dependent "*bar*". Then for "*poor*", we collect its governor "*fantastic*" with an inbound dependency type of "*advcl*", and dependents "*although*" and "*is*" with the outbound dependency types of "*mark*" and "*cop*", respectively. For "*bar*", it is not able to expand because it has no dependent,

because that many sentences have more than one aspect and such aspects usually have contrastive sentiment polarities.

[4]The "*conflict*" label is used in LAP14, REST14/16 to identify aspects that have conflict sentiment polarities.

[5]https://github.com/cuhksz-nlp/SAPar

[6]We use the converter of version 3.3.0 from https://stanfordnlp.github.io/CoreNLP/index.html.

Models	Lap14		Rest14		Rest15		Rest16		Twitter	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
BERT-base	77.90	73.30	84.11	76.66	83.02	67.92	89.38	64.98	73.27	71.52
+ $\mathcal{X}$-KVMN	78.37	74.18	84.46	**78.44**	84.14	66.12	90.36	72.77	74.13	72.16
+ $\mathcal{A}$-KVMN	**79.78**	**76.14**	**85.98**	77.94	**84.14**	**68.49**	**90.52**	**73.15**	**75.14**	**73.68**
+ $\mathcal{XA}$-KVMN	78.53	75.00	85.09	78.32	83.77	66.57	90.36	72.20	74.13	73.11
BERT-large	78.68	73.75	85.17	77.94	83.21	70.55	90.52	72.88	74.13	73.04
+ $\mathcal{X}$-KVMN	79.31	75.58	86.34	79.63	84.14	70.93	92.13	77.15	74.56	73.07
+ $\mathcal{A}$-KVMN	**80.41**	**77.38**	**86.88**	**80.92**	**84.70**	**72.71**	**92.48**	**79.54**	**76.59**	**74.91**
+ $\mathcal{XA}$-KVMN	80.16	77.20	86.70	79.95	84.58	71.05	91.83	77.28	74.57	72.76

Table 2: Experimental results (accuracy and F1 scores) of using different encoders (BERT-base and BERT-large) with and without KVMN on five benchmark datasets, where $\mathcal{X}$, $\mathcal{A}$, and $\mathcal{XA}$ refer to that KVMN models word dependencies from $\mathcal{X}$ only, $\mathcal{A}$ only, and $\mathcal{X} + \mathcal{A}$, respectively.

the collection thus stops here. Therefore, the resulted words (keys) in second-order dependencies and their corresponding dependency types (values) for "*service*" are $\mathcal{K}_{11} = \{bar, poor, fantastic, although, mark\}$, and $\mathcal{V}_{11} = \{bar\_compound, poor\_nsubj, fantastic\_advcl, although\_mark, is\_cop\}$, respectively.

3.3 Implementation Details

We adopt BERT-base-uncased and BERT-large-uncased[7] as the encoders in our approach, which are demonstrated to be the most effective encoders for many NLP tasks (Straková et al., 2019; Baldini Soares et al., 2019; Xu et al., 2019). In our experiments, we use their default settings for the two BERT encoders (i.e., for BERT-base-uncased, we use 12 layers with 768 dimensional hidden vectors; and for BERT-large-uncased, we use 24 layers with 1024 dimensional hidden vectors). For all experiments, we use Adam optimizer (Kingma and Ba, 2014) and try different combinations of learning rates, dropout rates, and batch size.[8] In addition, we apply Xavier initialization (Glorot and Bengio, 2010) on all trainable parameters including the embeddings for keys and values in the KVMN. Moreover, we use the cross-entropy loss function to optimize our model and follow the convention to evaluate our models via accuracy and macro-averaged F1 scores over all sentiment polarities, i.e., positive, neutral and negative.

[7]We obtain the BERT models from https://github.com/huggingface/pytorch-pretrained-BERT.

[8]We report the hyper-parameter settings of different models, as well as their size and running speed, in the Appendix.

4 Experimental Results

4.1 Effect of Using Word Dependencies

In the main experiments, we test our model with and without integrating word dependencies by KVMN, where both the base and large BERT encoders are used. In detail, when leveraging word dependencies, we run experiments on our proposed model to explore the effect of learning from different parts of the input, i.e., we try word dependencies from three sources: $\mathcal{X}$ only, $\mathcal{A}$ only, and both $\mathcal{X}$ and $\mathcal{A}$ (see §2.2). Experimental results are reported in Table 2 with the prefixes of KVMN denoting which part is encoded from.

There are several observations. First, KVMN works well with both the base and large BERT. Although BERT baselines have already achieved good performance, improvements of our proposed model over the baselines are observed on all datasets with respect to both accuracy and F1 scores. Second, among the three settings of encoding from different parts of the input (i.e., $\mathcal{X}$, $\mathcal{A}$, $\mathcal{X}$+$\mathcal{A}$) in KVMN, in most datasets (except for Twitter), the highest performance is observed on "$\mathcal{A}$-KVMN". These results comply with the intuition where extracting and learning word dependencies from $\mathcal{A}$ ensures KVMN only incorporates the information from the content directly associated with the aspect words, thus focuses the model on the words that are most likely to be helpful on ASA for a particular aspect in a sentence. Third, although the overall performance of $\mathcal{X}$-KVMN and $\mathcal{XA}$-KVMN are not as good as that of $\mathcal{A}$-KVMN, they are still better than the baselines without using word dependencies. Especially for $\mathcal{X}$-KVMN, where word dependencies are extracted from the entire sentence, in this case,

Order	Lap14			Rest14			Rest15			Rest16			Twitter		
	Acc	F1	Cvge	Acc	F1	Cvge	Acc	F1	Cvge	Acc	F1	Cvge	Acc	F1	Cvge.
1st	77.59	73.00	26.89	84.28	75.90	27.05	83.76	67.06	32.26	90.03	70.39	32.50	74.28	73.31	22.26
2nd	**79.78**	**76.14**	52.47	**85.98**	**77.94**	53.83	**84.14**	**68.49**	60.95	**90.52**	**73.15**	61.19	**75.14**	**73.68**	44.70
3rd	78.99	74.60	72.55	85.35	77.78	75.16	82.83	62.81	80.44	89.54	66.56	80.66	74.27	72.31	67.27

(a) BERT-base

Order	Lap14			Rest14			Rest15			Rest16			Twitter		
	Acc	F1	Cvge	Acc	F1	Cvge	Acc	F1	Cvge	Acc	F1	Cvge	Acc	F1	Cvge.
1st	80.25	76.74	26.89	86.43	79.55	27.05	84.33	69.47	32.26	92.12	79.09	32.50	75.43	73.45	22.26
2nd	**80.41**	**77.38**	52.47	**86.88**	80.92	53.83	**84.70**	**72.71**	60.95	**92.48**	**79.54**	61.19	**76.59**	**74.91**	44.70
3rd	80.09	76.84	72.55	86.52	**81.02**	75.16	84.14	68.05	80.44	92.27	79.20	80.66	76.16	74.85	67.27

(b) BERT-large

Table 3: Experimental results of our models with the best setting (i.e., using base and large BERT with $\mathcal{A}$-KVMN) of using dependency relations on different (i.e., 1st, 2nd and 3rd) orders. Average percentage of words in a sentence covered by word dependencies on different orders are also reported in the Cvge. column.

the dependency information also helps ASA even though it introduces some noise to the task when the entire sentence possesses a different sentiment polarity (as shown in the Diff. rows in Table 1), while such noise contributes to its inferior performance to the $\mathcal{A}$-KVMN setting. Therefore, for the case that the sentiment is agreed between the entire sentence and its aspect (e.g., Twitter dataset is in this case according to Table 1), $\mathcal{X}$-KVMN and $\mathcal{A}$-KVMN have similar performance.

4.2 Effect of Different Dependency Orders

Previous experiments showed the effectiveness of our model with KVMN on first-order word dependencies. In this experiment, we use the best setting (i.e., models using $\mathcal{A}$-KVMN) for base and large BERT and run them with encoding higher-order dependencies to further investigate the effectiveness of our model with more dependency information. Particularly, we try second- and third-order word dependencies and compare their results with the previous first-order ones. The results on all datasets, as well as average coverage (%) of words in each sentence with respect to different dependency orders,[9] are reported in Table 3, where (a) and (b) show the results of models with BERT-base and BERT-large encoders, respectively. From the results, it is found that in most cases (e.g., for both base and large BERT), models using second-order word dependencies achieve the overall highest performance, which can be explained by that first-order dependency for aspect words is not enough to cover enough salient information helping ASA. This is a common phenomenon when negation is included in a sentence. For example, in "*the pizza is not good*", for its aspect "*pizza*", whose first-order dependencies only link "*pizza*" with "*good*", the classifier is thus misled to predict a positive sentiment polarity. Compared to using second-order word dependencies, third-order dependencies in general do not provide further improvement to ASA, which owes to the reason that more irrelevant information is introduced to the encoder thus distract the model for final prediction. In fact, third-order dependencies lead to that around 75% words in each sentence are fed into KVMN, which could severely affect ASA by sentence-level sentiment polarities, and eventually harm model performance especially when an aspect-level sentiment differs from the sentence-level sentiment.

4.3 Comparison with Previous Studies

To further demonstrate the effectiveness of our approach, we compare our best-performing model, i.e., the BERT-large encoder with second-order word dependencies incorporated through $\mathcal{A}$-KVMN, with previous studies, where the comparisons on all datasets are reported in Table 4, where the results of BERT-large baseline, as well as the ones using BERT-base, are also reported for references. It is observed that, our model consistently outperforms the BERT-large baseline on all datasets and achieves state-of-the-art on three of them (i.e., Lap14, Rest15, Rest16) in terms of

[9] This metric is used to present how many words in each input sentence are involved when different orders are applied for extracting word dependencies, so as to illustrate how much information in a sentence is helpful for ASA.

Models	Lap14		Rest14		Rest15		Rest16		Twitter	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ATAE-LSTM (Wang et al., 2016)	68.70	-	77.20	-	-	-	-	-	-	-
MemNet (Tang et al., 2016b)	72.21	-	80.95	-	-	-	-	-	-	-
IAN (Ma et al., 2017)	72.10	-	78.60	-	-	-	-	-	-	-
RAM (Chen et al., 2017)	74.49	71.35	80.23	70.80	-	-	-	-	69.36	67.30
PBAN (Gu et al., 2018)	74.12	-	81.16	-	-	-	-	-	-	-
TNet-AS (Li et al., 2018b)	76.54	71.75	80.69	71.27	-	-	-	-	74.97	73.60
PRET+MULT (He et al., 2018b)	71.15	67.46	79.11	69.73	81.30	68.74	85.58	69.76	-	-
SynATT (He et al., 2018a)	72.57	69.13	80.63	71.32	81.67	66.05	64.61	67.45	-	-
PF-CNN (Huang and Carley, 2018)	70.06	-	79.20	-	-	-	-	-	-	-
MGAN (Fan et al., 2018)	75.39	72.47	81.25	71.94	-	-	-	-	72.54	70.81
CAN (Hu et al., 2019)	-	-	84.28	74.45	78.58	54.72	-	-	-	-
TransCap (Chen and Qian, 2019)	73.87	70.10	79.55	71.41	-	-	-	-	-	-
IACapsNet (Du et al., 2019)	76.80	73.29	81.79	73.40	-	-	-	-	75.01	73.81
ANTM (Mao et al., 2019)	75.84	72.49	82.49	72.10	-	-	-	-	72.35	69.45
CDT (Sun et al., 2019)	77.19	72.99	82.30	74.02	-	-	85.58	69.93	74.66	73.66
ASGCN (Zhang et al., 2019)	75.55	71.05	81.22	72.94	79.89	61.89	88.99	67.48	72.69	70.59
†TD-GAT-BERT (Huang and Carley, 2019)	80.10	-	83.00	-	-	-	-	-	-	-
†AEN-BERT (Song et al., 2019)	79.93	76.31	83.12	73.76	-	-	-	-	74.71	73.13
†BERT-PT (Xu et al., 2019)	78.07	75.08	84.95	76.96	-	-	-	-	-	-
†R-GAT-BERT (Wang et al., 2020)	78.21	74.07	86.60	**81.35**	-	-	-	-	76.15	74.88
†DGEDT-BERT (Tang et al., 2020)	79.8	75.6	86.3	80.0	84.0	71.0	91.9	79.0	**77.9**	**75.4**
BERT-base	77.90	73.30	84.11	76.66	83.02	67.92	89.38	64.98	73.27	71.52
Our Best Model (BERT-base)	*79.78	*76.14	*85.98	*77.94	*84.14	*68.49	*90.52	*73.15	*75.14	*73.68
†BERT-large	78.68	73.75	85.17	77.94	83.21	70.55	90.52	72.88	74.13	73.04
†Our Best Model (BERT-Large)	***80.41**	***77.38**	***86.88**	*80.92	***84.70**	***72.71**	***92.48**	***79.54**	*76.59	*74.91

Table 4: Performance Comparison (on accuracy and F1 scores) of our best model (BERT-large + $\mathcal{A}$-KVMN with second-order word dependencies) with previous studies on all datasets. The results of BERT-large baseline are also reported for references. Models that use BERT-large as the encoder are marked by "†". The results marked by "*" indicate that our model is significantly better than the corresponding baseline model (t-test with $p < 0.05$).

both accuracy and F1 scores. Specifically, compared with previous studies that also leverage dependency information, our approach outperforms He et al. (2018a); Sun et al. (2019); Huang and Carley (2019); Zhang et al. (2019) on all dataset and outperforms Wang et al. (2020) and Tang et al. (2020) on most datasets. This observation is valid because, in previous models, they are weighting or averaging hidden vectors of the (aspect related) words rather than on the relations, and omitting dependency types which provide guidance to emphasize some useful relations, e.g., the "*amod*" (i.e., adjectival modifier) type identifies that an adjectival modifier could be the sentiment words of a corresponding aspect. Therefore, the superiority of our model comes from two aspects, weighting word-word relations and leveraging dependency types. KVMN highlights salient dependency relations and learns from them and their dependency types, which alleviates the influence of noisy dependency information. In addition, we note that our approach achieves inferior results on Twitter dataset compared with Tang et al. (2020). One possible explanation is that a dependency parser trained in the general domain can get inferior parsing results on Twitter texts from the social media domain, which makes it harder for our approach to improve the BERT-large baseline compared to other datasets. Nevertheless, the effectiveness of our approach is still valid given that our approach outperforms Tang et al. (2020) on all other datasets.

5 Analyses

5.1 Ablation Study

To confirm the validity of using both word relations (keys) and their corresponding dependency types (values) for ASA, we conduct an ablation study by learning from either part of the two types of dependency information. We choose the models using BERT-base and BERT-large with our best setting (i.e., models with second-order word dependencies and $\mathcal{A}$-KVMN) for this study and adapt the KVMN module to key-only or value-only input. The experimental results on all benchmark datasets are reported in Table 5, where keys or values are ablated. It is clearly indicated in the table that, for models with different encoders (i.e., base and large BERT), the model performance drops on all datasets if either keys or values are excluded. Specifically, in most cases, the drop of performance (especially on accuracy) is higher when keys are ablated ("− Keys"), comparing with the ablation of values ("− Values"). This phenomenon indicates the context words, which attracts much attention from previous studies (He et al., 2018a; Sun et al., 2019; Zhang et al., 2019; Huang and Carley, 2019), play a more important role compared with their

Setting	Lap14		Rest14		Rest15		Rest16		Twitter	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Full Model	**79.78**	**76.14**	**85.98**	**77.94**	**84.14**	**68.49**	**90.52**	**73.15**	**75.14**	**73.68**
− Keys	79.47	75.29	84.91	77.33	82.65	67.95	89.54	72.21	74.41	73.32
Δ	–0.31	–0.85	–1.07	–0.61	–1.49	–0.54	–0.98	–0.94	–0.73	–0.36
− Values	79.62	75.45	85.18	77.71	83.21	68.20	89.71	72.57	74.70	73.45
Δ	–0.16	–0.69	–0.80	–0.23	–0.93	–0.29	–0.81	–0.58	–0.44	–0.23

(a) BERT-base

Setting	Lap14		Rest14		Rest15		Rest16		Twitter	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Full Model	**80.41**	**77.38**	**86.88**	**80.92**	**84.70**	**72.71**	**92.48**	**79.54**	**76.59**	**74.91**
− Keys	79.94	77.06	86.43	79.60	83.84	70.89	91.50	77.99	75.58	72.94
Δ	–0.47	–0.32	–0.45	–1.32	–0.86	–1.82	–0.98	–1.55	–1.01	–1.97
− Values	80.26	77.34	86.63	80.82	84.40	70.31	91.82	78.53	75.87	73.69
Δ	–0.15	–0.04	–0.25	–0.10	–0.30	–2.04	–0.66	–1.01	–0.72	–1.22

(b) BERT-large

Table 5: Results on five datasets from our full models (base and large BERT with $\mathcal{A}$-KVMN and second-order word dependencies) and its variants where keys ("− Keys") and values ("− Values") are ablated. Δ refers to the drop of accuracy and F1 score when keys or values are excluded from the full model.

dependency types. Still, one cannot deny the contribution of dependency types because the drop is still significant if values are excluded, where even on some datasets (e.g., Rest15 and Rest16) higher drops are observed on F1 than Key ablation. The results for this ablation study demonstrate that dependency type is of high importance to improve ASA if they are appropriately encoded.

5.2 Case Study

To illustrate the effect of KVMN module on weighting salient word dependencies and thus improve ASA, we conduct a case study on the sentence "*The falafel was rather overcooked and dried but the chicken was fine*" shown in Figure 2, in which it contains two aspects with contrast sentiment polarities, i.e., *negative* towards "*falafel*" and *positive* towards "*chicken*". For each aspect, we run our best model (BERT-large + $\mathcal{A}$-KVMN with second-order word dependencies), and visualize the weights ($p_{i,j}$ in Eq. (2)) assigned to all associated dependency types and their corresponding words, where darker color refers to higher weights.

For the first aspect "*falafel*" (Figure 2(a)), although there are some adjectives carrying opposite sentiment polarities within its second-order relations, KVMN successfully distinguishes "*overcooked*" is more important to it and assigns a relatively higher weight. This is because that the corresponding type ("*nsubjpass*", passive nominal subject) to "*overcooked*" is intensively highlighted

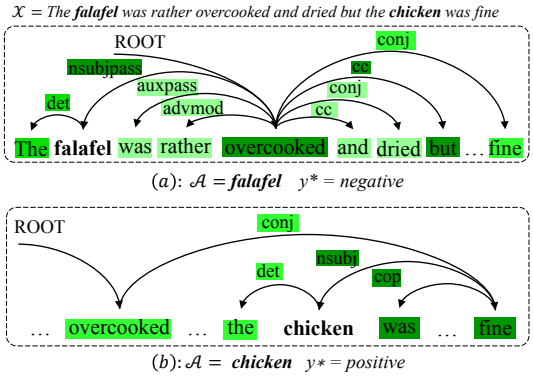

Figure 2: Illustration of an example sentence with two aspects in different sentiment polarities. For each aspect, weights (from our best model) assigned to dependent words and dependency types are visualized with colors, where darker color refers to higher weights.

so that the model identifies it as the main sentiment carrier for the aspect word "*falafel*" where other adjectives (i.e., "*dried*" and "*fine*") share the "*conj*" (conjunction) type and are distantly related to the aspect words, making them less important.

For the other aspect "*chicken*" (Figure 2(b)), similar to the first one, both "*overcooked*" and "*fine*" are included in its associated context words. In this case, "*fine*" is more closely dependent on "*chicken*" than "*overcooked*", where it has a "*nsubj*" (noun subject) type showing a predicate role thus receives higher weight from KVMN, resulting in a positive sentiment polarity prediction towards "*chicken*". Overall, this case study perfectly explains the ef-

fectiveness of our model, where two aspects share the same context and the only change is the dependency information ($\mathbf{o_i}$) comes from KVMN. Therefore, the different prediction results for the two aspects suggest that KVMN appropriately learns from salient dependency relations and types for each aspect, where different types have their own capabilities to enhance ASA accordingly (e.g., "*nsubj*" may contribute more than "*conj*").

6 Related Work

Different from sentiment analysis for large granular texts, such as document and sentences, ASA focuses on processing sentiment polarities for a specific aspect (e.g., "*pizza*") or category (e.g., "*food*") in a piece of text. To address this task, early approaches (Jiang et al., 2011; Dong et al., 2014) followed the sentence classification paradigm and recent studies enhanced it as a mission of sentence-aspect pair classification with applying neural approaches (Wang et al., 2016; Tang et al., 2016a; Ma et al., 2017; Chen et al., 2017; Xue and Li, 2018; Li et al., 2018b; Hu et al., 2019; Xu et al., 2019) such as recurrent models (e.g., bi-LSTM) and pre-trained encoders (e.g., BERT) for effectively capturing contextual information. In addition to improving the input form, advanced models such as memory networks (Tang et al., 2016b; Chen et al., 2017; Wang et al., 2018; Zhu and Qian, 2018; Mao et al., 2019), attention mechanism (Wang et al., 2016; Ma et al., 2017; Hazarika et al., 2018), capsule networks (Du et al., 2019; Chen and Qian, 2019; Jiang et al., 2019), GNN (Huang and Carley, 2019; Sun et al., 2019; Zhang et al., 2019; Wang et al., 2020), and transformer (Tang et al., 2020) are applied to this task, with other studies leveraging external resources, including position information (Gu et al., 2018), document information (He et al., 2018b), commonsense knowledge (Ma et al., 2018), etc. Among all resources, syntactic information was proved to be the most effective one and successfully adopted in recent studies with GNN (Huang and Carley, 2019; Sun et al., 2019; Zhang et al., 2019). Compared with previous studies, our approach offers an alternative way to use KVMN and syntactic information for ASA. Consider those studies using memory networks where their memories are represented by contextual features of the aspect terms, dependency information was not leveraged in their work. In addition, compared with those approaches leveraging word dependencies (i.e., using attention mechanism or GNN), where they not only omitted useful dependency information such as relation types, but also demanded a complicated model structure in doing so, our approach ensures comprehensively encoding from both word-word relations and their dependency types, and models them in an efficient way by KVMN.

7 Conclusion

In this paper, we propose an effective neural approach to improve ASA with word dependencies by KVMN, where for each aspect term, we firstly extract the words associated to it according to the dependency parse of the input sentence and their corresponding dependency relation types, then use KVMN to encode and weight such information to enhance ASA accordingly. In our approach, not only word-word relations but also their dependency types are leveraged in a KVMN, which to our best knowledge are the first attempts in all related syntax-driven studies for ASA. Experimental results on five widely used benchmark datasets demonstrate the effectiveness of our approach, and shows that second-order word dependency is the best choice for ASA, where the new state-of-the-art results are achieved on three datasets. Moreover, further analyses illustrate the validity of applying KVMN on both dependency relation and type information, especially the effectiveness of dependency types, which are often omitted in previous studies.

Acknowledgements

This work is supported by The Chinese University of Hong Kong (Shenzhen) under University Development Fund UDF01001809. This work is also partially supported by NSFC under the project "The Essential Algorithms and Technologies for Standardized Analytics of Clinical Texts" (12026610).

References

Nastaran Babanejad, Ameeta Agrawal, Aijun An, and Manos Papagelis. 2020. A Comprehensive Analysis of Preprocessing for Word Representation Learning in Affective Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5799–5810, Online.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

Guimin Chen, Yuanhe Tian, and Yan Song. 2020. Joint Aspect Extraction and Sentiment Analysis with Directional Graph Convolutional Networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 272–279, Barcelona, Spain (Online).

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 452–461.

Zhuang Chen and Tieyun Qian. 2019. Transfer capsule network for aspect level sentiment classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 547–556.

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2019. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. *ArXiv*, abs/1911.00720.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 49–54.

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 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5492–5501.

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

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Shuqin Gu, Lipeng Zhang, Yuexian Hou, and Yin Song. 2018. A position-aware bidirectional attention network for aspect-level sentiment analysis. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 774–784.

Chaoyu Guan, Yuhao Cheng, and Hai Zhao. 2019. Semantic Role Labeling with Associated Memory Network. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3361–3371, Minneapolis, Minnesota.

Devamanyu Hazarika, Soujanya Poria, Prateek Vij, Gangeshwar Krishnamurthy, Erik Cambria, and Roger Zimmermann. 2018. Modeling inter-aspect dependencies for aspect-based sentiment analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 266–270.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018a. Effective attention modeling for aspect-level sentiment classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1121–1131.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018b. Exploiting document knowledge for aspect-level sentiment classification. *arXiv preprint arXiv:1806.04346*.

Mengting Hu, Shiwan Zhao, Li Zhang, Keke Cai, Zhong Su, Renhong Cheng, and Xiaowei Shen. 2019. CAN: Constrained Attention Networks for Multi-Aspect Sentiment Analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4593–4602.

Binxuan Huang and Kathleen M Carley. 2018. Parameterized Convolutional Neural Networks for Aspect Level Sentiment Classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1091–1096.

Binxuan Huang and Kathleen M Carley. 2019. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5472–5480.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6281–6286.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Junjie Li, Haitong Yang, and Chengqing Zong. 2018a. Document-level multi-aspect sentiment classification by jointly modeling users, aspects, and overall ratings. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 925–936.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018b. Transformation Networks for Target-Oriented Sentiment Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 946–956.

Bin Liang, Jiachen Du, Ruifeng Xu, Binyang Li, and Hejiao Huang. 2019. Context-aware Embedding for Targeted Aspect-based Sentiment Analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4678–4683, Florence, Italy.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4068–4074.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Qianren Mao, Jianxin Li, Senzhang Wang, Yuanning Zhang, Hao Peng, Min He, and Lihong Wang. 2019. Aspect-based sentiment classification with attentive neural turing machines. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5139–5145.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409.

Yuyang Nie, Yuanhe Tian, Yan Song, Xiang Ao, and Xiang Wan. 2020. Improving Named Entity Recognition with Attentive Ensemble of Syntactic Information. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4231–4245, Online.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Dominic Seyler, Tatiana Dembelova, Luciano Del Corro, Johannes Hoffart, and Gerhard Weikum. 2018. A Study of the Importance of External Knowledge in the Named Entity Recognition Task. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 241–246.

Yan Song, Jiaqing Guo, and Dongfeng Cai. 2006. Chinese Word Segmentation Based on an Approach of Maximum Entropy Modeling. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 201–204, Sydney, Australia.

Yan Song, Chunyu Kit, and Xiao Chen. 2009. Transliteration of Name Entity via Improved Statistical Translation on Character Sequences. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 57–60, Suntec, Singapore.

Yan Song, Prescott Klassen, Fei Xia, and Chunyu Kit. 2012. Entropy-based Training Data Selection for Domain Adaptation. In *Proceedings of COLING 2012: Posters*, pages 1191–1200, Mumbai, India.

Yan Song, Chia-Jung Lee, and Fei Xia. 2017. Learning Word Representations with Regularization from Prior Knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152.

Yan Song and Shuming Shi. 2018. Complementary Learning of Word Embeddings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4368–4374.

Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional Skip-Gram: Explicitly Distinguishing Left and Right Context for Word Embeddings. In *Proceedings of the 2018 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 175–180, New Orleans, Louisiana.

Yan Song, Yuanhe Tian, Nan Wang, and Fei Xia. 2020. Summarizing Medical Conversations via Identifying Important Utterances. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 717–729, Barcelona, Spain (Online).

Yan Song and Fei Xia. 2012. Using a Goodness Measurement for Domain Adaptation: A Case Study on Chinese Word Segmentation. In *LREC*, pages 3853–3860.

Yan Song and Fei Xia. 2013. A Common Case of Jekyll and Hyde: The Synergistic Effect of Using Divided Source Training Data for Feature Augmentation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 623–631, Nagoya, Japan.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *arXiv preprint arXiv:1902.09314*.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5683–5692.

Xingwei Tan, Yi Cai, and Changxi Zhu. 2019. Recognizing Conflict Opinions in Aspect-level Sentiment Classification with Dual Attention Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3417–3422.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective LSTMs for Target-Dependent Sentiment Classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect Level Sentiment Classification with Deep Memory Network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency Graph Enhanced Dual-transformer Structure for Aspect-based Sentiment Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6578–6588, Online.

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 57th Annual Meeting of the Association for Computational Linguistics*, pages 557–566.

Yuanhe Tian, Wang Shen, Yan Song, Fei Xia, Min He, and Kenli Li. 2020a. Improving Biomedical Named Entity Recognition with Syntactic Information. *BMC Bioinformatics*, 21:1471–2105.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020b. Joint Chinese Word Segmentation and Part-of-speech Tagging via Two-way Attentions of Auto-analyzed Knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8286–8296, Online.

Yuanhe Tian, Yan Song, and Fei Xia. 2020c. Joint Chinese Word Segmentation and Part-of-speech Tagging via Multi-channel Attention of Character N-grams. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2073–2084, Barcelona, Spain (Online).

Yuanhe Tian, Yan Song, and Fei Xia. 2020d. Supertagging Combinatory Categorial Grammar with Attentive Graph Convolutional Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6037–6044, Online.

Yuanhe Tian, Yan Song, Fei Xia, and Tong Zhang. 2020e. Improving Constituency Parsing with Span Attention. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1691–1703, Online.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020f. Improving Chinese Word Segmentation with Wordhood Memory Networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285, Online.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online.

Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018. Target-sensitive memory networks for aspect sentiment classification. In

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 957–967.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335.

Wei Xue and Tao Li. 2018. Aspect Based Sentiment Analysis with Gated Convolutional Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4560–4570.

Peisong Zhu and Tieyun Qian. 2018. Enhanced Aspect Level Sentiment Classification with Auxiliary Memory. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1077–1087.

Appendix

A. Model Size and Running Speed

Table 6 reports the number of trainable parameters and inference speed (sentences/second)[10] of baseline (i.e., the ones without using KVMN and the dependency information) and our best performing models (i.e., the ones with $\mathcal{A}$-KVMN and the second-order dependencies) on all datasets.

B. Hyper-parameter Settings

Table 7 reports the hyper-parameters we used for tuning our models. For each dataset, we try all combinations of the hyper-parameters and report the one with the highest accuracy score in our paper.

[10]The test is performed on a Quadro RTX 6000 GPU.

MODELS	LAP14		REST14		REST15		REST16		TWITTER	
	PARA.	SPEED	PARA.	SPEED	PARA.	SPEED	PARA.	SPEED	PARA.	SPEED
BERT-BASE	109.5M	37.1	109.5M	38.1	109.5M	37.3	109.5M	38.5	109.5M	38.2
FULL MODEL	125.2M	34.6	125.2M	30.6	125.2M	34.5	125.2M	34.6	147.2M	34.3
BERT-LARGE	335.1M	20.0	335.1M	20.1	335.1M	20.5	335.1M	20.5	335.1M	19.6
FULL MODEL	356.1M	19.6	356.1M	19.0	356.1M	19.2	356.1M	19.6	385.4M	19.8

Table 6: The number of trainable parameters (PARA.) and the running speed (sentences/second) on the test sets of the baseline models (the ones without using KVMN and the dependency information) and our best performing models (the ones with $\mathcal{A}$-KVMN and the second-order dependencies).

HYPER-PARAMETER TYPES	TRIED HYPER-PARAMETER VALUES
LEARNING RATE	$e^{-5}, 2e^{-5}, 3e^{-5}, 4e^{-5}, 5e^{-5}, 6e^{-5}, 7e^{-5}, 8e^{-5}, 9e^{-5}, e^{-4}$
DROPOUT RATE	$0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$
BATCH SIZE	$8, 16, 32$

Table 7: The hyper-parameters for tuning our models.