Cross-Domain Aspect Extraction using Transformers Augmented with Knowledge Graphs

Phillip Howard
phillip.r.howard@intel.com
Intel Labs
Chandler, Arizona, USA

Arden Ma
Intel Labs
Santa Clara, California, USA

Vasudev Lal
Intel Labs
Hillsboro, Oregon, USA

Ana Paula Simoes
Intel Labs
Santa Clara, California, USA

Daniel Korat
Intel Labs
Petah Tikva, Israel

Oren Pereg
Intel Labs
Petah Tikva, Israel

Moshe Wasserblat
Intel Labs
Petah Tikva, Israel

Gadi Singer
Intel Labs
Santa Clara, California, USA

ABSTRACT

The extraction of aspect terms is a critical step in fine-grained sentiment analysis of text. Existing approaches for this task have yielded impressive results when the training and testing data are from the same domain. However, these methods show a drastic decrease in performance when applied to cross-domain settings where the domain of the testing data differs from that of the training data. To address this lack of extensibility and robustness, we propose a novel approach for automatically constructing domain-specific knowledge graphs that contain information relevant to the identification of aspect terms. We introduce a methodology for injecting information from these knowledge graphs into Transformer models, including two alternative mechanisms for knowledge insertion: via query enrichment and via manipulation of attention patterns. We demonstrate state-of-the-art performance on benchmark datasets for cross-domain aspect term extraction using our approach and investigate how the amount of external knowledge available to the Transformer impacts model performance.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; *Supervised learning*; *Neural networks*.

KEYWORDS

Knowledge graphs, transformers, aspect extraction, knowledge injection, aspect-based sentiment analysis

ACM Reference Format:
Phillip Howard, Arden Ma, Vasudev Lal, Ana Paula Simoes, Daniel Korat, Oren Pereg, Moshe Wasserblat, and Gadi Singer. 2022. Cross-Domain Aspect Extraction using Transformers Augmented with Knowledge Graphs. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557275

CIKM '22, October 17–21, 2022, Atlanta, GA, USA.

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557275

1 INTRODUCTION

Sentiment analysis is a fundamental task in NLP which has been widely studied in a variety of different settings. While the majority of existing research has focused on sentence- and document-level sentiment extraction, there is considerable interest in fine-grained sentiment analysis that seeks to understand sentiment at a word or phrase level. For example, in the sentence *"The appetizer was delicious"*, it may be of interest to understand the author's sentiment regarding a specific aspect (*appetizer)* in the form of an expressed opinion (*delicious*). This task is commonly referred to as Aspect-Based Sentiment Analysis (ABSA).

ABSA is often formulated as a sequence tagging problem, where the input to a model is a sequence of tokens $X = \{x_1, x_2, ..., x_n\}$. For each token $x_i \in X$, the objective is to correctly predict a label $y_i \in \{BA, IA, BO, IO, N\}$. The labels *BA* and *IA* denote the beginning and inside tokens of aspect phrases while *BO* and *IO* indicate the beginning and inside tokens of opinions. The class *N* denotes tokens that are neither aspects nor opinions. The focus of our work is improving the identification of aspects within the context of the ABSA sequence tagging problem.

Existing work on aspect term extraction has achieved promising results in single-domain settings where both the training and testing data arise from the same distribution. However, such methods typically perform much worse when the training (or *source*) domain differs from the testing (or *target*) domain. This cross-domain setting for aspect extraction poses a greater challenge because there is often very little overlap between aspects used in different domains. For example, aspects prevalent in consumer reviews about laptops (e.g., *processor*, *hardware*) are unrelated to common aspects in restaurant reviews (e.g., *food*, *appetizer*).

To address this challenging task, we introduce a novel method for enhancing pretrained Transformer models [35] with information from domain-specific knowledge graphs that are automatically constructed from semantic knowledge sources. We show how injecting

information from these knowledge graphs into Transformer models improves domain transfer by providing contextual information about potential aspects in the target domain.

This work consists of four primary contributions. First, we introduce an approach for constructing domain-specific knowledge graphs from unlabeled text using an existing large-scale commonsense knowledge graph (ConceptNet, Speer et al. [33]) and a Transformer -based generative knowledge source fine-tuned for the task of predicting relations within a domain (COMET, Bosselut et al. [2]). Second, we present a methodology for determining when it is beneficial to inject external knowledge into a Transformer model for aspect extraction through the application of syntactic information. Third, we explore two alternative approaches for injecting knowledge into language models: via insertion of a pivot token for query enrichment and through a disentangled attention mechanism. Experimental results demonstrate how this methodology achieves state-of-the-art performance on cross-domain aspect extraction using benchmark datasets from three different domains of consumer reviews: restaurants, laptops and digital devices [28, 29, 38]. Finally, we contribute an improved version of the benchmark digital devices dataset to facilitate future work on aspect-based sentiment analysis.

2 RELATED WORK

2.1 Knowledge Graphs

A variety of knowledge graphs have been created in recent years to store large quantities of factual and commonsense knowledge about the world. ConceptNet is a widely-used and freely-available source of commonsense knowledge that was constructed from both expert sources and crowdsourcing. A variety of solutions that leverage ConceptNet have been developed for NLP tasks in recent years, including multi-hop generative QA [1], story completion [5], and machine reading comprehension [41].

The main challenge in using ConceptNet is the selection and quality assessment of paths queried from the graph to produce relevant subgraphs for downstream use. A variety of heuristic approaches have been proposed for this task, including setting a maximum path length [12], limiting the length of the path based on the number of returned nodes [3], and utilizing measures of similarity calculated over embeddings [10]. Auxiliary models that assess the naturalness of paths have also been proposed for predicting path quality [42].

2.2 Domain Adaptation

Developing models that can generalize well to unseen and out-of-domain examples is a fundamental challenge in robust solution design. A key objective of many previous domain adaptation approaches has been to learn domain-invariant latent features that can be used by a model for its final predictions. Prior to the widespread usage of Deep Neural Networks (DNNs) for domain adaptation tasks, various methods were proposed that attempted to learn the latent features by constructing a low-dimensional space where the distance between features from the source and target domain is minimized [22, 23].

With the recent introduction of DNNs for domain adaptation tasks, there has been a shift towards monolithic approaches in which the domain-invariant feature transformation is learned simultaneously with the task-specific classifier as part of the training process. These methods incorporate mechanisms such as a Gradient Reversal Layer [9] and explicit partitioning of a DNN [4] to implicitly learn both domain-invariant and domain-specific features in an end-to-end manner.

Such approaches have been applied to various problems in NLP, including cross-domain sentiment analysis. Du et al. [8] and Gong et al. [11] introduce additional training tasks for BERT [6] in an effort to learn both domain-invariant and domain-specific feature representations for sentiment analysis tasks. The utilization of syntactic information has also been shown to be an effective way of introducing domain-invariant knowledge, which can help bridge the gap between domains [7, 16, 25, 36].

2.3 Knowledge Informed Architectures

An alternative paradigm for developing robust solutions is to augment models using external knowledge queried from a large non-parametric memory store, commonly known as a Knowledge Base (KB) or Knowledge Graph (KG). We refer to this class of models as knowledge informed architectures. Much of the existing work on knowledge informed architectures augments BERT [6] with external knowledge from sources such as WordNet [20] and ConceptNet. These approaches have led to a myriad of new BERT-like models such as KnowBERT [26], K-BERT [39], and E-BERT [27] which attempt to curate and inject knowledge from KBs in various ways. How knowledge is acquired and used in these models is highly task dependent.

Knowledge informed architectures have been shown to be effective at a variety of tasks, achieving superior performance in recent challenges such as Efficient Question-Answering [21] and Open-domain Question-Answering [17, 32] where external knowledge is used to enrich input queries with additional context that supplements the implicit knowledge stored in the model's parameters. To the best of our knowledge, no previous knowledge informed architectures have been developed for cross-domain aspect extraction.

3 METHODOLOGY

Our approach consists of a three-step process: (1) preparing a domain-specific KG for each of the target domains, (2) determining when the model can benefit from external information, and (3) injecting knowledge retrieved from the KG when applicable. We explore two alternative methods for the final knowledge injection step of this process: insertion of a pivot token into the original query, and knowledge injection into hidden state representations via a disentangled attention mechanism. We provide an illustration of our approach in Figure 1 and detail the methodology for each step of the process in the subsequent sections.

3.1 Domain-Specific KG Preparation

In order to ground the model in concepts related to the target domain, we create a domain-specific KG by first querying a subgraph from ConceptNet using a list of seed terms that are related to the domain. For each target domain d, the seed term list $S_d = \{s_1, s_2, ..., s_k\}$ is generated by applying TF-IDF to all of the unlabeled text in domain d and identifying the top-k scoring noun phrases. We use $k = 7$ seed terms in this work but note that the number of seed terms can be adjusted based on the desired size of the KG.

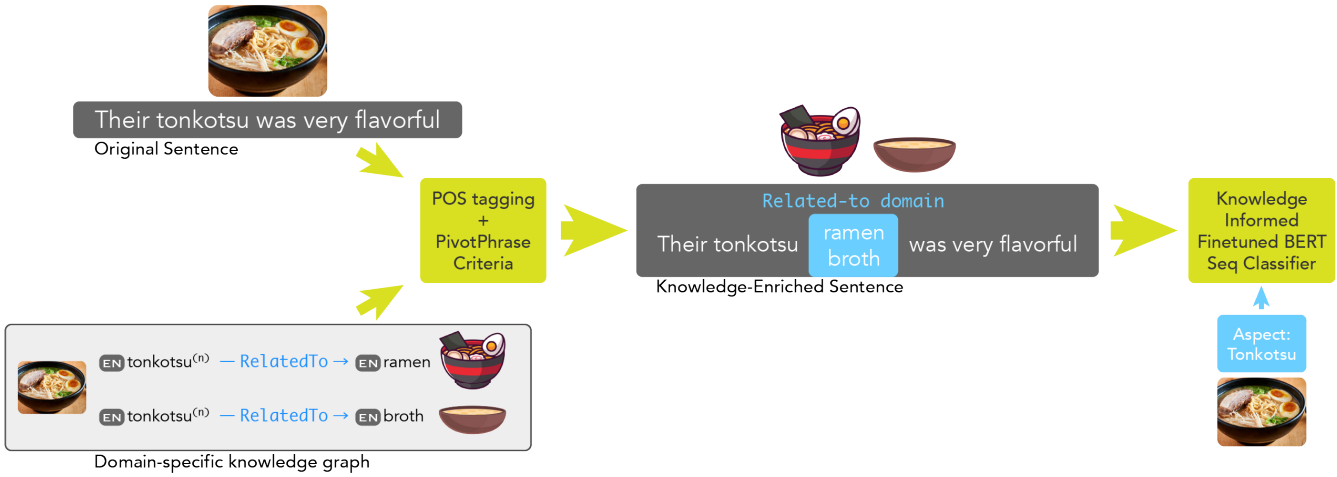


Figure 1: Illustration of our pivot token knowledge injection approach for aspect extraction

For each seed term $s \in S_d$, we query ConceptNet for all English-language nodes connected by an edge to s and add them to the domain-specific subgraph along with the seed term s. The subgraph is further expanded by iteratively querying additional nodes that are connected by an edge to a node already present in the subgraph, up to a maximum distance of h edges from s. In our experiments, we utilized a maximum edge distance of $h = 2$ for efficiency and based on the observation that querying beyond two edges from a given node in ConceptNet does not significantly increase the identification of domain-relevant concepts.

To increase the relevancy of the queried subgraph to the target domain, we prune nodes on paths in the graph that have a low relatedness score to the seed term s from which they originated. While our approach is compatible with various embedding methods, we utilize pre-computed ConceptNet Numberbatch embeddings [33] that combine information from the graph structure of ConceptNet with other word embedding techniques such as Word2vec [19] and GloVe [24]. Let $\mathbf{e}_i$ denote the embedding vector of a given node i in the subgraph. The relatedness score $r_{i,j}$ for a pair of nodes i and j in the graph is calculated as the cosine similarity between their embedding vectors:

$$r_{i,j} = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\|\|\mathbf{e}_j\|} \tag{1}$$

For a given path $P = \{n_s, n_1, ..., n_h\}$ connecting nodes $n_1, ...n_h$ to the node n_s corresponding to seed term s, we calculate its minimum path relatedness score, denoted $P_{\min}$, as the minimum of the pairwise relatedness scores between each node in the path and the seed term:

$$P_{\min} = \min_{\forall i \in \{1,..,h\}} r_{s,i} \tag{2}$$

Nodes terminating a path for which $P_{\min} < 0.2$ are discarded from the subgraph, where this threshold was chosen heuristically and can be tuned based on the application. This path filtering criteria helps disambiguate edges in ConceptNet for words that have multiple different meanings. Higher values of $P_{\min}$ reduce the number of unrelated nodes in the subgraph at the cost of decreased coverage.

To further expand the coverage of the domain-specific KGs, we employ a generative commonsense model called COMET [2] to automatically augment the KG with additional terms that are related to those already present in the graph. Given a head h and relation r, COMET is trained to predict the tail t completing an (h, r, t) triple. We chose to use COMET for augmenting our KGs due to the incompleteness of ConceptNet, which can vary significantly in coverage across domains as a result of its reliance on crowdsourcing for knowledge acquisition.

The original implementation of COMET consisted of GPT [30] fine-tuned to complete (h, r, t) triples sampled from either ConceptNet or ATOMIC [31]. Motivated by the observation that the original COMET lacks coverage for certain concepts in our target domains, we improve the relevancy of its predictions by fine-tuning COMET on ConceptNet triples that are selectively chosen. For each target domain, we identify all nouns and noun phrases occurring in its text using spaCy [14] and then query ConceptNet for triples that contain one of these nouns. A domain-specific instance of COMET is then trained by fine-tuning GPT on the task of (h, r, t) completion using only the sampled set of domain-relevant triples. For each seed term $s \in S_d$, we use our domain-tuned implementation of COMET to generate 100 completions for the triple $(s, \text{RelatedTo}, t)$ and add them to the domain-specific KG if they are not already present.

3.2 Determining when to Inject Knowledge

To determine when to inject knowledge, we identify tokens that are potential aspects by first using spaCy to extract POS and dependency relations. Motivated by the observation that aspects tend to

be either individual nouns or noun phrases, we extract the candidate set of tokens by identifying noun forms in the input sequence. Multi-word phrases are extracted when one or more adjacent tokens have a dependency relation of either "amod" or "compound" and are followed immediately by a noun. Examples of multi-word phrases identified using this criteria include "iMac backup disc" and "external hard drive."

The set of tokens extracted by this process are then compared to the domain-specific KG to determine which should be flagged as related to the domain. For single-token nouns, we look for an exact match to one of the nodes appearing in the domain-specific KG. We also search for an exact match to multi-word noun phrases, but iteratively shorten the phrase by removing the left-most token if an exact match is not found. This iterative approach is used to identify the longest subset of the noun phrase that is present in the KG and is based on the observation that the right-most token in a compound noun is typically the head, which conveys the main meaning of the phrase.

This process results in the identification of a set of tokens within each sentence which are more likely to be aspects due to their syntactic context and relation to the target domain. The final step in our approach injects this knowledge into the language model to improve the accuracy of aspect classification.

3.3 Knowledge Injection Mechanisms

We explore two alternative methods for injecting knowledge into Transformer models. The first approach enriches the input query by inserting a pivot token after tokens identified as potential aspects as described in the previous section. The second approach is inspired by DeBERTa's [13] Disentangled Attention mechanism and utilizes the decomposition of attention to condition each token's attention distribution on the pivoting information.

3.3.1 Knowledge injection via Pivot Token. The pivot token is a special token that serves the purpose of indicating to the model that the preceding token has a greater likelihood of being labeled an aspect. We reserve two distinct pivot tokens in a model's vocabulary, one corresponding to the *BA* class ([DOMAIN-B]) and another to the *IA* class ([DOMAIN-I]). The [DOMAIN-B] pivot token is inserted after single-token aspect candidates or after the first token in a multi-word phrase, with the [DOMAIN-I] pivot token being used for the remaining tokens in multi-word phrases.

The following sentence is an example of a query from a restaurant review that was enriched via this process: "*It was the best pad* [DOMAIN-B] *thai* [DOMAIN-I] *I've ever had.*" In this example, "*pad thai*" was marked for knowledge injection based on its syntactic information and presence in the domain-specific KG.

While the criteria described in Section 3.2 is used to determine when to insert pivot tokens in the target domain datasets, we use a different method of stochastic pivot token insertion for the training data in order to teach the Transformer model how to use the injected knowledge. Specifically, we define hyperparameters p and r that correspond to the desired precision and recall of the pivot token (respectively) when used to identify aspects. We then perform a stochastic insertion of pivot tokens after a portion of the labeled aspects in the training dataset as well as some non-labeled tokens such that the precision and recall of the pivot token approximates p and r. The purpose of this is to adapt the base language model to potential inaccuracies in the pivot token insertions while removing any dependency between the coverage of the KGs in the source and target domains.

3.3.2 Knowledge Injection Using Disentangled Attention. The second approach we consider for injecting knowledge consists of modifying the attention patterns in a Transformer model based on the candidate aspect terms identified. Inspired by the success of DeBERTa on a variety of NLU benchmarks, we utilize the Disentangled Attention mechanism introduced in DeBERTa and augment it with new attention score terms that encode positional information about the location of the candidate aspect terms. The motivation for this approach is twofold. First, it preserves the structure of the original input sequence by not requiring the injection of additional tokens. Second, it allows for finer-grained control over the attention patterns exhibited in the model.

In DeBERTa, each token t_i, $i = 1...N$ in the input sequence is represented by two embeddings: a content embedding c_i and a position embedding p_i. This decomposed representation leads to the formulation of Disentangled Attention as follows.

$$\begin{aligned} A^{c2c}_{i,j} &= Q^c_i K^{cT}_j, A^{c2p}_{i,j} = Q^c_i K^{p^T}_{\delta(i,j)} \\ A^{p2c}_{i,j} &= K^c_j Q^{p^T}_{\delta(j,i)} \\ A_{i,j} &= A^{c2c}_{i,j} + A^{c2p}_{i,j} + A^{p2c}_{i,j} \\ H &= (\frac{A}{\sqrt{3d}})V^c \end{aligned} \tag{3}$$

Here $Q^c, K^c, V^c \in \mathbb{R}^{N\times d}$ are the Query, Key, and Value projections for the content embeddings, and $Q^p, K^p \in \mathbb{R}^{N\times d}$ are the Query and Key projections for the position embeddings. $\delta(i,j) \in [0, 2k)$ is the relative distance between token i and token j where k is the maximum relative distance possible. The joint attention matrix A is then used to construct the next set of hidden states H in the standard manner.

To encode the pivoting information, we define two new learned embedding vectors $m^+, m^- \in \mathbb{R}^d$ to denote whether or not a token is a candidate aspect term (respectively). We use the learned embedding vectors to create a sequence of embeddings $S^m = m_1...m_N$ where

$$m_i = \begin{cases} m^+ \text{ if } t_i \text{ is a candidate aspect} \\ m^- \text{ otherwise} \end{cases}$$

The Query and Key projections for these embedding vectors $Q^m, K^m \in \mathbb{R}^{N\times d}$ are learned and used in our modified attention formulation (Equation 4) through two new constituent terms, A^{c2m} and A^{m2c}.

$$\begin{aligned} A^{c2m}_{i,j} &= Q^c_i K^{mT}_j, A^{m2c}_{i,j} = Q^m_i K^{cT}_j \\ \hat{A}_{i,j} &= A^{c2c}_{i,j} + A^{c2p}_{i,j} + A^{p2c}_{i,j} \\ &\quad + A^{c2m}_{i,j} + A^{m2c}_{i,j} \\ \hat{H} &= (\frac{\hat{A}}{\sqrt{5d}})V^c \end{aligned} \tag{4}$$

Intuitively, the A^{c2m} and A^{m2c} terms act as a mechanism by which the model can adjust the attention distribution for a given token based on relationships between the content representations of tokens created by the Transformer and the pivoting information in S^m. We hypothesize that the A^{c2m} and A^{m2c} terms together encourage the model to learn attention patterns that highlight contributions from the candidate aspect terms, leading to hidden state representations that carry additional relevant information about the locations of potential aspects.

4 EXPERIMENTS

4.1 Experimental Setup

We evaluate the cross-domain aspect extraction performance of our approach on three benchmark ABSA datasets consisting of English-language consumer reviews for restaurants (5,841 sentences), laptops (3,845 sentences), and digital devices (3,836 sentences) [28, 29, 38]. This three-dataset experimental setting is one of the largest available for ABSA, which is more limited in data availability than other sentiment analysis tasks due to the difficult and time-consuming nature of labeling aspects. To assess cross-domain performance, we create pairings of the three data domains as follows: let L, R, and D denote the laptops, restaurants, and device review datasets (respectively). The cross-domain settings on which we evaluate our models is represented by the set $\mathcal{D}$ in Equation 5, where the first element in each tuple is the source domain and the second element is the target domain.

$$\mathcal{D} = \{(L, R), (L, D), (R, L), (R, D), (D, L), (D, R)\} \tag{5}$$

Within each domain, we create 3 separate partitions of the data which are then further divided into a train, validation, and test set following a 3:1:1 ratio. The performance of each model is evaluated by first being fine-tuned for ABSA on the source domain data and then being tested on the target domain data. To control for elements of randomness, we repeat each of our experiments using three different random seeds, reporting the mean and standard deviation of the aspect extraction F1 score calculated over all combinations of random seeds and data partitions (9 in total). In accordance with prior work, only exact matches between the predicted labels and gold labels are counted as correct.

We adopt the HuggingFace [40] implementations of Transformer models used in our experiments and open-source our code[1]. Following the experimental setup of Pereg et al. [25], we use the validation dataset to determine when to apply early stopping as well as the hyperparameters p and r defined in Section 3. Specifically, we use a heuristic approach of setting p and r equal to the evaluated precision and recall of the pivot token insertions on the validation dataset. Other hyperparameters were chosen by adopting the same configuration used previously by Pereg et al. [25], which includes fine-tuning the model using the AdamW optimizer [18] with a learning rate of 5×10^{-5}, a batch size of 8, and a maximum sequence length of 64 tokens for up to 10 epochs. Our experiments were conducted on a Ubuntu 18.04 system with an Intel(R) Xeon(R) Platinum 8280 CPU and three Nvidia RTX 3090 GPUs.

Model	(L, R)	(L, D)	(R, L)	(R, D)	(D, L)	(D, R)	**Mean**
KG-only	56.0	27.3	40.5	28.1	39.5	56.2	41.3
ARNN-GRU*	52.9	40.4	40.4	35.1	51.1	48.4	44.7
	(1.8)	(0.7)	(1.0)	(0.6)	(1.7)	(1.1)	(1.1)
BERT	45.1	42.2	44.6	38.1	47.0	51.9	44.8
	(3.6)	(0.5)	(1.9)	(1.3)	(2.2)	(2.2)	(2.0)
TRNN-GRU*	53.8	41.2	40.2	37.3	51.7	51.2	45.9
	(0.9)	(1.1)	(0.8)	(0.9)	(1.3)	(1.0)	(1.0)
DeBERTa	54.3	40.5	47.5	39.6	47.1	54.5	47.3
	(1.7)	(1.4)	(2.3)	(1.6)	(2.1)	(2.2)	(1.9)
SA-EXAL*	54.7	42.2	47.6	**40.5**	47.7	54.5	47.9
	(2.0)	(0.5)	(1.9)	(1.1)	(2.8)	(1.9)	(1.7)
DeBERTa-MA	61.5	40.2	43.4	38.0	47.2	62.0	48.7
	(1.4)	(1.1)	(2.5)	(1.8)	(1.1)	(0.7)	(1.4)
DeBERTa-PT	66.0	41.0	49.7	38.5	52.5	64.9	52.1
	(1.8)	(1.2)	(1.3)	(0.8)	(1.6)	(0.8)	(1.3)
BERT-PT	**66.4**	**42.3**	**49.9**	39.5	**55.3**	**65.8**	**53.2**
	(1.1)	(1.1)	(1.4)	(1.8)	(1.4)	(0.7)	(1.3)

Table 1: Comparison of average aspect extraction F1 scores (with standard deviation in parentheses). An asterisk indicates previously-reported model results.

4.2 Results

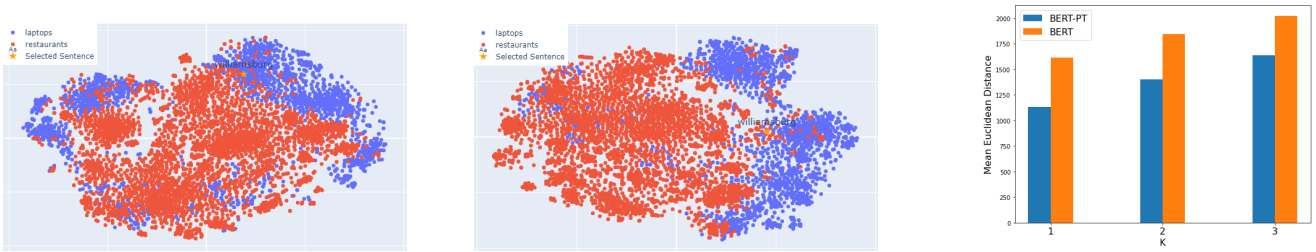
We evaluate the performance of state-of-the-art Transformer models on cross-domain aspect extraction when they are coupled with KGs using the two knowledge injection mechanisms detailed in Section 3.3. The -PT and -MA suffixes within model names indicate the knowledge injection method used by our models, where -PT denotes knowledge injection using pivot tokens and -MA denotes knowledge injection via the modified attention scheme. We compare our knowledge-informed models to the following existing solutions for cross-domain ABSA:

- ARNN-GRU [36], a RNN architecture comprised of GRU blocks augmented with information from dependency trees through an auxiliary dependency relation classification task.
- TRNN-GRU [37], an extension of ARNN-GRU incorporating a conditional domain adversarial network to explicitly align word feature spaces in the source and target domains.
- SA-EXAL [25], a BERT-like model that incorporates syntactic information into its self-attention mechanism.

Additionally, we include several baseline models in our experiments as ablation studies. These include BERT and DeBERTa models that were fine-tuned on the ABSA task without knowledge injection as well as a KG-only solution that classifies aspects solely based on our knowledge injection methodology. These ablations are discussed further in Section 4.3.2.

Table 1 shows the mean and standard deviation of aspect extraction F1 scores in each cross-domain setting for our three knowledge-informed transformers and the baseline models. The best overall performance is obtained by BERT-PT, which injects the knowledge by inserting pivot tokens into the input sequence. BERT-PT achieves substantial improvements over the existing state-of-the-art SA-EXAL with a 5% absolute increase in mean F1 and a 10% absolute F1 improvement when the restaurants dataset is the target domain.

[1] Our code is available via NLP Architect.



(a) (b) (c)

Figure 2: Plots (a) and (b) depict t-SNE projections of the final hidden state embeddings of aspect terms produced by BERT-PT and BERT (respectively) in the laptops (purple) and restaurants (red) domains. Plot (c) shows the mean distance between aspects from one domain to the K-closest aspects in the other domain.

4.3 Analysis & Discussion

All three of our KG-enhanced models (BERT-PT, DeBERTa-PT, and DeBERTa-MA) outperform the existing state-of-the-art solutions, which highlights the usefulness of our knowledge injection methods for improving aspect identification. BERT-PT performed better overall than both DeBERTa-PT and DeBERTa-MA, which is surprising because the baseline DeBERTa model significantly outperformed the baseline BERT model. These results suggest that the optimal knowledge-injection mechanism is highly model-dependent.

One possible explanation for the lower performance of DeBERTa-PT relative to BERT-PT is that inserting tokens into the input sequence acts as a large disruption to the relative positional information used by the model. Inserting a pivot token shifts the relative positional encodings of every pair of tokens on opposite sides of it by one position. This can be contrasted with the absolute positional encodings used by BERT, for which the insertion of a pivot token affects only the positional encodings of the tokens following the pivot token. Moreover, because BERT's positional information is combined with the content information by summing the corresponding embeddings, we hypothesize that the inherent noisiness of this process would make the token representations more robust to changes in the positional encodings.

The overall lower performance of DeBERTa-MA relative to our pivot token models could be due to the knowledge-injection method introducing complexity overhead. The goal of this mechanism is to condition the attention by injecting a binary indicator for candidate aspect terms into the model. However, this requires learning high-dimensional embeddings and projection matrices to represent the binary indicator in the attention computation. The information conveyed by the binary indicators we use to identify candidate aspects may not be enough to offset the added model complexity in a way that improves upon the performance of using a pivot token. However, this mechanism could be beneficial when injecting more fine-grained and semantically-rich knowledge which can make use of the high-dimensional embedding space. We leave the exploration of this topic to future work.

4.3.1 Impact of Knowledge Injection by Domain. Knowledge injection yielded the greatest improvements in aspect identification when the restaurants dataset was used as the target domain. Differences in performance across domains can be attributed to variations in the size of the domain-specific KGs, the cardinality of the set of labeled aspects, and the consistency with which aspect tokens are labeled as aspects. Table 2 provides a comparison of these three metrics across each domain. Aspect cardinality was measured as the number of unique aspect tokens occurring in the domain's test dataset. For each unique aspect, we measured the proportion of times the token was labeled as an aspect and averaged these proportions across all aspects to obtain the aspect consistency. The superior performance of BERT-PT in the restaurants domain appears to be driven by its much larger KG size relative to the other domains, which can be attributed to better coverage of food-related concepts in ConceptNet. The lower consistency of aspect usage in the devices domain is a major factor contributing to worse aspect extraction in this domain. We describe potential annotation differences between domains which may have caused this variation in aspect consistency in Section 4.3.4.

Domain	KG Size	Aspect cardinality	Aspect consistency
Restaurants	8,547	1,321	0.78
Laptops	4,651	847	0.77
Devices	5,824	516	0.49

Table 2: KG size, aspect set cardinality, and mean aspect consistency by target domain

4.3.2 Transformer and KG Baselines. To explore the relative contributions of information embedded in the parameters of the Transformer model and the KG, we independently measure their performance at cross-domain aspect extraction. Specifically, we train baseline BERT and DeBERTa models on the same dataset described previously but without injecting knowledge. We also measure the performance of a KG-only model that utilizes only the information conveyed by our knowledge injection methodology. In the KG-only model, every token that is followed by a pivot token in our knowledge-enriched queries is labeled as an aspect as opposed to using a Transformer model to classify labels.

The results in Table 1 show that DeBERTa outperforms BERT in five out of the six cross-domain settings. The KG-only model performs best when the restaurants dataset is used as the target domain,

even outperforming the other baseline models. These cross-domain results are consistent with those of our knowledge-informed Transformers, which also exhibit the greatest performance when the restaurants dataset is the target domain.

To illustrate the impact of external knowledge injection, Figure 2a and 2b depict the final hidden state embeddings of restaurant and laptop aspects produced by BERT-PT and BERT (respectively) after being projected to 2-D using t-SNE [34]. These visualizations show that BERT-PT can better bridge the gap between aspects from the two domains, as evidenced by the increased overlap between representations of source and target domain aspects. This same effect is measured quantitatively in Figure 2c, which provides the mean Euclidean distance between the embedding of each aspect term and its closest $K \in \{1, 2, 3\}$ embeddings of aspects from the opposite domain.

4.3.3 Stochastic Insertion of Pivot Information During Training. As described in Section 3.3.1, one component of our methodology involves stochastic insertion of the pivot token into the training dataset. Our motivation for using stochastic insertion is to adapt the model to differences in the accuracy and coverage of the KG in different domains. As shown in Table 1, the performance of the KG differs substantially across each of the three domains. This can be attributed both to differences in the coverage of the domain-specific KGs and the degree to which domain-specific words are labeled as aspects in each of the three domains, as described previously.

An alternative to stochastic insertion of the pivot token into the training dataset is to assume both the training and testing domains have similar performance with respect to knowledge insertion. Under this assumption, the pivot token is inserted deterministically into the training dataset using the criteria described previously in Section 3.2. We conducted ablation studies on the training data insertion method and provide a detailed comparison of the aspect identification performance of our models under stochastic and deterministic pivot insertions in Table 3. Method S corresponds to stochastic insertion of the pivot token during training while D corresponds to deterministic insertion during training. These results show that stochastic insertion provides the greatest improvement in performance when the training domain KG performs worse than the testing domain KG. When the converse is true, slightly better performance is achieved by the deterministic insertion method.

Model	Method	(L, R)	(L, D)	(R, L)	(R, D)	(D, L)	(D, R)
BERT-PT	S	**66.4**	42.3	49.9	39.5	**55.3**	**65.8**
BERT-PT	D	47.3	**44.0**	**52.5**	41.8	48.8	52.8
DeBERTa-PT	S	66.0	41.0	49.7	38.5	52.5	64.9
DeBERTa-PT	D	57.9	43.5	51.8	**43.2**	50.1	56.8
DeBERTa-MA	S	61.5	40.2	43.4	38.0	47.2	62.0
DeBERTa-MA	D	51.6	41.7	44.1	38.5	47.1	55.4

Table 3: Comparison of the average aspect extraction F1 score for our knowledge-informed Transformer models trained with stochastic (S) and deterministic (D) training set injection methods

4.3.4 Improving Aspect Label Distribution in the Digital Devices Dataset. Table 1 show that all evaluated models perform the worst when the digital devices dataset is used as the target domain. We believe this is partially attributable to a difference in the distribution of labels in the devices domain relative to that of the restaurant and laptop domains. Specifically, we observe that only 37% of instances within the devices dataset contain a labeled aspect, whereas 50% of instances in the laptop domain and 66% of instances in the restaurant domain contain aspect labels.

Model	(L, R)	(L, D')	(R, L)	(R, D')	(D', L)	(D', R)	Mean
BERT	45.1	53.1	44.6	44.3	**59.1**	57.2	50.5
	(3.6)	(0.7)	(1.9)	(2.0)	(1.8)	(2.6)	(4.2)
DeBERTa	54.3	52.3	47.5	44.3	57.1	60.9	52.7
	(1.7)	(0.3)	(2.3)	(2.7)	(1.4)	(2.5)	(1.6)
DeBERTa-MA	61.5	49.9	43.4,	41.4	50.1	63.9	51.7
	(1.4)	(0.7)	(2.5)	(2.4)	(1.4)	(1.0)	(1.6)
DeBERTa-PT	66.0	53.8	49.7	46.5	53.6	66.0	55.9
	(1.8)	(1.0)	(1.3)	(1.3)	(1.9)	(1.0)	(1.4)
BERT-PT	**66.4**	**56.1**	**49.9**	**46.7**	56.5	**66.8**	**57.0**
	(1.1)	(1.0)	(1.4)	(1.1)	(2.1)	(1.1)	(1.3)

Table 4: Comparison of average aspect extraction F1 scores (with standard deviation in parentheses) using improved devices dataset (D').

One possible cause for this inconsistency is differences in the annotation process used to collect the datasets. While the restaurant and laptop datasets were annotated under the same guidelines, the digital device reviews were collected nearly ten years earlier using different annotation instructions. During annotation of the devices dataset, aspects were only labeled for sentences in which the writer expresses an opinion [15]. This requirement was not specified in the annotation guidelines for the other two domains [29], which may explain why the devices dataset has fewer aspect labels.

Motivated by this observation, we asked crowdsourced workers from Amazon Mechanical Turk to label the devices dataset in an effort to identify missing aspects. Each sentence in the dataset was labeled by five workers. New aspect labels were created when there was a majority agreement among the workers, which were then used to supplement the set of labels in the original devices dataset. We provide additional details on the collection of these new annotations in the appendix.

As a result of this process, the percentage of device reviews containing an aspect increased from 37% to 57%. Table 4 compares our knowledge-informed models to baseline Transformer models on cross-domain aspect extraction tasks utilizing this improved devices dataset (denoted D'). All methods performed better overall, indicating that the increased labeling consistency across the datasets improves domain transfer. We believe that our release of this updated devices dataset will facilitate future ABSA research by reducing annotation inconsistencies between domains.

4.3.5 Impact of Model and External Knowledge Size. A recent trend in language modeling is the use of increasingly larger Transformers to achieve state-of-the-art performance on benchmark datasets, which has raised questions about the sustainability of this continued growth in model size. To explore the potential for external knowledge sources to mitigate this trend, we investigate the effect that increasing the size of the knowledge graph has on aspect extraction performance and contrast this with an increase in the model size of the Transformer.

Figure 3 shows the results of an ablation study on KG and Transformer size using the (L, R) experimental setting, in which the laptops dataset is used as the source domain and the restaurants dataset is used as the target domain. The target domain aspect extraction F1 of our BERT-PT model is provided across varying sizes of the knowledge graph used to inject the pivot token. KG sizes less than 100% were obtained by randomly sampling a percentage of the original target domain KG obtained from ConceptNet. The 100%+COMET result corresponds to using the full target domain KG, which includes all triples extracted from ConceptNet as well as automatically-generated triples produced by COMET. Note that we only vary the size of the KG used to inject knowledge into the target domain test dataset in order to simulate the effect of altering the knowledge source without retraining the model.

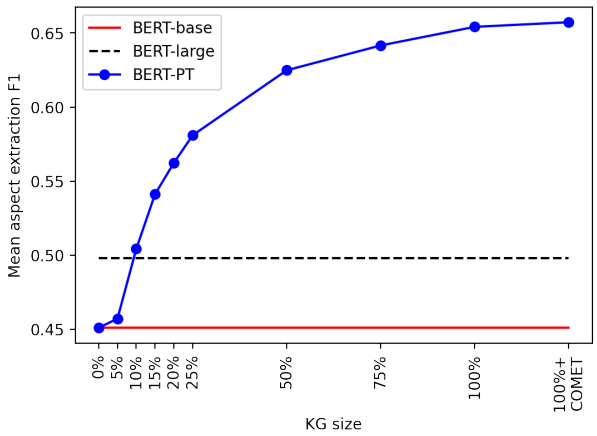


Figure 3: Target domain aspect extraction F1 in the (L, R) experimental setting for different KG and Transformer sizes.

We observe that the aspect extraction F1 of BERT-PT monotonically increases with the size of the KG, which suggests that expanding the amount of external knowledge available to the model can lead to improved performance at inference time without requiring the Transformer to be retrained. In contrast, increasing the size of a BERT model which does not leverage external knowledge from 110M parameters (BERT-base) to 336M parameters (BERT-large) produces a much smaller improvement in aspect extraction F1. Our BERT-PT model outperforms BERT-large when as little as 10% of the original ConceptNet subgraph is used as the KG, despite BERT-large having over 3 times more parameters than BERT-PT.

To illustrate the difference in size of KGs used in this analysis, Figure 4 visualizes the 25% and 100% KGs produced for the restaurant domain. Due to the large size of the full KGs, only a subgraph consisting of triples matching the pattern $(x, AtLocation, restaurant)$ are depicted. Green nodes correspond to labeled aspect tokens in the restaurants domain while non-aspect tokens are represented by blue nodes. The 25% KG lacks many nodes that correspond to labeled aspects, including tokens such as "hostess", "butter", and "pizza". However, many of these missing nodes could be easily added by human annotators in order to improve the performance of the model without the need for retraining. Such human-in-the-loop systems are a promising research direction for future studies on Transformers augmented with external knowledge sources.

BERT		BERT-PT	
TP	FN	TP	FN
service (1066)	food (1288)	food (1797)	place (325)
food (527)	place (376)	service (1075)	of (196)
staff (318)	of (202)	staff (330)	with (141)
menu (189)	pizza (196)	menu (239)	and (138)
atmosphere (145)	dinner (174)	pizza (216)	food (129)
prices (135)	dishes (165)	atmosphere (215)	dinner (122)
price (114)	wine (153)	wine (213)	the (114)
decor (110)	with (147)	dishes (156)	indian (106)
wine (96)	chicken (135)	prices (153)	lunch (93)
list (74)	and (135)	sushi (134)	dumplings (76)

Table 5: The top-10 true positive (TP) and false positive (FP) aspects identified by BERT and BERT-PT in the (L, R) test dataset. Frequency counts are provided in parentheses.

4.3.6 Analysis of errors in the (L, R) experimental setting. In our experiments, BERT-PT achieves its greatest performance when the restaurants dataset is used as the target domain. To better understand how the use of external knowledge is improving aspect extraction in this domain, we investigated which aspects are correctly identified or missed by BERT and BERT-PT in the (L, R) setting.

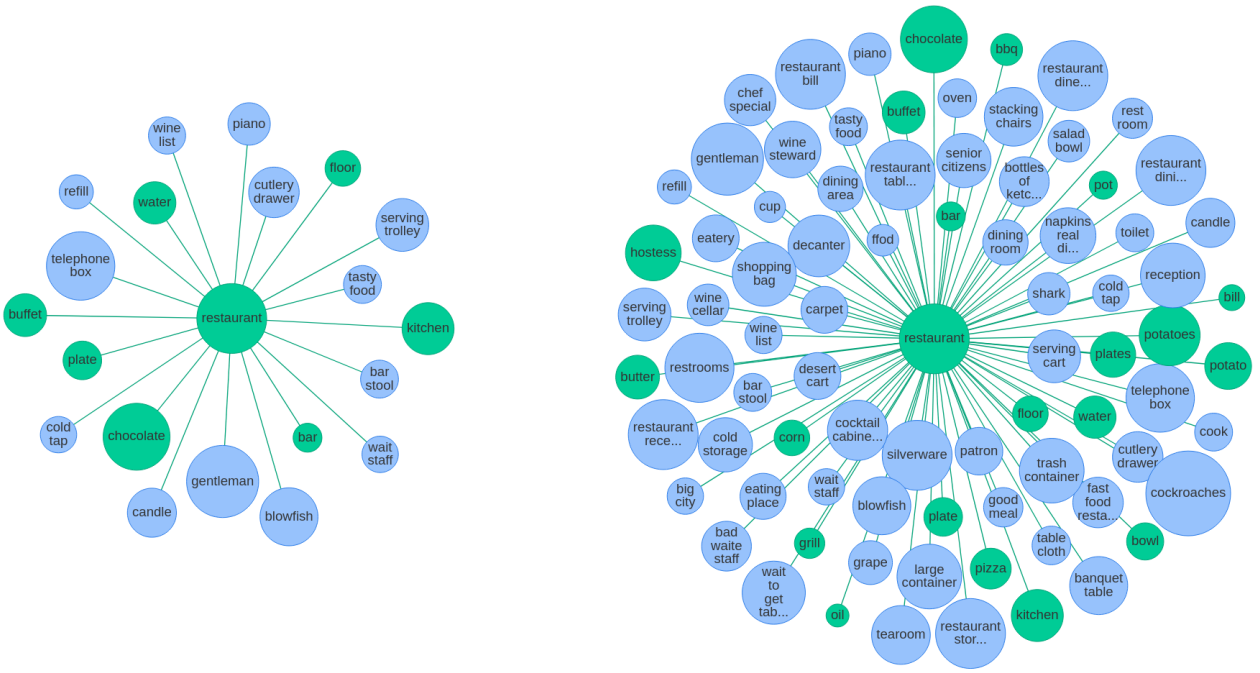
Table 5 provides the top-10 most frequent true positive (TP) and false negative (FN) aspect tokens identified by BERT and BERT-PT. In this analysis, a TP is defined as a token which is labeled as either aspect type and is correctly predicted to be an aspect. A FN is defined as tokens which are labeled as an aspect, but are not predicted to be an aspect. Counts of the number of TP and FN occurences for each token are expressed in parentheses in Table 5.

Both BERT and BERT-PT share many of the same TP aspect tokens. However, BERT-PT correctly classifies more of these aspects than BERT, which suggests that the injection of knowledge helps improve the consistency with which the model correctly classifies aspects in the target domain. A higher proportion of the TP aspects identified by BERT are not specific to the target domain (e.g., 'service', 'price'), whereas BERT-PT correctly identifies more domain-specific aspects (e.g., 'pizza', 'sushi'). Finally, BERT-PT has a higher proportion of stopwords among its top FN aspects, suggesting that more of its errors are associated with tokens that are infrequently labeled as aspects as opposed to the domain-specific aspect words more frequently missed by BERT.

Many of the aspects identified by BERT-PT are never labeled as aspects by BERT. Figure 5 provides a wordcloud visualization of such newly-identified aspects by our model. The size of each aspect token depicted in the wordcloud is proportional to its frequency of occurrence in the restaurants test dataset. This figure shows that BERT-PT identifies a broad range of new aspect words specific to the restaurants domain, which can be attributed to the breadth of knowledge made available to the model through the KG.

5 CONCLUSION

We have presented a comprehensive approach for constructing domain-specific KGs and determining when it is beneficial to inject knowledge into Transformers for aspect extraction. Additionally, we introduced two alternative approaches for knowledge injection:



(a) (b)

Figure 4: Visualizations of a 25% sampled subgraph (a) and the full subgraph (b) of ConceptNet nodes connected to the "restaurant" concept by the "AtLocation" relation. Green nodes correspond to labeled aspect tokens in the restaurants domain while non-aspect tokens are represented by blue nodes.

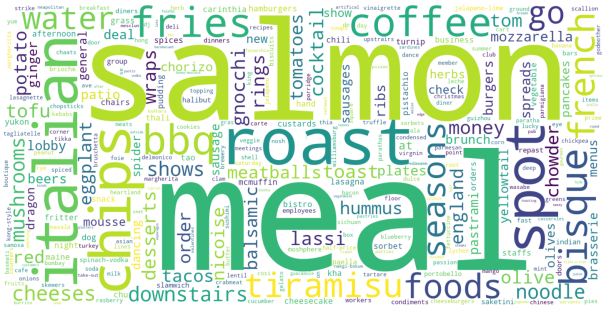


Figure 5: Wordcloud visualization of our most common newly-identified aspects in the restaurants domain

via query enrichment and using a disentangled attention mechanism. Our experimental results show that our knowledge-informed Transformers outperform existing state-of-the-art models on cross-domain aspect extraction tasks. Finally, we release an improved version of the benchmark digital device reviews dataset to support future research on aspect-based sentiment analysis.

While this work focused on identifying aspects for ABSA, our domain-specific KG construction and knowledge injection methods can be applied to other NLP tasks for which external knowledge is beneficial. In future work, we intend to explore extensions of our approach to such applications and investigate alternative methods for injecting knowledge into language models.

A APPENDIX

A.1 Crowdsourcing Annotations for Improved Devices Dataset

Annotations for the digital devices dataset were collected from crowdsourced workers using Amazon Mechanical Turk. Workers were paid a reward of $0.14 per annotated sentence and were required to meet certain criteria in order to work on the task. Specifically, workers were required to have a Human Intelligence Task (HIT) approval rate > 95%, completed at least 5000 HITs, and obtained the Mechanical Turk Masters qualification.

Workers were asked to apply aspect and opinion labels by highlighting portions of each sentence. Aspect labels were further segmented into positive, negative, neutral, and conflict polarities in accordance with prior labeling tasks for ABSA. Detailed instructions were provided on how to identify aspect and opinion words, including multiple examples from the laptop reviews dataset.

Overlapping text selections were required from at least 3 out of the 5 workers to generate a new label. For multi-word phrases, we took the longest subset of the phrase which was selected by at least 3 workers as the label. New aspect annotations were added to the existing set of aspect labels to produce the improved devices dataset. In cases where multi-word labels overlapped between the original and new label set, we kept the longer form of the label.

REFERENCES

[1] Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. *arXiv preprint arXiv:1809.06309* (2018).

[2] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, A. Çelikyilmaz, and Yejin Choi. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *ACL*.

[3] Adrian Boteanu and Sonia Chernova. 2015. Solving and explaining analogy questions using semantic networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[4] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain Separation Networks. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/45fbc6d3e05ebd93369ce542e8f2322d-Paper.pdf

[5] Jiaao Chen, Jianshu Chen, and Zhou Yu. 2019. Incorporating structured commonsense knowledge in story completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6244–6251.

[6] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[7] Ying Ding, Jianfei Yu, and Jing Jiang. 2017. Recurrent neural networks with auxiliary labels for crossdomain opinion target extraction. In *Association for the Advancement of Artificial Intelligence*. 3436––3442.

[8] Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. 2020. Adversarial and Domain-Aware BERT for Cross-Domain Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4019–4028. https://doi.org/10.18653/v1/2020.acl-main.370

[9] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France) *(ICML'15)*. JMLR.org, 1180–1189.

[10] Matt Gardner, Partha Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 397–406.

[11] Chenggong Gong, Jianfei Yu, and Rui Xia. 2020. Unified Feature and Instance Based Domain Adaptation for Aspect-Based Sentiment Analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7035–7045. https://doi.org/10.18653/v1/2020.emnlp-main.572

[12] Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094* (2015).

[13] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. arXiv:2006.03654 [cs.CL]

[14] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python*. https://doi.org/10.5281/zenodo.1212303

[15] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 168–177.

[16] Vasudev Lal, Arden Ma, Estelle Aflalo, Phillip Howard, Ana Simoes, Daniel Korat, Oren Pereg, Gadi Singer, and Moshe Wasserblat. 2021. InterpreT: An Interactive Visualization Tool for Interpreting Transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Online, 135–142. https://www.aclweb.org/anthology/2021.eacl-demos.17

[17] Patrick Lewis, Ethan Perez, Aleksandara Piktus, F. Petroni, V. Karpukhin, Naman Goyal, Heinrich Kuttler, M. Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *ArXiv* abs/2005.11401 (2020).

[18] I. Loshchilov and F. Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546* (2013).

[20] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (Nov. 1995), 39–41. https://doi.org/10.1145/219717.219748

[21] Sewon Min, Jordan L. Boyd-Graber, C. Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, T. Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Kuttler, L. Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, F. Petroni, Lucas Hosseini, Nicola De Cao, E. Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, S. Sato, Ryo Takahashi, J. Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, P. Smrz, Hao Cheng, Y. Shen, X. Liu, Pengcheng He, W. Chen, Jianfeng Gao, Barlas Oğuz, Xilun Chen, V. Karpukhin, Stanislav Peshterliev, Dmytro Okhonko, M. Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen tau Yih. 2021. NeurIPS 2020 EfficientQA Competition: Systems, Analyses and Lessons Learned. *ArXiv* abs/2101.00133 (2021).

[22] Sinno Jialin Pan, James T. Kwok, and Qiang Yang. 2008. Transfer Learning via Dimensionality Reduction. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2* (Chicago, Illinois) *(AAAI'08)*. AAAI Press, 677–682.

[23] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. 2011. Domain Adaptation via Transfer Component Analysis. *IEEE Transactions on Neural Networks* 22, 2 (2011), 199–210. https://doi.org/10.1109/TNN.2010.2091281

[24] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[25] Oren Pereg, Daniel Korat, and Moshe Wasserblat. 2020. Syntactically Aware Cross-Domain Aspect and Opinion Terms Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 1772–1777. https://doi.org/10.18653/v1/2020.coling-main.158

[26] Matthew E. Peters, Mark Neumann, Robert L Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *EMNLP*.

[27] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 803–818. https://doi.org/10.18653/v1/2020.findings-emnlp.71

[28] 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)*. Association for Computational Linguistics, Denver, Colorado, 486–495. https://doi.org/10.18653/v1/S15-2082

[29] 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)*. Association for Computational Linguistics, Dublin, Ireland, 27–35. https://doi.org/10.3115/v1/S14-2004

[30] A. Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training.

[31] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 3027–3035. https://doi.org/10.1609/aaai.v33i01.33013027

[32] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and J. Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. *ArXiv* abs/2104.07567 (2021).

[33] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence* 31, 1 (Feb. 2017). https://ojs.aaai.org/index.php/AAAI/article/view/11164

[34] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[36] Wenya Wang and Sinno Jialin Pan. 2018. Recursive Neural Structural Correspondence Network for Cross-domain Aspect and Opinion Co-Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 1—-11.

[37] Wenya Wang and Sinno Pan. 2019. Syntactically-Meaningful and Transferable Recursive Neural Networks for Aspect and Opinion Extraction. *Computational Linguistics* 45 (10 2019), 1–32. https://doi.org/10.1162/COLI_a_00362

[38] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 616–626. https://doi.org/10.18653/v1/D16-1059

[39] Zhe Zhao Zhiruo Wang Qi Ju Haotang Deng Ping Wang Weijie Liu, Peng Zhou. 2020. K-BERT: Enabling Language Representation with Knowledge Graph. In *Proceedings of AAAI 2020*.

[40] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu,

Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

[41] Jiangnan Xia, Chen Wu, and Ming Yan. 2019. Incorporating relation knowledge into commonsense reading comprehension with multi-task learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2393–2396.

[42] Yilun Zhou, Steven Schockaert, and Julie Shah. 2019. Predicting conceptnet path quality using crowdsourced assessments of naturalness. In *The World Wide Web Conference*. 2460–2471.