A Knowledge-Driven Approach to Classifying Object and Attribute Coreferences in Opinion Mining

Jiahua Chen, Shuai Wang, Sahisnu Mazumder, Bing Liu
Department of Computer Science, University of Illinois at Chicago, USA
jiahuaqy@gmail.com, shuaiwanghk@gmail.com
sahisnumazumder@gmail.com, liub@uic.edu

Abstract

Classifying and resolving coreferences of objects (e.g., *product names*) and attributes (e.g., product *aspects*) in opinionated reviews is crucial for improving the opinion mining performance. However, the task is challenging as one often needs to consider domain-specific knowledge (e.g., *iPad is a tablet and has aspect resolution*) to identify coreferences in opinionated reviews. Also, compiling a handcrafted and curated domain-specific knowledge base for each domain is very time consuming and arduous. This paper proposes an approach to *automatically* mine and leverage domain-specific knowledge for classifying objects and attribute coreferences. The approach extracts domain-specific knowledge from *unlabeled* review data and trains a knowledge-aware neural coreference classification model to leverage (useful) domain knowledge together with general commonsense knowledge for the task. Experimental evaluation on real-world datasets involving five domains (product types) shows the effectiveness of the approach.

1 Introduction

Coreference resolution (CR) aims to determine whether two mentions (linguistic referring expressions) corefer or not, i.e., they refer to the same entity in the discourse model (Jurafsky, 2000; Ding and Liu, 2010; Atkinson et al., 2015; Lee et al., 2017, 2018; Joshi et al., 2019; Zhang et al., 2019b). The set of coreferring expressions forms a coreference chain or a cluster. Let's have an example:

> **[S1]** *I bought a green Moonbeam for myself.* **[S2]** *I like its voice because it is loud and long.*

Here all colored and/or underlined phrases are mentions. Considering S1 (sentence-1) and S2 (sentence-2), the three mentions "*I*", "*myself*" in S1 and "*I*" in S2 all refer to the same person and form a cluster. Similarly, "*its*" in S2 refers to the object "*a green Moonbeam*" in S1 and the cluster is {"*its*" (S2), "*a green Moonbeam*" (S1) }. The mentions "*its voice*" and "*it*" in S2 refer to the same attribute of the object "*a green Moonbeam*" in S1 and form cluster {"*its voice*" (S2), "*it*" (S2)}.

CR is beneficial for improving many downstream NLP tasks such as question answering (Dasigi et al., 2019), dialog systems (Quan et al., 2019), entity linking (Kundu et al.), and opinion mining (Nicolov et al., 2008). Particularly, in opinion mining tasks (Liu, 2012; Wang et al., 2016; Zhang et al., 2018; Ma et al., 2020), Nicolov et al. (2008) reported performance improves by 10% when CR is used. The study by Ding and Liu (2010) also supports this finding. Considering the aforementioned example, without resolving "*it*" in S2, it is difficult to infer the opinion about the attribute "*voice*" (i.e., the *voice*, which "*it*" refers to, is "*loud and long*"). Although CR plays such a crucial role in opinion mining, only limited research has been done for CR on opinionated reviews. CR in opinionated reviews (e.g., Amazon product reviews) *mainly* concerns about resolving coreferences involving objects and their attributes. The objects in reviews are usually the names of products or services while attributes are aspects of those objects (Liu, 2012).

Resolving coreferences in text broadly involves performing three tasks (although they are often performed jointly or via end-to-end learning): (1) identifying the list of mentions in the text (known as ***mention detection***); (2) given a pair of candidate mentions in text, making a binary classification decision: *coreferring* or *not* (referred to as ***coreference classification***), and (3) grouping coreferring mentions (referring to the same discourse entity) to form a coreference chain (known as ***clustering***). In reviews, **mention detection** is equiv-

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1616–1626

alent to extracting entities and aspects in reviews which has been widely studied in opinion mining or sentiment analysis (Hu and Liu, 2004; Qiu et al., 2011; Xu et al., 2019; Luo et al., 2019; Wang et al., 2018; Dragoni et al., 2019; Asghar et al., 2019). Also, once the coreferring mentions are detected via classification, clustering them could be straightforward[1]. Thus, following (Ding and Liu, 2010), *we only focus on solving the* ***coreference classification*** *task in this work, which we refer to as the* ***object and attribute coreference classification*** *(OAC2) task onwards.* We formulate the OAC2 problem as follows.

Problem Statement. Given a review ***text*** u (context), an ***anaphor***[2] p and a ***mention*** m which refers to either an object or an attribute (including their position information), our goal is to *predict whether the anaphor* p *refers to mention* m*, denoted by a binary class* $y \in \{0, 1\}$. Note. an anaphor here can be a *pronoun* (e.g., "it") or *definite noun phrase* (e.g., "the clock") or *ordinal* (e.g., "the green one").

In general, to classify coreferences, one needs intensive knowledge support. For example, to determine that "*it*" refers to "*its voice*" in S2, we need to know that "*voice*" can be described as "*loud and long*" and "*it*" can not refer to "*a green Moonbeam*" in S1, since "*Moonbeam*" is a clock which cannot be described as "*long*".

Product reviews contain a great many such domain-specific concepts like brands (e.g., "*Apple*" in the laptop domain), product name (e.g., "*T490*" in the computer domain), and aspects (e.g."*hand*" in the alarm clock domain) that often do not exist in general knowledge bases (KBs) like WordNet (Miller, 1998), ConceptNet (Speer and Havasi, 2013), etc. Moreover, even if a concept exists in a general KB, its semantics may be different than that in a given product domain. For example, "*Moonbeam*" in a general KB is understood as "*the light of the moon*" or *the name of a song*, rather than a clock (in the alarm clock domain). To encode such domain-specific concepts, we need to mine and feed domain knowledge (e.g., "*clock*" for "*Moonbeam*", "*laptop*" for "*T490*") to a coreference classification model. Existing CR methods (Zhang et al., 2019b) do not leverage such domain knowledge and thus, often fail to resolve such co-references that require explicit reasoning over domain facts.

In this paper, we propose to *automatically* mine such domain-specific knowledge from *unlabeled* reviews and leverage the useful pieces of the extracted domain knowledge together with the (general/comensense) knowledge from general KBs to solve the OAC2 task[3]. Note the extracted domain knowledge and the general knowledge from the existing general KBs are both considered as candidate knowledge. To leverage such knowledge, we design a novel knowledge-aware neural coreference classification model that selects the useful (candidate) knowledge with attention mechanism. We discuss our approach in details in Section 3.

The main contributions of this work can be summarized:

1. We propose a knowledge-driven approach to solving OAC2 in opinionated reviews. Unlike existing approaches that mostly dealt with general CR corpus and pronoun resolution, we show the importance of leveraging domain-specific knowledge for OAC2.
2. We propose a method to automatically mine domain-specific knowledge and design a novel knowledge-aware coreference classification model that leverages both domain-specific and general knowledge.
3. We collect a new review dataset[4] with five domains or product types (including both unlabeled and labeled data) for evaluation. Experimental results show the effectiveness of our approach.

2 Related Work

Coreference resolution has been a long-studied problem in NLP. Early approaches were mainly rule-based (Hobbs, 1978) and feature-based (Ding and Liu, 2010; Atkinson et al., 2015) where researchers focused on leveraging lexical, grammatical properties and semantic information. Recently, end-to-end solutions with deep neural models (Lee

[1]Given a text (context), if pairs (m, p), (m, q) are classified as co-referring mentions, then m, p, q belong to same cluster.

[2]The term anaphor used in this work does not have to be the same as defined in other related studies, as here it can also appear before m though rarely. We still name it as anaphor for simplicity, mainly following (Ding and Liu, 2010).

[3]The unlabeled data are from the same source as the annotated data (i.e., the same domain, but without labels), which can ensure the reliability of the domain knowledge as well as the coverage of mention words. With the domain-specific knowledge mined, the meaning of a mention in a certain domain can be better understood (by a model) with the support of its relevant mentions (extracted from the self-mined KB).

[4]https://github.com/jeffchen2018/review_coref

et al., 2017, 2018; Joshi et al., 2019) have dominated the coreference resolution research. But they did not use external knowledge.

Conisdering CR approaches that use external knowledge, Aralikatte et al. (2019) solved CR task by incorporate knowledge or information in reinforcement learning models. Emami et al. (2018) solved the binary choice coreference-resolution task by leveraging information retrieval results from search engines. Zhang et al. (2019a,b) solved pronoun coreference resolutions by leveraging contextual, linguistic features, and external knowledge where knowledge attention was utilized. However, these works did not deal with opinionated reviews and also did not mine or use domain-driven knowledge.

In regard to CR in opinion mining, Ding and Liu (2010) formally introduced the OAC2 task for opinionated reviews, which is perhaps the only prior study on this problem. However, it only focused on classifying coreferences in comparative sentences (not on all review sentences). We compare our approach with (Ding and Liu, 2010) in Section 4.

Many existing general-purpose CR datasets are not suitable for our task, which include MUC-6 and MUC-7 (Hirschman and Chinchor, 1998), ACE (Doddington et al., 2004), OntoNotes (Pradhan et al., 2012), and WikiCoref (Ghaddar and Langlais, 2016). Bailey et al. (2015) proposed an alternative Turing test, comprising a binary choice CR task that requires significant commonsense knowledge. Yu et al. (2019) proposed visual pronoun coreference resolution in dialogues that require the model to incorporate image information. These datasets are also not suitable for us as they are not opinionated reviews. We do not focus on solving pronoun resolution here because, for opinion text such as reviews, discussions and blogs, personal pronouns mostly refer to one person (Ding and Liu, 2010). Also, we aim to leverage domain-specific knowledge on (unlabeled) domain-specific reviews to help the CR task which has not been studied by any of these existing CR works.

3 Proposed Approach

Model Overview. Our approach consists of the following three main steps: **(1) knowledge aquisition**, where given the (input) pair of mention m (e.g., "*a green Moonbeam*") and anaphor p (e.g., "*it*") and the context t (i.e., the review text), we acquire candidate knowledge involving m, denoted as K_m. K_m consists of both domain knowledge (mined from unlabeled reviews) as well as general knowledge (compiled from existing general KBs) (discussed in Section 3.1). Next, in **(2) syntax-based span representation**, we extract syntax-related phrases for mention m and anaphor p. Syntax-related phrases are basically noun phrases, verbs or adjectives that have a dependency relation[5] with m (or p). For example, "*bought*" is a syntax-related phrase of the mention "*a green Moonbeam*" and "*like*" and "*voice*" are two syntax-related phrases for the anaphor "it" in the example review text in Section 1. Once the syntax-related phrases are extracted and the candidate knowledge is prepared for m and p, we learn vector representations of the phrases and the knowledge (discussed in Section 3.2), which are used in step-3. Finally, in **(3) knowledge-driven OAC2 model**, we select and leverage useful candidate domain knowledge together with general knowledge to solve the OAC2 task. Figure 1 shows our model architecture. Table 1 summarizes a (non-exhaustive) list of notations, used repeatedly in subsequent sections.

Table 1: Summary of notations (non-exhaustive list)

d	a domain
t	a review text or context
m	a mention
p	an anaphor
K_m	(domain+general) knowledge involving m for domain d
K_m^d	domain knowledge involving m for d
S_m	syntax-related phrases of m
S_p	syntax-related phrases of p
T_d	labeled reviews in d
$\overline{T}_d$	unlabeled reviews in d

3.1 Knowledge Acquisition

Domain Knowledge Mining. Given the mention m, we first split the mention into words. Here, we only keep the words that satisfy one of the following two conditions[6]: (1) a word is a noun (determined by its POS tag); (2) a word is part of a named entity (by NER). For example, "*a westclox clock*" will result in words "*westclox*" and "*clock*". We use the mention words as the keys to search a domain knowledge base (KB) to retrieve domain

[5] We use `spacy.io` for dependency parsing, POS tagging and Named Entity Recognition (NER) in our implementation.

[6] When only using two features of words, we already achieve good results. More features are left for future work.

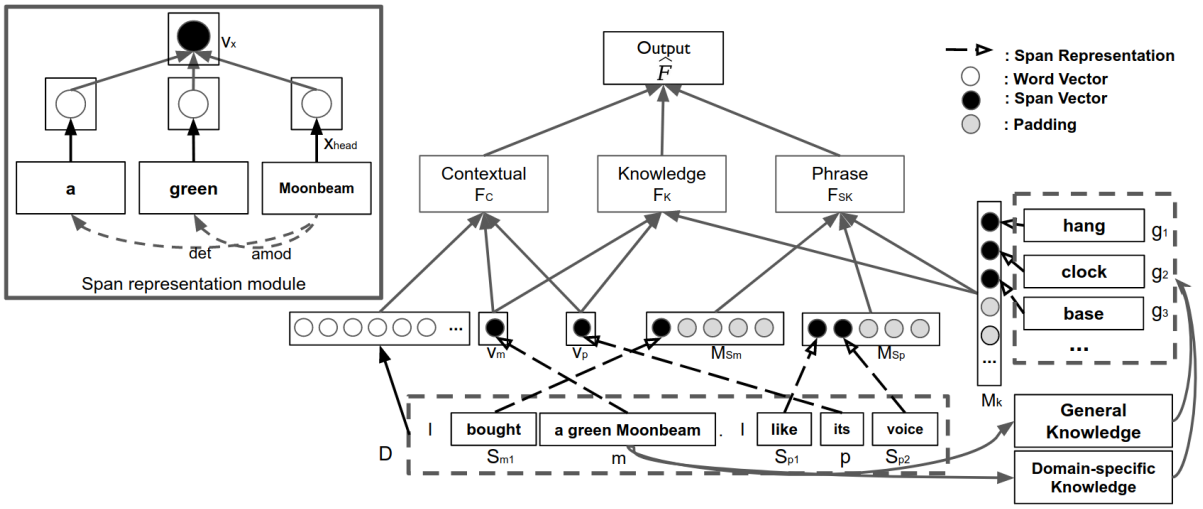

Figure 1: The architecture of our knowledge-driven OAC2 model.

knowledge for the mention m.

To construct the domain KB, we use unlabeled review data in the particular domain. Specifically, all unlabeled sentences that contain mention words are extracted. Next, we collect domain knowledge for m as K_m^d, where $K_m^d = \{k_{m,1}^d, k_{m,2}^d, ...\}$. The elements in K_m^d are phrases of nouns, adjectives, and verbs co-occurring with m in the unlabeled review sentences.

Domain Knowledge Filtering. Some domain knowledge (i.e., co-occurring phrases) can be too general to help reason over the mention. For example, given mention "*Moonbeam*", the verb "*like*" can be related to any objects or attributes and thus, is not a very useful knowledge for describing the mention. To filter such unimportant phrases from K_m^d, we use $tf\text{-}idf$ (Aizawa, 2003) scoring.

Given mention m and a phrase $k \in K_m^d$, we compute $tf\text{-}idf$ score of k, denoted as $tf\text{-}idf_k$ as given below:

$$tf_k = \frac{C_k}{max_{k' \in K_m^d} C_{k'}} \quad (1)$$

$$idf_k = \log \frac{|\overline{T}_d|}{|\{t' \in \overline{T}_d : k \in t'\}|} \quad (2)$$

$$tf\text{-}idf_k = tf_k \cdot idf_k \quad (3)$$

where C_k denotes the co-occurrence count of phrase k with m in unlabeled domain reviews $\overline{T}_d$ and $|\cdot|$ denotes set count. We retain phrase k in k_m^d, if $tf\text{-}idf_k \geq \rho$, where ρ is a (empirically set) threshold value.

General Knowledge Aquisition. General Knowledge bases like ConceptNet, WordNet, etc. store facts as triples of the form (e_1, r, e_2), denoting entity e_1 is related to entity e_2 by a relation r. e.g., ("*clock*", "*UsedFor*", "*set an alarm*").

To acquire and use general knowledge for mention m, we first split m into words (in the same way as we do during domain knowledge construction) and use these words as keywords to retrieve triples such that one of the entities (in a given triple) contains a word of m. Finally, we collect the set of entities (from the retrieved triples) as general knowledge for m, by selecting the other entity (i.e., instead of the entity involving a mention word) from each of those retrieved triples.

3.2 Syntax-based Span Representation

Once the domain-specific and general knowledge for mention m is acquired, we extract all syntax-related phrases for m and anaphor p from review text t (see "Model Overview" in Section 3). We denote the syntax-related phrases of m and p as S_m and S_p respectively.

We represent mention, anaphor, the syntax-related phrases, and also the phrases of knowledge from domain-specific and general KBs as **spans** (a continuous sequence of words), and learn a vector representation for each span (we call it a **span vector**) based on the embeddings of words that compose the span. The span vectors are then used by our knowledge-driven OAC2 model (discussed in Section 3.3) for solving the OAC2 task. Below, we discuss the span vector representation learning for a given span (corresponding to a syntax-related phrase or a phrase in KB).

We use BERT (Devlin et al., 2019) to learn the vector representation for each span. To encode

the words in a span, we use BERT's WordPiece tokenizer. Given a span x, let $\{x_i\}_{i=1}^{N_1}$ be the output token embeddings of x from BERT, where N_1 is the total number of word-piece tokens for span x.

BERT is a neural model consisting of stacked attention layers. To incorporate the syntax-based information, we want the head of a span and words that have a modifier relation to the head to have higher attention weights. To achieve the goal, we adopt syntax-based attention (He et al., 2018). The weight of a word in a span depends on the dependency parsing result of the span. Note, the dependency parsing of a span is different from what is described in Section 3.1. The dependency parsing in Section 3.1 extracts the relation between chunks of words while here we extract relations between single words.

An example has been shown in top left corner of Figure 1. The head of "*a green Moonbeam*" is "*Moonbeam*" that we want to have the highest attention weight when computing the embedding of the span. The distance of ("*a*", "*Moonbeam*") and ("*green*", "*Moonbeam*") considering the dependency path are both 1.

To learn the span vector v_x for span x, we first compute the attention weights b_i's for each x_i, as:

$$f_i = FFN_1([x_i, x_{head}, x_i \odot x_{head}]) \quad (4)$$

$$a_i = \begin{cases} \frac{1}{2^{l_i}} \cdot exp(f_i), & if\ l_i \leq L \\ 0, & otherwise \end{cases} \quad (5)$$

$$b_i = \frac{a_i}{\sum_{j=1}^{N_1} a_j} \quad (6)$$

where FFN_1 is a feed-forward layer that projects the input into a score f_i, $\odot$ is element-wise multiplication, $[,]$ is concatenation, x_{head} is the head of the span, l_i is the distance to the head along the dependency path, L is the attention window size.

Next, we learn the attention-based representation of the span x, denoted as $\hat{x}$ as:

$$\hat{x} = \sum_{i=1}^{N_1} b_i \cdot x_i \quad (7)$$

Finally, we concatenate the start and end word embeddings of the span x_{start} and x_{end}, attention-based representation $\hat{x}$ and a length feature $\phi(x)$ following (Lee et al., 2017) to learn span vector v_x:

$$v_x = FFN_2([x_{start}, x_{end}, \hat{x}, \phi(x)]). \quad (8)$$

where FFN_2 is a feed-forward layer.

3.3 Knowledge-driven OAC2 Model

The knowledge-driven OAC2 model leverages the syntax-related phrases together with the domain knowledge and general knowledge to solve the OAC2 task. The model first computes three relevance scores: **(a)** a contextual relevance score F_C between m and p, **(b)** a knowledge-based relevance score F_K between m and p, and **(c)** a relevance score F_{SK} between knowledge and syntax-related phrases (see Figure 1) and then, these scores are summed up to compute the final prediction score $\hat{F}$, as shown below:

$$\hat{F} = sigmoid(F_C + F_K + F_{SK}) \quad (9)$$

(a) Contextual Relevance Score (F_C). F_C is computed based on the context t, mention m and anaphor p. We use BERT to encode t. Let the output BERT embeddings of words in t be $\{t_i\}_{i=1}^{N_2}$, where N_2 is length of t. Also, let the span vector representations of m and p are v_m and v_p respectively. Then, for each $v \in \{v_m, v_p\}$, we compute cross attention between t and v as follows:

$$g_i = FFN_3([t_i, v, t_i \odot v]) \quad (10)$$

$$w_i^v = \frac{e^{g_i}}{\sum_{j=1}^{N_2} e^{g_j}} \cdot t_i \quad (11)$$

where FFN_3 is a feed-forward layer.

We learn the interaction of $\{t_i\}_{i=1}^{N_2}$ with v_m and v_p to get attention-based vector representations $\{w_i^m\}_{i=1}^{N_2}$ and $\{w_i^p\}_{i=1}^{N_2}$ for m and p respectively. Next, we concatenate these vectors and their pointwise multiplication for each context word, sum up the concatenated representations and feed it to a feed-forward layer to compute $F_C \in \mathcal{R}^{1\times 1}$:

$$F_C = FFN_4(\sum_{i=1}^{N_2}[w_i^m, w_i^p, w_i^m \odot w_i^p]) \quad (12)$$

where FFN_4 is a feed-forward layer.

(b) Knowledge-based Relevance Score (F_K). The OAC2 model leverages the external knowledge to compute a relevance score F_K between m and p. Let v_m and v_p be the span vectors for m and p and $\{v_i^K\}_{i=1}^{N_3}$ be the span vectors for phrases in K_m (see Sec 3.1 and Table 1), where N_3 is size of K_m. Then, we compute F_K using v_m, v_p and $\{v_i^K\}_{i=1}^{N_3}$ as discussed below.

To leverage external knowledge information, we first learn cross attention between the mention and

the knowledge as:

$$h_i = FFN_5([v_i^K, v_m, v_i^K \odot v_m]) \quad (13)$$

$$c_i = \frac{e^{h_i}}{\sum_{j=1}^{N_3} e^{h_j}} \quad (14)$$

where FFN_5 is a feed-forward layer.

Next, we learn an attention-based representation $\hat{v}_m$ of mention m as:

$$\hat{v}_m = \sum_{i=1}^{N_3} c_i \cdot v_i^K \quad (15)$$

We now concatenate v_m, v_p, the attention-based representation $\hat{v}_m$ and learn interaction between them to compute $F_K \in \mathcal{R}^{1\times1}$ as:

$$F_K = FFN_6([v_m, v_p, \hat{v}_m, v_p \odot \hat{v}_m, v_p \odot \hat{v}_m]) \quad (16)$$

where FFN_6 is a feed-forward layer.

(c) Syntax-related Phrase Relevance Score (F_{SK}). F_{SK} measures the relevance between the knowledge (i.e., phrases) in K_m and the syntax-related phrases in S_m (S_p) corresponding to m (p).

Let v_i^K be the span vector for i^{th} phrase in K_m and v_i^m (v_i^p) be the span vector for i^{th} phrase in S_m (S_p). Then, we concatenate these span vectors row-wise to form matrices $M_K = v_i^K \Big\|_{i=1}^{N_3} \in \mathcal{R}^{N_3\times d}$, $M_{Sm} = v_i^m \Big\|_{i=1}^{N_4} \in \mathcal{R}^{N_4\times d}$ and $M_{Sp} = v_i^p \Big\|_{i=1}^{N_5} \in \mathcal{R}^{N_5\times d}$ respectively, where $\Big\|_{i=1}^{Q}$ denotes concatenation of Q elements, d is dimension of span vector, N_4 (N_5) is size of S_m (S_p).

Next, we learn interaction between these matrices using scaled dot attention (Vaswani et al., 2017) as:

$$\tilde{M}_{Sm} = softmax(\frac{M_{Sm}M_K^T}{\sqrt{d}})M_K \quad (17)$$

$$\tilde{M}_{Sp} = softmax(\frac{M_{Sp}M_K^T}{\sqrt{d}})M_K \quad (18)$$

Finally, the syntax-related phrase relevance score $F_{SK} \in \mathcal{R}^{1\times1}$ is computed as:

$$F_{SK} = FFN_8(FFN_7(\tilde{M}_{Sm}\tilde{M}_{Sp}^T)) \quad (19)$$

where FFN_7 and FFN_8 are two feed-forward network layers.

Loss Function. As shown in Equation 9, given three scores F_C, F_K, and F_{SG}, we sum them up and then feed the sum into a sigmoid function to get the final prediction $\hat{F}$. The proposed model is trained in an end-to-end manner by minimizing the following cross-entropy loss $\mathcal{L}$:

$$\mathcal{L} = -\frac{1}{N}\sum_{i}^{N}[y_i \cdot \log(\hat{F}_i) + (1-y_i)\cdot\log(1-\hat{F}_i)] \quad (20)$$

where, N is the number of training examples and y_i is the ground truth label of i^{th} training example.

Table 2: Dataset Statistics. #R means the number of annotated reviews and #E indicates total entities that refer to objects or attributes. P and N stand for positive and negative examples and the values under them are the numbers of those examples.

Domain	#R	#E	Train		Dev		Test	
			P	N	P	N	P	N
alarm	100	924	647	1533	96	243	89	187
camera	100	871	632	1709	69	160	83	174
cellphone	100	938	679	1693	62	148	73	189
computer	100	1035	703	1847	86	227	112	273
laptop	100	893	641	1618	88	244	77	209

4 Experiments

We evaluate our proposed approach using five datasets associated with five different domains: (1) *alarm clock*, (2) *camera*, (3) *cellphone*, (4) *computer*, and (5) *laptop* and perform both quantitative and qualitative analysis in terms of predictive performance and domain-specific knowledge usage ability of the proposed model.

4.1 Evaluation Setup

Labelled Data Collection. We use the product review dataset[7] from Chen and Liu (2014), where each product (domain) has 1,000 unlabeled reviews. For each domain, we randomly sample 100 reviews, extract a list of (*mention*, *anaphor*) pairs from each of those reviews and label them manually with ground truths. That is, given a review text and a candidate (*mention*, *anaphor*) pair, we assign a binary label to denote whether they co-refer or not. In other words, we view each labeled example as a triple (u, m, p), consisting of the **context** u, a **mention** m and an **anaphor** p. Considering the review example (in Section 1), the triple ("*I bought ... loud and long*", "*a green Moonbeam*", "*its*") is a positive example, since "*a green Moonbeam*" and "*its*" refers to the **same entity** (i.e., they are

[7] https://www.cs.uic.edu/~zchen/downloads/ICML2014-Chen-Dataset.zip

in the same coreference cluster). Negative examples are naturally constructed by selecting m and p from two different clusters under the same context like ("*I bought ... loud and long*", "*a green Moonbeam*", "*its voice*").

Next, we randomly split the set of all labeled examples (for a given domain) into 80% for training, 10% as development, and rest 10% as test data. The remaining 900 unlabeled reviews form the *unlabeled domain corpus* is used for domain-specific knowledge extraction (as discussed in Section 3.1). All sentences in reviews and (*mention*, *anaphor*) pairs were annotated by two annotators independently who strictly followed the MUC-7 annotation standard (Hirschman and Chinchor, 1998). The Cohen's kappa coefficient between two annotators is 0.906. When disagreement happens, two annotators adjudicate to make a final decision. Table 2 provides the statistics of labeled dataset used for training, development and test for each of the five domains.

Knowledge Resources. We used three types of knowledge resources as listed below. The first two are general KBs, while the third one is our mined domain-specific KB.

1. Commonsense knowledge graph (OMCS). We use the open mind common sense (OMCS) KB as general knowledge (Speer and Havasi, 2013). OMCS contains 600K crowd-sourced commonsense triplets such as (*clock*, *UsedFor*, *keeping time*). We follow (Zhang et al., 2019b) to select highly-confident triplets and build the OMCS KG consisting of total 62,730 triplets.

2. Senticnet (Cambria et al., 2016). Senticnet is another commonsense knowledge base that contains 50k concepts associated with affective properties including sentiment information. To make the knowledge base fit for deep neural models, we concatenate SenticNet embeddings with BERT embeddings to extend the embedding information.

3. Domain-specific KB. This is mined from the unlabeled review dataset as discussed in Sec 3.1.

Hyper-parameter Settings. Following the previous work of (Joshi et al., 2019; Lee et al., 2018), we use (Base) BERT[8] embeddings of context and knowledge representation (as discussed in Section 3). The number of training epochs is empirically set as 20. We train five models on five datasets separately, because the domain knowledge learned from a certain domain may conflict with that from others. Without loss of generality and model extensibility, we use the same set of hyper-parameter settings for all models built on each of the five different domains. We select the best model setting based on its performance on the development set, by averaging five F1-scores on the five datasets. The best model uses maximum length of a sequence as 256, dropout as 0.1, learning rate as $3e^{-5}$ with linear decay as $1e^{-4}$ for parameter learning, and $\rho = 5.0$ (threshold for *tf-idf*) in domain-specific knowledge extraction (Section 3.1). The tuning of the other baseline models is the same as we do for our model.

Baselines. We compare following state-of-the art models from existing works on CR task:

(1) Review CR (Ding and Liu, 2010): A review-specific CR model that incorporates opinion mining based features and linguistic features.

(2) Review CR+BERT: For a fairer comparison, we further combine BERT with features from (Ding and Liu, 2010) as additional features. Specifically, we combine the context-based BERT to compute $F_C(m, p)$ (see Section 3.3 (a)).

(3) C2f-Coref (Lee et al., 2018): A state-of-the-art end-to-end model that leverages contextual information and pre-trained Glove embeddings.

(4) C2f-Coref+BERT (Joshi et al., 2019): This model integrates BERT into C2f-Coref. We use its *independent* setting which uses non-overlapping segments of a paragraph, as it is the best performing model in Joshi et al. (2019).

(5) Knowledge+BERT (Zhang et al., 2019b): This is a state-of-the-art knowledge-base model, which leverages different types of general knowledge and contextual information by incorporating an attention module over knowledge. General knowledge includes the aforementioned OMCS, linguistic feature and selectional preference knowledge extracted from Wikipedia. To have a fair comparison, we replace the entire LSTM-base encoder with BERT-base transformer.

To accommodate the aforementioned baseline models into our settings, which takes *context*, *anaphor*, and *mention* as input and perform binary classification, we change the input and output of the baseline models, i.e., the models compute a score between mention and anaphor and feeds the score to a sigmoid function to get a score within $[0, 1]$. Note, this setting is consistently used for all

[8] https://storage.googleapis.com/bert_models/2020_02_20/uncased_L-12_H-768_A-12.zip

Table 3: Performance (+ve F1 scores) of all models on all test datasets. Here, "cam", "com", "lap" are the abbreviation for "camera", "computer", "laptop" respectively.

Model	alarm	cam	phone	com	lap	average
Review CR	58.2	60.5	57.7	59.6	58.9	58.98
Review CR +BERT	67.2	69.3	67.0	68.4	66.7	67.72
C2f-Coref	68.8	70.1	67.2	69.5	67.4	68.60
C2f-Coref +BERT	70.2	71.6	68.6	71.3	68.2	69.98
Knowledge +BERT	72.0	73.4	71.8	72.6	70.0	71.96
Our model	**73.6**	**74.5**	**72.4**	**73.8**	**71.3**	**73.12**

candidate models (including our proposed model).

Evaluation Metrics. As we aim to solve the OAC2 problem, a focused coreference classification task, we use the standard evaluation metrics *F1-score* ($F1$), following the same setting of the prior study (Ding and Liu, 2010). In particular, we report positive (+ve) F1-score [F1(+)]. The average +ve F1-score is computed over five domains.

4.2 Results and Analysis

Comparison with baselines. Table 3 reports F1 scores of all models for each of five domains and average F1 over all domains. We observe the following: (1) Overall, our model performs the best considering all five domains, outperforming the no-knowledge baseline model C2f-Coref+BERT by 3.14%. On the cellphone domain, our model outperforms it by 3.8%. (2) Knowledge+BERT turns out to be the strongest baseline, outperforming the other three baselines, which also shows the importance of leveraging external knowledge for the OAC2 task. However, our model achieves superior performance over Knowledge+BERT which indicates leveraging domain-specific knowledge indeed helps. (3) C2f-Coref+BERT achieves better scores than C2f-Coref and Review CR. This demonstrates that both representation (using pre-trained BERT) and neural architectures are important for feature fusions in this task.

Ablation study. To gain further insight, we ablate various components of our model with the results reported in Table 4. For simplicity, we only show the average F1-scores on the five domain datasets. The results indicate how each knowledge resource or module contributes, from which we have the following observations.

1. From comparison Knowledge resources in Table 4, we see that domain knowledge contributes the most. General OMCS knowledge also contributes 0.84 to the model on average, so general knowledge is still needed. Senticnet contributes the least as it is more about sentiment rather than the relatedness between mentions. If we remove all knowledge sources (-all knowledge), performance drop becomes the highest which shows the importance of leveraging external knowledge in OAC2.
2. Considering comparisons of various types of scores in Table 4, we see that the disabling the use of context score F_C has the highest drop in performance, showing the importance of contextual information for this task. Disabling the use of knowledge scores F_G and F_{SG} also impact the predictive performance of the model, by causing a drop in performance.
3. From the comparison of attention mechanism for span representation in Table 4, we see that, before summing up the embedding of each word of the span, the attention layer is necessary. Note, we use the selected attention instead of popular dot attention in (Vaswani et al., 2017) during span representation. The influence of the syntax-based attention layer is slightly better than the dot attention layer. Therefore, we use the selected attention for better interpretability.

Table 4: Performance of our model with different types of knowledge or module removed (-). Δ F1(+) is the performance difference between our model and model with module remove.

Comparison	Model	Avg. F1(+)	Δ F1(+)
	Our model	73.12	0.00
Knowledge source	-OMCS knowledge	72.28	0.84
	-Domain knowledge	72.22	0.90
	-Senticnet	72.82	0.30
	-all knowledge	70.56	2.56
Score	-context F_c	71.14	1.98
	-knowledge F_K	71.80	1.48
	-phrase F_{SG}	72.58	0.56
attention	-syntax-based attention	72.50	0.62
	+dot attention	72.96	0.16

Qualitative Evaluation. We first give a real example to show the effectiveness of our model by comparing it with two baseline models C2f-coref+BERT and Knowledge+BERT. Table 5 shows a sample in the *alarm* domain. Here the major difficulty is to identify "Moonbeam" as a "clock". Knowledge+BERT fails due to its lack of domain-specific knowledge. C2f-coref+BERT

Table 5: A test example from alarm domain with class probability distributions by three models during prediction.

Context	...after I bought (a green Moonbeam for myself ... potential buyer also should know that , as with (the other Westclox clock), (the clock) also have (a gold band) ...
(Mention, Anaphor)	(*a darkgreen Moonbeam*, *the clock*)
Domain knowledge	drop, hang, clock, put, alarm, clear, beautiful, expensive, worthwhile ...
Our model	(0: 0.47, 1: 0.53)
Knowledge+BERT	(0: 0.87, 1: 0.13)
C2f-coref+BERT	(0: 0.79, 1: 0.21)

Table 6: An example showing the domain knowledge extraction quality of our model from laptop domain.

Mention (Domain)	windows (laptop)
Extracted knowledge (before filtering)	keep, like, product, battery, fast, microsoft, system, upgrade, xp, laptop..
Candidate knowledge (after filtering by tf-idf)	microsoft, system, upgrade, xp, laptop..

fails as well because it simply tries to infer from contextual information only, where there is no domain knowledge support. In contrast, with our domain-specific knowledge base incorporated, "Moonbeam" can be matched to the knowledge like "clock", "alarm", and "hang" which are marked with green color. So our model successfully addresses this case. In other words, in our model, not only the mention "a green Moonbeam" but also syntax-related phrase "a gold band" of "the clock" will be jointly considered in reasoning. We can see the modeling superiority of our knowledge-aware solution. Table 6 shows the effectiveness of our extraction module introduced in Section 3.1, especially the usage of tf-idf to filter out useless knowledge.

5 Conclusion

This paper proposed a knowledge-driven approach for object and attribute coreference classification in opinion mining. The approach can automatically extract domain-specific knowledge from unlabeled data and leverage it together with the general knowledge for solving the problem. We also created a set of annotated opinionated review data (including 5 domains) for object and attribute coreference evaluation. Experimental results show that our approach achieves state-of-the-art performance.

Acknowledgments

This work was supported in part by two grants from National Science Foundation: IIS-1910424 and IIS-1838770, and one research gift from Tencent.

References

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, pages 45–65.

Rahul Aralikatte, Heather Lent, Ana Valeria González-Garduño, Daniel Hershcovich, Chen Qiu, Anders Sandholm, Michael Ringaard, and Anders Søgaard. 2019. Rewarding coreference resolvers for being consistent with world knowledge. 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 1229–1235.

Muhammad Zubair Asghar, Aurangzeb Khan, Syeda Rabail Zahra, Shakeel Ahmad, and Fazal Masud Kundi. 2019. Aspect-based opinion mining framework using heuristic patterns. *Cluster Computing*, pages 7181–7199.

John Atkinson, Gonzalo Salas, and Alejandro Figueroa. 2015. Improving opinion retrieval in social media by combining features-based coreferencing and memory-based learning. *Information Sciences*, pages 20–31.

Daniel Bailey, Amelia J. Harrison, Yuliya Lierler, Vladimir Lifschitz, and Julian Michael. 2015. The winograd schema challenge and reasoning about correlation. In *2015 AAAI Spring Symposia*. AAAI Press.

Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. 2016. Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *COLING*, pages 2666–2677.

Zhiyuan Chen and Bing Liu. 2014. Topic modeling using topics from many domains, lifelong learning and big data. In *ICML*, pages 703–711.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. 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 5925–5932.

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*, pages 4171–4186.

Xiaowen Ding and Bing Liu. 2010. Resolving object and attribute coreference in opinion mining. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 268–276.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, pages 837–840.

Mauro Dragoni, Marco Federici, and Andi Rexha. 2019. An unsupervised aspect extraction strategy for monitoring real-time reviews stream. *Inf. Process. Manag.*, pages 1103–1118.

Ali Emami, Adam Trischler, Kaheer Suleman, and Jackie Chi Kit Cheung. 2018. A generalized knowledge hunting framework for the winograd schema challenge. In *NAACL-HLT Workshop*, pages 25–31.

Abbas Ghaddar and Philippe Langlais. 2016. Wikicoref: An english coreference-annotated corpus of wikipedia articles. In *LREC*, pages 136–142.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective attention modeling for aspect-level sentiment classification. In *COLING*, pages 1121–1131.

Lynette Hirschman and Nancy Chinchor. 1998. Appendix F: MUC-7 coreference task definition (version 3.0). In *Seventh Message Understanding Conference: Proceedings of a Conference Held in Fairfax, Virginia, USA*. ACL.

Jerry R Hobbs. 1978. Resolving pronoun references. *Lingua*, pages 311–338.

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*, pages 168–177. ACM.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and 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 5803–5808.

Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

Gourab Kundu, Avirup Sil, Radu Florian, and Wael Hamza. Neural cross-lingual coreference resolution and its application to entity linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 395–400.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 687–692.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, pages 1–167.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. DOER: dual cross-shared RNN for aspect term-polarity co-extraction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pages 591–601.

Nianzu Ma, Sahisnu Mazumder, Hao Wang, and Bing Liu. 2020. Entity-aware dependency-based deep graph attention network for comparative preference classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5782–5788.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Nicolas Nicolov, Franco Salvetti, and Steliana Ivanova. 2008. Sentiment analysis: Does coreference matter. In *AISB 2008 convention communication, interaction and social intelligence*, page 37.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1):9–27.

Jun Quan, Deyi Xiong, Bonnie Webber, and Changjian Hu. 2019. GECOR: an end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue. 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 4546–4556.

Robert Speer and Catherine Havasi. 2013. Conceptnet 5: A large semantic network for relational knowledge. In *The People's Web Meets NLP*, pages 161–176. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Shuai Wang, Mianwei Zhou, Sahisnu Mazumder, Bing Liu, and Yi Chang. 2018. Disentangling aspect and opinion words in target-based sentiment analysis using lifelong learning. *arXiv preprint arXiv:1802.05818*.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 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 Philip S. Yu. 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, NAACL-HLT*, pages 2324–2335.

Xintong Yu, Hongming Zhang, Yangqiu Song, Yan Song, and Changshui Zhang. 2019. What you see is what you get: Visual pronoun coreference resolution in dialogues. 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 5122–5131.

Hongming Zhang, Yan Song, and Yangqiu Song. 2019a. Incorporating context and external knowledge for pronoun coreference resolution. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 872–881.

Hongming Zhang, Yan Song, Yangqiu Song, and Dong Yu. 2019b. Knowledge-aware pronoun coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 867–876.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1253.