

A Survey on Aspect-Based Sentiment Classification

GIANNI BRAUWERS and FLAVIUS FRASINCAR, Erasmus University Rotterdam, The Netherlands

With the constantly growing number of reviews and other sentiment-bearing texts on the Web, the demand for automatic sentiment analysis algorithms continues to expand. Aspect-based sentiment classification (ABSC) allows for the automatic extraction of highly fine-grained sentiment information from text documents or sentences. In this survey, the rapidly evolving state of the research on ABSC is reviewed. A novel taxonomy is proposed that categorizes the ABSC models into three major categories: knowledge-based, machine learning, and hybrid models. This taxonomy is accompanied with summarizing overviews of the reported model performances, and both technical and intuitive explanations of the various ABSC models. State-of-the-art ABSC models are discussed, such as models based on the transformer model, and hybrid deep learning models that incorporate knowledge bases. Additionally, various techniques for representing the model inputs and evaluating the model outputs are reviewed. Furthermore, trends in the research on ABSC are identified and a discussion is provided on the ways in which the field of ABSC can be advanced in the future.

CCS Concepts: • **Information systems** → **Sentiment analysis**; **Data mining**; • **Computing methodologies** → **Machine learning**.

Additional Key Words and Phrases: aspect-based sentiment classification, knowledge-based models, deep learning, attention models, hybrid models

ACM Reference Format:

Gianni Brauwiers and Flavius FrasinCAR. 2021. A Survey on Aspect-Based Sentiment Classification. *ACM Comput. Surv.* 0, 0, Article 0 (2021), 35 pages. <https://doi.org/doi-number>

1 INTRODUCTION

The World Wide Web has provided a means for people to share their opinions online through various text-based channels, such as online reviews and social media. Being able to use these opinions to accurately assess what people think of a certain product, person, or place is highly valuable in many industries. Restaurants can adjust their menus based on online food reviews, companies can improve their products based on what consumers exactly want, and the effects of political campaigns can be evaluated by analysing social media posts. As such, with the rise of the Internet, the task of sentiment analysis has become more and more significant [149].

Sentiment analysis is the task of extracting and analysing people’s sentiments towards certain entities from text documents [118]. Sentiment analysis is sometimes also referred to as opinion mining in the literature. Yet, a distinction must be made between “sentiments” and “opinions”. Namely, opinions indicate a person’s views on a specific matter, while sentiment indicates a person’s feelings towards something. Yet, the two concepts are highly related and opinion words can typically be used to extract sentiments [94, 95]. In sentiment analysis, the goal is to assign sentiment polarities

Authors’ address: Gianni Brauwiers, gianni.brauwiers@gmail.com; Flavius FrasinCAR, frasincar@ese.eur.nl, Erasmus University Rotterdam, Erasmus School of Economics, The Netherlands, Rotterdam, P.O. Box 1738, 3062 PA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

Manuscript submitted to ACM

Manuscript submitted to ACM

53 based on a body of text. Although the words “polarity” and “sentiment” are often used interchangeably, a distinction
54 can be made again. Namely, a sentiment indicates a feeling, while the polarity expresses the orientation (e.g., “positive”,
55 “neutral”, or “negative”). The granularity of a sentiment analysis task can be described using three separate characteristics:
56 the sentiment type, the level of the task, and the target. First, the type of sentiment output must be defined. Namely, the
57 task can entail a simple binary classification (“positive” or “negative” labels), but there could also be additional labels
58 (“neutral”), or even a sentiment intensity or score that must be predicted. Next, the level of the task determines at
59 which level of the text sentiment is extracted. For example, in document-level sentiment analysis, the sentiment is
60 analysed for the entire document. For sentence-level sentiment analysis, the sentiment is analysed for each sentence
61 within the document. There are many other levels, such as words, paragraphs, groups of sentences, or chunks of text.
62 Lastly, the target of the task determines the focus of the sentiment. First, there may be no sentiment target, which
63 means that the task entails assigning a sentiment score or label to the text itself. On the other hand, one may want to
64 know the sentiment regarding specific topics, entities, or aspects within the text. To illustrate, suppose we analyse a
65 product review. If we do not define a target, we simply consider the sentiment of the entire text. Yet, knowing exactly
66 which aspects of the product the consumer is satisfied or dissatisfied with can be highly useful, as it provides detailed
67 information necessary for many applications [115]. The task of identifying aspects and analysing their sentiments in
68 texts is known as aspect-based sentiment analysis (ABSA).

73 ABSA is a relatively new field of research that has gained significant popularity due to the many useful applications
74 [32]. However, ABSA and sentiment analysis generally are difficult tasks due to the general structures of sentences and
75 writing styles [174]. In [143], a comprehensive overview is provided of the issues and challenges present in ABSA. Like
76 general sentiment analysis tasks, ABSA can be performed at multiple levels of which the document level and sentence
77 level are typically the most popular [157]. Document-level ABSA methods are concerned with finding general aspects
78 linked to a certain entity in the document and assigning sentiments to them. Sentence-level ABSA approaches, on the
79 other hand, attempt to identify all aspects for each sentence individually, determine the sentiment associated with
80 these aspects, and possibly aggregate sentiment at the review-level [157]. As such, document-level ABSA considers the
81 general concepts that summarize the sentiment in the text, while sentence-level ABSA considers each mention of an
82 aspect individually. The task of ABSA can be further split up into three tasks: aspect detection/extraction, sentiment
83 classification, and sentiment aggregation [174]. Aspect extraction involves identifying the aspects present in a text, the
84 classification step consists of assigning a sentiment label or score to the extracted aspects, and the aggregation step
85 amounts to summarizing the aspect sentiment classifications. In this survey, we focus on the sentiment classification
86 step, which is generally referred to as aspect-based sentiment classification (ABSC).

91 Various surveys on ABSA have already been published [152, 174]. However, it seems that a survey focused specifically
92 on ABSC would be more effective at providing an in-depth discussion and evaluation of ABSC models. The only survey
93 dedicated solely to ABSC is [230], which provides an overview of deep learning techniques. While deep learning models
94 are currently state-of-the-art for ABSC, we would argue that a broader scope would allow for a more effective evaluation
95 of the current and future state of research in ABSC. For example, a significant part of ABSC is the research on approaches
96 that incorporate knowledge bases into classification models. Furthermore, various important deep learning models,
97 such as transformer-based models [201], are missing from previous surveys. For these reasons, this survey presents
98 a comprehensive overview of the state-of-the-art ABSC models. For this purpose, we propose a novel taxonomy of
99 ABSC models, which is currently missing from the literature. The taxonomy categorizes ABSC models into three major
100 categories: knowledge-based methods, machine learning models, and hybrid models. Based on the structure of this
101 taxonomy, we discuss and compare the architectures of the various ABSC models using both technical and intuitive
102
103
104

105 explanations. Furthermore, we also provide summarizing overviews of the performances of various ABSC models on a
106 larger scale than previous surveys. Last, we identify trends in the research on ABSC and use these findings to discuss
107 ways in which the field of ABSC can be advanced in the future.

108 The sections of this survey are based on the main steps taken when designing ABSC methods. In Section 2, we discuss
109 the different ways of representing the inputs for ABSC. In Section 3, techniques for evaluating the model performance
110 are reviewed. In Section 4, various ABSC models are presented according to a proposed taxonomy. We discuss how
111 these models use the preprocessed inputs to produce the desired classification output, and compare the performances
112 of various models found in the literature. In Section 5, additional topics related to ABSC are discussed. In Section 6, we
113 give our conclusions and discuss directions for future research.
114
115
116

117 2 INPUT REPRESENTATION

119 In this section, we explain in detail the input representations necessary for ABSC. We start by introducing some
120 definitions in accordance with Schouten and Frasinca [174]. Given a corpus C containing the records R_1, \dots, R_{n_R} ,
121 ABSC can be formally defined as finding all quadruples (y, a, h, t) [118] for each record R_j , where y is the sentiment,
122 a represents the target aspect for the sentiment, h is the holder, or the individual expressing the sentiment, and t
123 represents the time that the sentiment was expressed [174]. A record is defined as an individual piece of text in the
124 corpus, which can be a single phrase, a sentence, or a large body of text, like a document. Furthermore, in general, most
125 methods are concerned with finding (y, a) , the aspects and the corresponding sentiments. Since this survey is only
126 focused on the classification step of ABSC, we assume the target aspects a are already identified in the text. As such,
127 ABSC models are focused only on finding the sentiments y corresponding to the given aspects a . For example, consider
128 the following record that takes the form of a restaurant review: *"The atmosphere was fantastic, but the food was bland"*.
129 This sentence contains two aspects *"atmosphere"* and *"food"*, which we assume have been identified in a prior aspect
130 extraction phase. The eventual goal is to use an ABSC model to determine a sentiment classification for each of these
131 aspects. The correct sentiments, in this case, would be *"positive"* and *"negative"*, respectively. However, before one can
132 attempt this task, input representations need to be constructed.
133
134
135
136

137 Since a piece of text can generally not directly be used as input for a classification model, the aspect and corresponding
138 context must be represented through numeric features. Note that, in aspect-based, or feature-based, sentiment analysis,
139 the term 'features' is sometimes used to describe the aspects of, for example, a product. However, 'features' in this
140 context refers not to the aspects themselves, but to the features of the data. Before an ABSC model can be implemented,
141 a preprocessing phase is necessary to construct a numeric representation [70, 80]. The features that are used as inputs
142 are a crucial part of the classification process since they determine what information the ABSC model has access
143 to. It is important to understand how to represent a piece of a text in such a way that the classification model has
144 access to as much information as possible without including redundant or irrelevant features such that it can perform
145 optimally. Input representations for ABSC generally consist of three characteristics: **context**, **dimensionality**, and
146 **feature types**. We discuss these in Subsection 2.1, Subsection 2.2, and Subsection 2.3, respectively.
147
148
149
150

151 2.1 Context

152 Given a record R_j , we define the **context** as the subset of words that is considered as the input. If the text only contains
153 a single aspect, then one can choose to simply consider all words. However, if the record contains multiple aspects, a
154 representation needs to be developed for each aspect individually by, for example, taking a subset of the words for each
155
156

157 aspect. The methods of representing the input for each aspect depend on whether a target phrase is present in the text.
 158 As such, we make a distinction in the input representation techniques for **explicit** and **implicit** aspects.

159 Explicit aspects are generally the most common type of aspects. An example of such an aspect can be seen in the
 160 review sentence “*The price of this phone is very high.*”, where the aspect is explicitly stated to be “*price*”. The simplest
 161 method of determining the context for such an explicit aspect is by employing a window around the target phrase and
 162 developing an input representation based only on the words in the window. For example, Guha et al. [77] only consider
 163 the aspect itself, the three words to the left of the aspect, and the three words to the right of the aspect. However,
 164 such methods based on the physical proximity may not be optimal since the words expressing the sentiment may be
 165 far removed from the aspect. As such, more robust methods do not rely on the physical distance between words. For
 166 example, one can use the grammatical dependencies to determine which words in the record are related to the aspect
 167 and should therefore be considered in the context [193]. Another technique for determining the context is the use of
 168 text kernels to express the distances based on the relations between words. Nguyen and Shirai [144] implement tree
 169 kernels for relation extraction to determine which words they consider in the analysis.

170 An example of a sentence with an implicit aspect is “*This phone is really expensive.*”, where the aspect is again “*price*”,
 171 but the aspect is not directly mentioned in the text. Implicit aspects must be treated differently since we do not have a
 172 target phrase to center a window around or to determine distances to words with. If we assume the record contains
 173 only one aspect, such as in the given example sentence, then this is not a major problem as one can simply consider
 174 the whole sentence for the input representation. However, as Dosoula et al. [52] argue, reviews may often contain
 175 multiple implicit aspects, even within the same sentence. For instance, the previously given example can be extended
 176 with another implicit aspect: “*This phone is really expensive, but also very fast.*”. Dosoula et al. [52] present different
 177 methods based on aspect proxy words to determine the context for each aspect in a sentence.

184 2.2 Dimensionality

185 The **dimensionality** of the input representation is directly determined by the type of model used in the subsequent
 186 sentiment classification step. Some models, such as support vector machines and decision trees, work only with a single
 187 vector representation, whereas other models, such as recurrent neural networks, can process a set of vectors or a data
 188 matrix. Suppose record R_j contains a single aspect and we wish to represent the entire record using numeric features.
 189 Then, depending on the ABSC models used, record R_j can be represented using a single vector $x_j \in \mathbb{R}^{d_x}$, or a matrix
 190 $X_j \in \mathbb{R}^{d_x \times n_j}$, where d_x represents the number of features used for the representation, and n_j represents the number of
 191 words in record R_j . A single vector representation of the record can be interpreted as a record embedding where each
 192 element of the vector indicates the presence of a feature. For example, whether certain sentiment-indicating words or
 193 phrases are present. On the other hand, a data matrix can provide more detailed information since it does not have
 194 to summarize all information into a single vector. Each column of the matrix is an embedding containing features
 195 representing a part of the record, such as a sentence, word, or character. The dimensionality of the inputs generally
 196 influences the types of features used, as some feature types work better with a vector representation, whereas others
 197 can be used more effectively with a matrix representation. The next subsection discusses this in further detail.

202 2.3 Feature Types

203 When representing text as a single vector, classical text classification **feature types** can be used. Arguably one of the
 204 simplest and most popular ways of representing a sentence or piece of text is via a bag-of-words (BoW) representation.
 205 A BoW vector representation is a feature vector where each element represents a word. It is called a “bag” because we
 206

Table 1. Overview of various types of features.

Feature Types	Feature Examples
Word Features	Bag-of-words [177], word embeddings [132], n -grams [148]
Syntactic Features	Part-of-speech tagging [54, 162] and dependency parsing [54, 111, 140, 173]
Proximity-Based Features	Proximity relative to sentiment words [82, 141] or target words [100]
Semantic Features	Sentiment scores of context words [83, 198]
Morphological Features	Lexemes and lemmas [1, 6, 173]

simply congregate the words together and disregard the structure of the text. Again, suppose we wish to represent record R_j using a single vector \mathbf{x}_j . We start by building a vocabulary that contains every unique word used in any of the records in our corpus C . For a BoW representation, each element $x_{i,j}$ in vector \mathbf{x}_j represents a unique word from the vocabulary. The simplest BoW vector is a binary vector where each element $x_{i,j}$ in vector \mathbf{x}_j is set equal to 1 if the corresponding word occurs in R_j , and 0 otherwise. A more popular version is to also consider the frequency of the words by setting each element $x_{i,j}$ equal to the number of times the corresponding word occurs in R_j [169]. However, some words are simply used more often than others, which can cause a bias towards certain words in the word counts. Thus, instead of using the direct word counts, one can also implement the term frequency-inverse document frequency (TF-IDF) measure [103]. This measure is essentially the word frequency in a record scaled by how often the word appears in all records. For word w_i in the vocabulary, the TF-IDF can be calculated using Equation (1).

$$x_{i,j} = \frac{n_{i,j}}{n_j} \times \log \frac{n_R}{|\{R_j \in C : w_i \in R_j\}|}, \quad (1)$$

where $n_{i,j}$ denotes the number of times word w_i occurs in record R_j , n_j represents the number of words in record R_j , n_R denotes the total number of records, and C indicates the set of records. As previously mentioned, for ABSC, one should preferably not use a singular text representation when multiple aspects are present in the record. As such, one can simply create a BoW representation based only on the words considered to be in the context.

BoW features work well for ABSC but may come with certain problems that make the classification process difficult. Firstly, a large corpus can contain a substantial number of different words, meaning that the vocabulary and corresponding feature vectors will be very large if one would represent every single word with an element. Thus, methods have been developed to select the most important words that can be used for features. For example, based on a list or dictionary, one can filter out the words that generally hold no meaningful or semantic value, such as stop words. Even though such techniques can significantly reduce the number of features, the vectors generally remain large. Because of this, an important characteristic of ABSC models is that they should typically be able to handle high-dimensional vectors well, which is discussed in more detail in Section 4. Secondly, as previously mentioned, the structure of the words is completely disregarded, making it difficult to capture relations between the aspect and its context. One solution is to include n -grams, but this further exacerbates the previously mentioned high dimensionality problem. Another solution is to include features that define the relations between the words. Additional feature types are generally used and can take a variety of forms to include different types of information. For example, Al-Smadi et al. [6] enhance a TF-IDF BoW representation with additional morphological, syntactic, and semantic features. Similarly, Mullen and Collier [141] implement semantic, syntactic, and proximity-based features. Table 1 provides an overview of a variety of feature types with the corresponding examples.

261 Suppose we now wish to represent the record R_j using a matrix X_j . In this case, the idea is that each word in the
262 text is represented by a vector, which is stored as a column in X_j . A standard choice, in this case, is the use of word
263 embeddings. For an overview of the many types of word embeddings, we refer to [27]. Some examples are: *GloVe* [153],
264 *fastText* [73], and *Word2Vec* [133]. Each of these embedding types attempts to represent a word's meaning using a vector
265 of limited size. While pre-trained word embeddings are often used, it is also possible to train word embeddings along
266 with the ABSC model. Additionally, the previously mentioned embedding models are all non-contextual, which means
267 that an embedding is produced for each word independently from the other words. Yet, it is known that a word can have
268 multiple meanings depending on the context that it is used in. Therefore, another solution is to use contextual word
269 embeddings, such as embeddings based on *ELMo* [154] or *BERT* [48]. These word embedding models produce different
270 vectors for words depending on the surrounding context. For example, the word "playing" has a different meaning in
271 "playing tennis" than in "playing piano" and will therefore get a different word embedding assigned to it. Furthermore,
272 there are also word embeddings made specifically for sentiment analysis [189]. Word embeddings are a powerful
273 representation tool on their own, but one can still add additional feature types like the examples provided in Table 1.
274 For example, in [7], word embeddings are enhanced with syntactic and semantic features. The input representations
275 discussed in this section are the basis for the classification models discussed in Section 4. In the rest of this paper, we
276 assume that input representations have already been constructed beforehand. We only further elaborate on the input
277 representation if the classification model modifies the input representations. Furthermore, in the rest of this paper, for
278 clarity purposes, we do not use record-specific or aspect-specific subscripts.
279
280
281
282
283
284

285 3 PERFORMANCE EVALUATION

286 The goal of an ABSC model is to produce an output using the input representations. Thus, given a record R containing
287 an aspect a , an ABSC model produces a label $\hat{y} \in \mathbb{R}^1$ using the feature vector $\mathbf{x} \in \mathbb{R}^{d_x}$ or matrix $X \in \mathbb{R}^{d_x \times n_x}$, where d_x
288 represents the number of features used, and n_x indicates the number of words considered to be in the context. The
289 effectiveness of ABSC models can be compared by evaluating the output \hat{y} using a variety of performance measures.
290 Each of these measures highlights certain strengths and weaknesses of the classification models. In this section, we
291 represent various techniques for evaluating ABSC models. These measures are used to compare and contrast the various
292 ABSC approaches in Section 4. The most commonly used performance measures for ABSC and ABSA, in general, are
293 the well-known accuracy, precision, recall, and F_1 -measure [174]. These measures evaluate the sentiment classification
294 \hat{y} by comparing it to the true sentiment label y of the aspect. Models that achieve high values for these performance
295 measures, will produce higher-quality predictions. The accuracy measures the ratio of correctly classified aspects to the
296 total number of aspects in the dataset. This is intuitively clear, as it is a direct indication of how well the model predicts.
297 In a binary classification setting, a simple baseline to use is a random classifier. A fully random classifier would obtain
298 an expected accuracy of 0.5, which any predictive model should be able to outperform. Yet, the accuracy measure is
299 often not a valid indicator of performance in imbalanced datasets. If a dataset contains two classes, where 90% of the
300 records are of one class, then always predicting that class will provide a high accuracy of 0.9. As such, other measures
301 like the precision and recall are used that provide more useful evaluations of classification performance in imbalanced
302 datasets. One can aggregate the precision and recall for each class to obtain a measure of the model performance. This
303 can be done by evaluating the mean of the measures (macro-averaging), or by aggregating based on the contributions
304 of all classes (micro-averaging). A similar process can be used when aggregating the results from multiple different
305 datasets. One can either take the mean of the performance measures from the various datasets (macro-averaging), or
306
307
308
309
310
311
312

one can aggregate based on the contributions of the datasets (micro-averaging). As the precision and recall focus on different parts of the model performance, the F_1 -measure is typically used to summarize the information.

The accuracy, precision, recall, and F_1 -measure are used in most of the works discussed in this survey. However, there are various alternative measures that are occasionally used as well. For example, the mean squared error (MSE) and the ranking loss. Suppose we have a training dataset consisting of the N feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$ with the corresponding labels y_1, \dots, y_N . Each of the feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$ represents an aspect with the corresponding context in a record, as explained in Section 2. The corresponding labels y_1, \dots, y_N indicate the true sentiment expressed towards the aspects. An ABSC model is used to produce the label predictions $\hat{y}_1, \dots, \hat{y}_N$. Since the labels for sentiment analysis are generally considered to be ordinal, the MSE can be calculated as follows:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2. \quad (2)$$

Due to the square, the MSE penalizes large errors more than small errors. An alternative is the ranking loss [40] that punishes small and large errors in a more equal manner. The ranking loss measure is closely related to the mean absolute error and can be calculated according to Equation (3).

$$\text{Ranking Loss} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|. \quad (3)$$

Both the Ranking Loss and the MSE measures are used to measure the errors in the classification predictions. As such, the lower the values attained for these measures, the better the model performs. A final performance measure is the area under the Receiver Operating Characteristics (ROC) curve, abbreviated as AUC. The ROC curve plots the recall against the true positive rate. The area under the curve (AUC) is then a measure of how well the model can separate classes, meaning that the higher the AUC value, the better the performance of the model.

4 SENTIMENT CLASSIFICATION

ABSC models can generally be classified into three major categories: knowledge-based approaches, machine learning models, and hybrid models. In Figure 1, a taxonomy consisting of these categories and their sub-categories is presented. The taxonomy categories are explained in more detail in Subsection 4.1, Subsection 4.2, and Subsection 4.3, respectively. Each of these subsections contains an overview of various prominent ABSC models for that model type in the form of a summarizing table. Each table consists of columns detailing the model, the types of data used, and the various performance measures reported for every model. A row corresponding to a model can consist of multiple entries if datasets from different domains are used. Additionally, we report results from other works that implement the same model using different datasets. When a model is reimplemented in another paper, we indicate this beneath the entry. Unavailable information is indicated by a “-”. Since we are unable to include all results reported by each paper, we take several steps to summarize the results. First of all, we only include the results of the best model architecture presented in each paper. Secondly, when multiple datasets are used in the same domain, we average the results across those datasets. The only exception is when a model is reimplemented in another paper since we cannot guarantee that the model implementation is the same. We have included the references for all datasets in the tables for easier model comparisons. General characteristics of some ABSC datasets are provided in Table 2. Note that this table only includes the most used domains and languages from the listed datasets. For example, the SemEval-2016 dataset contains many additional languages, such as Dutch, Chinese, and Turkish.

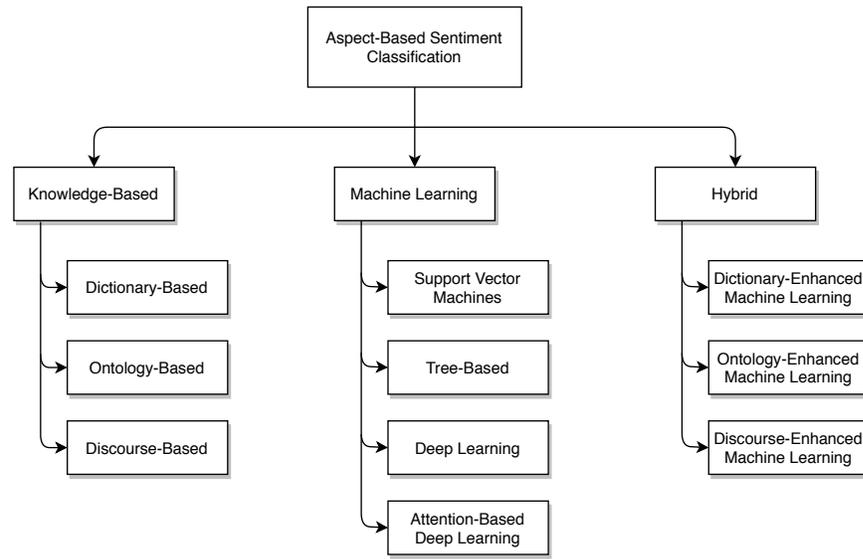


Fig. 1. Taxonomy of ABSC methods.

Table 2. Overview of datasets used for ABSC.

Dataset	Domain(s)	Language(s)	Positives	Neutrals	Negatives	Total
SemEval-2014 [159]	Reviews (Electronics)	English	1328	629	994	2951
	Reviews (Restaurants)	English	2894	829	1001	4724
SemEval-2015 [158]	Reviews (Electronics)	English	1644	185	1094	2923
	Reviews (Restaurants)	English	1652	98	749	2499
	Reviews (Hotels)	English	243	12	84	339
SemEval-2016 [157]	Reviews (Electronics)	English	1540	154	869	2563
	Reviews (Restaurants)	English	1802	104	623	2529
	Reviews (Hotels)	Arabic	7705	852	4556	13113
Twitter Data [51]	Social Media (Various)	English	1735	3470	1735	6940
SentiHood [168]	Social Media (Neighborhoods)	English	2460	0	1216	3676
Hindi Reviews [3]	Reviews (Various)	Hindi	1986	1914	569	4469
Indonesian Marketplace [61]	Reviews (Various)	Indonesian	7982	0	1677	9659
FiQA-2018 [126]	Social Media (Finance)	English	722	13	378	1113

4.1 Knowledge-Based

Knowledge-based methods, also known as symbolic AI, are approaches that make use of a knowledge base. Knowledge bases are generally defined as stores of information with underlying sets of rules, relations, and assumptions that a computer system can draw upon. Knowledge-based methods are heavily related to the input representation since these approaches often use a knowledge base to define features. One of the advantages of knowledge-based methods is interpretability. Namely, it is generally easy to identify the information used to produce the model output. The underlying mechanisms of knowledge-based methods are typically relatively simple, which allows for highly transparent approaches to ABSC. Knowledge-based methods require no training time, but the construction of knowledge bases can take considerable time. We discuss the following three knowledge-based approaches: dictionary-based, ontology-based, and discourse-based methods. The performances of various knowledge-based methods are displayed in Table 3.

Table 3. Overview of prominent knowledge-based ABSC models and their reported performances. If no performance measures are displayed, then no quantitative analysis was provided for ABSC in the original paper.

Reference	Category	Domain(s)	Accuracy	Precision	Recall	F ₁	Alternative Measures
Hu and Liu (2004) [94]	Dictionary-Based	Reviews (Electronics) [10, 37]	0.842	0.642	0.693	0.667	-
Zhu et al. (2009) [232]	Dictionary-Based	Reviews (Restaurants) [49]	-	0.755	0.755	0.755	-
Moghaddam and Ester (2010) [137]	Dictionary-Based	Reviews (Electronics) [60]	-	-	-	-	Ranking Loss: 0.49
Eirinaki et al. (2012) [58]	Dictionary-Based	Reviews (Various) [119]	-	-	-	-	-
Zhou and Chaovalit (2008) [231]	Ontology-Based	Reviews (Movies) [192]	0.722	-	-	-	-
Kontopoulos et al. (2013) [108]	Ontology-Based	Social Media (Electronics) [199]	-	-	-	-	-
Nie et al. (2013) [146]	Ontology-Based	Reviews (Electronics) [99]	-	0.587	0.625	0.605	-
Hoogervorst et al. (2016) [91]	Discourse-Based	Reviews (Electronics) [158] Reviews (Restaurants) [158, 159]	-	0.670 0.670	0.670 0.670	0.670 0.670	-
Sanglerdsinlapachai et al. (2016) [170]	Discourse-Based	Reviews (Electronics) [37]	0.633	-	-	-	-
Dragoni et al. (2018) [54]	Discourse-Based	Reviews (Electronics) [158] Reviews (Restaurants) [158]	0.859 0.779	- -	- -	- -	- -

4.1.1 Dictionary-Based. Early methods for ABSC were mostly based on dictionaries. Given a record R containing an aspect a , dictionary-based methods construct a feature vector \mathbf{x} using a dictionary, where each element x_i represents a sentiment score, or orientation, of a word in the context regarding the aspect. Various dictionaries can be used for ABSC, such as *WordNet* [134] and *SentiWordNet* [14]. These dictionaries define sets of words and the linguistic relations between them. For example, *WordNet* [134] groups nouns, verbs, adjectives, and adverbs into so-called *synsets*: sets of synonyms, or groups of words that have the same meaning. One can exploit these relations for ABSC since words generally portray the same sentiment as their synonyms [94]. As such, we start with a set of *seed words* for which the sentiment is known. For example, one can use “good”, “fantastic”, and “perfect” as positive seed words, and “bad”, “boring”, and “ugly” as negative seed words. These seed words can then be used to determine the sentiment of words surrounding an aspect via the synsets defined in a dictionary, like *WordNet* [134]. Words that are synonyms of a positive seed word, or antonyms of a negative seed word, will receive a positive sentiment score. *SentiWordNet* [14] is in part based on this idea. This dictionary generates a sentiment label (“positive”, “neutral”, or “negative”) for each of the synsets using a combination of a seed set expansion approach and a variety of classification models.

After determining sentiment scores for the context words, a method needs to be implemented to determine a sentiment output. For example, the authors of [94] examine whether there are mostly negative or positive words present in the context. In [94], the sentiment scores for the context words are encoded as 1 for positive, -1 for negative, and 0 for neutral. Then, the sentiment classification can be determined by summing the elements of \mathbf{x} . If the sum is positive, a positive label is returned, and a negative label otherwise. Similarly, in [58], the context words are assigned a sentiment score in the range $[-4, 4]$. The sentiment polarity of the aspect is then determined based on the average score per opinion word relating to the aspect.

4.1.2 Ontology-Based. An ontology is generally defined as an “explicit, machine-readable specification of a shared conceptualization” [74, 182], and defines a set of entities and relations corresponding to their properties. The main difference between an ontology and a dictionary is that a dictionary captures linguistic relations between words, while an ontology represents relations between real entities. These relations can be used for determining which words in the record are important for determining the sentiment towards the aspect. For ABSC, one can either use an existing

469 ontology or create one based on the domain at hand [108]. Examples of existing ontologies are the semantically
470 interlinked web communities (SIOC) ontology [181], which is an ontology that captures data from online community
471 websites, and the ontology of emotions proposed in [165]. Yet, rather than relying on existing ones, most researchers
472 choose to create their own ontology. This is because finding an ontology that relates well to a specific domain is difficult
473 since many existing ontologies typically capture rather generic concepts. Designing ontologies is often done using
474 specific methods, such as formal concept analysis [147] or the *OntoClean* methodology [76]. Ontologies are often
475 manually created to ensure the accuracy of the captured relations [131], but this process is rather time-consuming. As
476 such, methods for creating ontologies semi-automatically or even completely automatically can be vital when creating
477 larger ontologies. For example, in [38], an ontology for ABSC is created semi-automatically using a proposed semantic
478 asset management workbench (SAMW). A method for fully automatically creating ontologies is proposed in [8].

481 Ontologies capture the structure of objects in a domain. These relations can be used to determine the context
482 corresponding to the aspect [108]. Then, a method can be used to determine the sentiment label using the context
483 obtained using the ontology. For example, by using a dictionary-based sentiment classifier [231]. While regular ontologies
484 are useful tools for defining relations between objects, to further facilitate sentiment analysis, sentiment information
485 can be incorporated into the ontology. Such sentiment ontologies specifically define sentiment relations between words
486 or entities. For example, in [146], a sentiment ontology tree is constructed that links product aspects to opinion words
487 with the corresponding sentiment scores. In [215], a sentiment ontology is constructed where the sentiment relations
488 are defined using the *SentiWordNet* dictionary [14]. Zhuang et al. [235] propose a semi-automated ontology builder for
489 ABSA that constructs an ontology based on sentiment relations. When classifying an aspect, one can link the words in
490 the context to the concepts and relations in the sentiment ontology, and summarize the sentiment relations to produce
491 a sentiment classification. The semi-automatic method proposed in [235] mainly focuses on word frequencies, due
492 to the limited amount of domain data. In [46], another semi-automatic ontology builder is proposed that is based on
493 synsets from the *WordNet* [134] dictionary. Lastly, ten Haaf et al. [190] propose a semi-automatic ontology builder that
494 is based on word embeddings produced using the *word2vec* [133] method.

500 *4.1.3 Discourse-Based.* Another type of knowledge base that can be used for ABSC is a discourse tree based on
501 rhetorical structure theory (RST) [128]. RST can be used to define a hierarchical discourse structure within a record to
502 categorize phrases into elementary discourse units (EDU). Similar to ontologies, discourse trees define sets of relations
503 that can be used to determine which words are important when assigning a sentiment classification to the aspect. For a
504 record R , a discourse tree is constructed that defines the hierarchical discourse relations in the record. To determine the
505 context concerning an aspect a in the record, the authors of [91] use an RST-based method that produces a sub-tree of
506 the discourse tree that contains the discourse relations specifically regarding the aspect. This sub-tree is named the
507 context tree and can be used to determine the sentiment classification of the aspect. Since the context tree does not
508 contain any sentiment information, the authors of [91] use a dictionary-based method to label the leaf nodes of the
509 context tree with sentiment orientation scores for sentence-level ABSC. To determine the sentiment classification of
510 the aspect, one can simply evaluate the sum of the sentiment scores defined for the context tree. In [170], a similar
511 technique is used with a more sophisticated method of aggregating the scores.

515 There are other types of discourse structure theories besides RST [93]. One example of this is cross-document
516 structure theory (CST) [160], which can be used to analyse the structure and discourse relations between groups of
517 documents. Cross-document structure analysis can be useful in social media analysis since social media posts are short
518 documents that are typically highly interrelated. An interesting example of this is the SMACK system [53], which

521 analyses a cross-document structure based on abstract argumentation theory [56]. For example, products often receive
 522 multiple reviews from various users that may respond to each other. The SMACk system analyses the relations between
 523 the different arguments presented in the reviews to improve the aggregation of the sentiment expressed towards the
 524 different aspects [54].
 525

527 4.2 Machine Learning

528 As opposed to knowledge-based methods that exploit knowledge bases to achieve a sentiment classification, machine
 529 learning models, also known as subsymbolic AI, use a training dataset of feature vectors and corresponding correct
 530 labels. The machine learning model is trained to extract patterns from the data that can be used to distinguish between
 531 sentiment classes. There are many types of machine learning models, which can be grouped into the following categories:
 532 support vector machines, tree-based models, deep learning models, and attention-based deep learning models. The
 533 performances of various machine learning models are presented in Table 4.
 534

535 *4.2.1 Support Vector Machines.* Support vector machine (SVM) models [39] have long been a popular choice for
 536 sentiment analysis and ABSC [7, 141, 150, 151, 200]. SVM models are classifiers that distinguish between categories
 537 by constructing a hyperplane that separates data vectors belonging to the different classes [26, 39, 200]. In the case
 538 of ABSC, this comes down to separating aspects into sentiment classes (“positive”, “neutral”, and “negative”) based
 539 on the feature vector \mathbf{x} . Suppose we have a training dataset consisting of the N feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$ with the
 540 corresponding labels y_1, \dots, y_N . Each of the feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$ represents an aspect with the corresponding
 541 context in a record, as explained in Section 2. The corresponding labels y_1, \dots, y_N indicate the true sentiment expressed
 542 towards the aspects. We first examine the case when only two sentiment classes are considered: “positive” and “negative”.
 543 The labels for aspects for which a positive sentiment is expressed are encoded as a 1, while aspects toward which a
 544 negative sentiment is expressed receive a label of -1. The SVM classifier can then be summarized as follows:
 545

$$546 \hat{y} = \text{sign}(\underbrace{\mathbf{w}^T}_{1 \times 1} \times \underbrace{\phi(\mathbf{x})}_{d_z \times 1} + \underbrace{b}_{1 \times 1}), \quad (4)$$

547 where $\mathbf{w} \in \mathbb{R}^{d_x}$ is a learned vector of weights and $b \in \mathbb{R}^1$ is a learned bias constant. The weights are determined by
 548 constructing a hyperplane that maximizes the separation between the training data labels y_1, \dots, y_N based on the feature
 549 vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$. While some datasets can be separated using a linear function form, other problems may not be so
 550 easily separable. In such situations, the kernel function $\phi(\cdot)$ can be used to transform the feature vector \mathbf{x} into a higher
 551 dimension where the labels are separated more easily [171]. The disadvantage of using a kernel function is that the
 552 learned coefficients become difficult to interpret due to the non-linearity. When an aspect can be classified into multiple
 553 sentiment categories, the SVM model must be adjusted. An example solution is a “one-versus-all” implementation,
 554 where an SVM model is trained to separate each class versus all other classes. The final prediction is based on which
 555 decision function has the highest value. SVMs are known to generalize well and be robust towards noisy data [217].
 556 Furthermore, as mentioned in Section 2, feature vectors for ABSC can often consist of substantial amounts of features,
 557 which SVM models are known to handle well [102]. However, finding a kernel that works is generally a difficult task
 558 [26], and hand-crafted features are required for the model to perform well [7].
 559

560 *4.2.2 Tree-Based.* Tree-based approaches are methods based on the trainable decision tree [23] model. A decision tree
 561 model consists of a tree-like structure where each internal node, or decision node, in the tree represents a condition
 562 based on a particular feature from the vector \mathbf{x} , and each leaf node represents a particular sentiment class. Given
 563

Table 4. Overview of prominent machine learning ABSC models and their reported performances. DT = "Decision Tree" and RF = "Random Forest".

Reference	Category	Domain(s)	Accuracy	Precision	Recall	F ₁	Alternative Measures
Jiang et al. (2011) [101]	SVM	Social Media (Various) [199]	0.682	-	-	-	-
		Social Media (Various) [51] (reimplemented in [203])	0.634	-	-	0.633	-
Yu et al. (2011) [225]	SVM	Reviews (Electronics) [13, 37, 164, 202]	-	-	-	0.787	-
Pannala et al. (2016) [151]	SVM	Reviews (Electronics) [158]	0.732	-	-	-	-
		Reviews (Restaurants) [158]	0.823	-	-	-	-
Akhtar et al. (2016) [2]	SVM	Reviews (Electronics) [2]	0.511	-	-	-	-
		Reviews (Mobile Apps) [2]	0.421	-	-	-	-
		Reviews (Holidays) [2]	0.606	-	-	-	-
		Reviews (Movies) [2]	0.916	-	-	-	-
Akhtar et al. (2016) [3]	SVM	Reviews (Various) [3]	0.541	-	-	-	-
Hegde and Seema (2017) [86]	SVM	Reviews (Various) [86]	0.758	0.762	0.755	0.758	-
De França Costa and da Silva (2018) [43]	SVM	Social Media (Finance) [126]	-	-	-	-	MSE = 0.151
Al-Smadi et al. (2019) [6]	SVM	Reviews (Hotels) [157]	0.954	-	-	-	AUC = 0.989
Akhtar et al. (2016) [2]	Tree-Based (DT)	Reviews (Electronics) [2]	0.545	-	-	-	-
		Reviews (Mobile Apps) [2]	0.480	-	-	-	-
		Reviews (Holidays) [2]	0.652	-	-	-	-
		Reviews (Movies) [2]	0.916	-	-	-	-
Hegde and Seema (2017) [86]	Tree-Based (DT)	Reviews (Various) [86]	0.806	0.729	0.792	0.759	-
Al-Smadi et al. (2019) [6]	Tree-Based (DT)	Reviews (Hotels) [157]	0.949	-	-	-	AUC = 0.996
Gupta et al. (2014) [79]	Tree-Based (RF)	Reviews (Electronics) [159]	0.671	-	-	-	-
		Reviews (Restaurants) [159]	0.674	-	-	-	-
Tang et al. (2016) [187]	Deep Learning (RNN)	Social Media (Various) [51]	0.715	-	-	0.695	-
Ruder et al. (2016) [166]	Deep Learning (RNN)	Reviews (Electronics) [157]	0.790	-	-	-	-
		Reviews (Restaurants) [157]	0.807	-	-	-	-
		Reviews (Hotels) [157]	0.829	-	-	-	-
Al-Smadi et al. (2018) [7]	Deep Learning (RNN)	Reviews (Hotels) [157]	0.870	-	-	-	-
Yang et al. (2018) [224]	Deep Learning (RNN)	Social Media (Finance) [126]	-	-	-	-	MSE = 0.080
Dong et al. (2014) [51]	Deep Learning (RecNN)	Social Media (Various) [51]	0.663	-	-	0.659	-
		Reviews (Restaurants) [159] (reimplemented in [145])	0.604	0.368	0.604	0.457	-
Nguyen and Shirai (2015) [145]	Deep Learning (RecNN)	Reviews (Restaurants) [159]	0.639	0.624	0.639	0.622	-
Ruder et al. (2016) [167]	Deep Learning (CNN)	Reviews (Electronics) [157]	0.781	-	-	-	-
		Reviews (Restaurants) [157]	0.765	-	-	-	-
		Reviews (Hotels) [157]	0.827	-	-	-	-
Jangid et al. (2018) [98]	Deep Learning (CNN)	Social Media (Finance) [126]	-	-	-	-	MSE = 0.112
Xue and Li (2018) [221]	Deep Learning (CNN)	Reviews (Electronics) [159]	0.691	-	-	-	-
		Reviews (Restaurants) [159]	0.773	-	-	-	-
Zeng et al. (2019) [227]	Deep Learning (CNN)	Reviews (Restaurants) [159]	0.823	-	-	-	-

the feature vector \mathbf{x} , we start at the root node of the tree and examine the splitting condition. The condition and the corresponding feature in \mathbf{x} determine toward which internal node we move next. We move down the tree until a leaf node is reached, which has a certain sentiment class assigned to it, which determines the prediction \hat{y} . A significant advantage of decision trees is their explainability. The decision rules are typically easy to interpret for humans and can therefore be used to discover knowledge and obtain new insights [139]. The new insights can then also be used to, for example, improve the previously discussed knowledge bases. Although decision tree models have not been particularly popular for ABSC, there are some examples of works successfully implementing these models. In [86], an incremental decision tree model is proposed that outperforms an implementation of the previously discussed SVM model. Similarly,

in [2], a decision tree is shown to outperform several other models, including an SVM, on a variety of datasets. However, SVM models can outperform decision tree models for other problems [6].

The main issue with decision trees is the problem of overfitting, which can be especially problematic for ABSC due to the substantial amounts of features that are often used. A solution to this problem is the use of a random forest model, which is an ensemble of decision trees [22]. A random forest consists of a large number of decision tree models. Each tree receives a limited number of features and a bootstrapped sample of the training data to train with. By randomly sampling data and restricting features, the individual decision trees tend to overfit less on specific features or data. Predictions are achieved via majority voting among the aggregated predictions of all individual decision tree models. In [79], a random forest model is implemented for the SemEval-2014 ABSC tasks, but mixed results are obtained for the different datasets. Similar results are presented in [191]. Examples of other tree-based methods are the gradient boosted trees [65] and extra trees [69] classifiers. In [20], a comparison is made that indicates that these models may provide slight improvements compared to the random forest model.

4.2.3 Deep Learning. Deep learning models [72] have revolutionized many fields of research [178], including sentiment analysis and ABSC [50, 230]. Significant amounts of research have been put into developing deep learning models for various types of data and learning tasks. One of the main disadvantages of deep learning models is the fact that they are highly difficult to interpret. Basic machine learning models, like decision trees and linear SVMs, can provide some useful model interpretations. Yet, although attempts have been made to explain the predictions produced by deep learning models [55, 78], these black-box methods are still regarded as practically not explainable. Furthermore, effectively training deep learning models requires large amounts of computational resources. This is because deep learning models require large amounts of data to train, which have not always been available for ABSC. However, as both deep learning research and the amount of publicly available data grow [157–159], deep learning models become more and more popular due to their great predictive performances. Deep learning models used for ABSC include, but are not limited to, recurrent neural networks, recursive neural networks, and convolutional neural networks.

Recurrent neural network (RNN) models [92] have in recent years become one of, if not the most popular choice for ABSC models. RNN models are powerful tools used for learning sequence-based data [117]. They have achieved remarkable results for many language-based learning tasks, including ABSC, but also a wide variety of other sequence-based tasks. A basic RNN model is presented in Figure 2. In language processing, the main idea behind RNN models is that a sequence of words is sequentially fed through a neural network model. The hidden state produced by the neural network based on one word is used as input for the neural network at the next step, such that information is carried through the sequence. This same concept can also be used to process sequences of images, times series, or other sequences. We consider again a record R containing an aspect a with a label y . However, compared to the previously discussed classification methods, we now consider the case where the numeric features are represented in a matrix $X \in \mathbb{R}^{d_x \times n_x}$, where d_x represents the number of features used, and n_x indicates the number of words considered to be in the context. For any task, given an input matrix X , a general RNN model can be defined as follows:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t), \quad (5)$$

where $\mathbf{h}_t \in \mathbb{R}^{d_h}$ is the hidden state vector at step t , d_h is the pre-defined dimension of the hidden state vectors, and $\mathbf{x}_t \in \mathbb{R}^{d_x}$ is the t th column of X , for $t = 1, \dots, n_x$. In the most basic RNN form, the function $f(\cdot)$ represents a concatenation of the two vectors \mathbf{h}_t and \mathbf{x}_t that are then fed through a basic neural network model consisting of linear transformations and non-linear activation functions. Thus, at each step t , information from the previous words

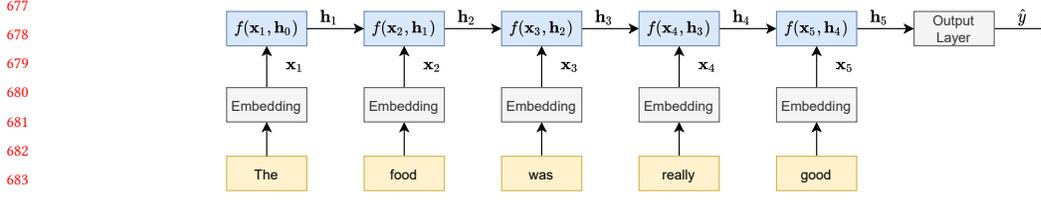


Fig. 2. An illustration of a basic RNN model for ABSC.

(contained in \mathbf{h}_{t-1}) is combined with the information contained in the following word (contained in \mathbf{x}_t). The final hidden state vector \mathbf{h}_{n_x} should then contain all information from the words corresponding to the context of the aspect, processed from left to right. The last hidden state can then be put through an output layer to produce a label prediction. A typical example of an output layer is a linear transformation and a softmax function:

$$\mathbf{s} = \text{softmax}(\mathbf{W}_f \times \mathbf{h}_{n_x} + \mathbf{b}_f), \quad (6)$$

$d_y \times 1$ $d_y \times d_h$ $d_h \times 1$ $d_y \times 1$

where $\mathbf{W}_f \in \mathbb{R}^{d_y \times d_h}$ and $\mathbf{b}_f \in \mathbb{R}^{d_y}$ are, respectively, the trainable weight matrix and bias vector of the final layer, and $\mathbf{s} \in \mathbb{R}^{d_y}$ contains the probabilities of being the correct label for each sentiment class. These probabilities can also be interpreted as sentiment scores. After applying the softmax function, a label prediction can be obtained by predicting the label with the highest sentiment score or probability of being correct. While basic RNN models work well with shorter sequences, problems occur when they are used to process longer and more complex sentences. Due to the multiplicative nature of neural networks, RNN models generally suffer heavily from the vanishing gradient problem, which in turn means that these models have trouble learning long-term dependencies [19, 87]. As such, more advanced RNN models, such as long short-term memory (LSTM) [88] and gated recurrent unit (GRU) [35] models, improve the function $f(\cdot)$ by incorporating a series of gates. These gates allow for the information to flow through the model without losing critical details. These types of RNN models have been proven to work for many different problems and have also become the standard when implementing RNNs for ABSC. Further improvements can be made to RNN models by not only processing the words from left to right but also from right to left. These bi-directional RNN (Bi-RNN) models reverse the flow of information, which allows information at each end of the sequence of words to be preserved.

RNN models are versatile models that can be applied to many tasks that involve sequence data, including ABSC. An implementation of an LSTM model for sentence-level ABSC is presented in [7] where it is compared to an SVM model for ABSC of hotel reviews. The SVM model significantly outperforms the RNN model, which the authors explain to be due to the rich hand-crafted feature vectors used to train the SVM model. In contrast, in [166], a hierarchical LSTM model is implemented and compared to other deep learning architectures using a collection of SemEval-2016 [157] datasets. This hierarchical LSTM model leverages document-level information to perform sentence-level ABSC. The authors show that this model achieves competitive results compared to the best models of the competition, even though no rich hand-crafted features were used. In [187], a different approach is proposed that can be used when processing explicit aspects. Two LSTM models are used to model the left and right parts of the context relative to the target. This approach improves upon the basic LSTM model but is still similar in performance compared to advanced SVM models with rich features and pooled word embeddings [203]. Nevertheless, RNN models seem to generally outperform SVMs, especially for more recent tasks (e.g., the FiQA-2018 task [126]). In recent research, deep learning models like RNNs are quickly outpacing SVM models in terms of predictive performance. This is the case for many language processing tasks,

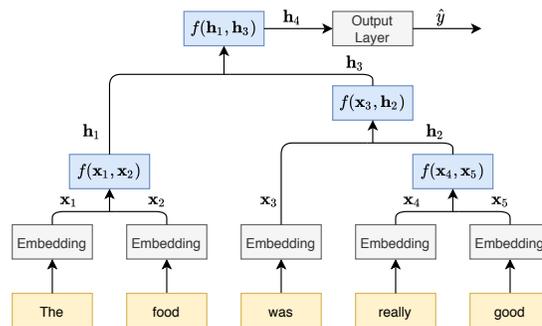


Fig. 3. An illustration of a basic RecNN model for ABSC.

which is in part due to an increase in available training data. See for example the GLUE [206] and SuperGLUE [205] benchmarks, where all the top-performing models are deep learning approaches. Yet, as previously mentioned, the increase in performance comes at a great computational cost, since deep learning models typically require significantly more resources to train than simpler models like SVMs. On the other hand, time is saved on designing handcrafted features. Therefore, when determining the optimal model choice for a certain task, one must take into consideration various factors besides predictive performance on benchmark datasets. Namely, there may not be enough computational resources available, or it could be too costly to obtain enough training data to be able to properly train a deep learning model. On the other hand, there may not be enough time to design handcrafted features for a high-performance SVM.

Recursive neural network (RecNN) models [71] are a generalization of RNN models that process the words using a tree-like structure. Similar to RNNs, RecNNs are general models that can be applied to a variety of tasks. A basic RecNN model is presented in Figure 3. Similarly to RNN models, a function $f(\cdot)$ is defined that is used to combine input vectors. This function is shared throughout the network and is therefore used in each processing step. As seen in Figure 3, RecNN trees consist of leaf nodes, which are the embedded input vectors, and internal nodes (depicted in blue) where vectors are combined using the function $f(\cdot)$. Given a certain internal node with two child nodes, the output of the internal node can be defined as follows:

$$\mathbf{h} = f(\mathbf{c}_1, \mathbf{c}_2), \quad (7)$$

where $\mathbf{h} \in \mathbb{R}^{d_x}$ is the output of the internal node, and $\mathbf{c}_1 \in \mathbb{R}^{d_x}$ and $\mathbf{c}_2 \in \mathbb{R}^{d_x}$ are the outputs of the child nodes. By iteratively combining the word embeddings and internal node output vectors, the model can extract information from the words. After the final step, the output vector \mathbf{h}_4 in Figure 3 should contain the information from the sentence or context. While the example we used considers a binary tree, which is the most common type in ABSC, trees with three or more children per internal node are also possible. Although, the function $f(\cdot)$ would have to be adjusted to process more than two inputs. In the most basic RecNN setup, the function $f(\cdot)$ takes the form of a standard neural network. However, RecNN models using this basic setup may suffer from the same problems as RNNs do when using this function. As such, gated functions like the LSTM module have been adapted for RecNN models [186].

The main advantage of using RecNN models over RNNs is the fact that the model is no longer restricted to processing the words sequentially from left to right and right to left. The order in which the inputs are processed depends on the type of relations that define the tree. The tree may also be structured such that the model processes the words in the original order, meaning that it would be equivalent to a standard RNN. However, herein also lies one of the

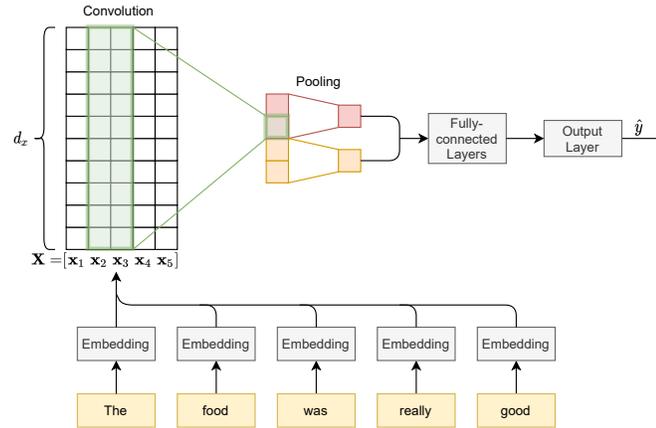


Fig. 4. An illustration of a basic CNN model for ABSC. Each square box represents a value or element of a vector or matrix.

disadvantages of these models. Namely, choosing how to define the tree structure can be difficult. RecNN trees are generally constructed using a language parser, which analyses the structure and dependencies of a sentence and deconstructs the sequence of words. Parsing is an entirely separate field in which much research has been done [111]. Various parsers exist, but the tree in Figure 3 has been generated using the popular Stanford neural network parser [33].

The previously discussed RecNN characteristics are general attributes that apply to any task. Yet, more specialized RecNN models have been proposed for ABSC. For example, trees can be constructed in a manner that conforms specifically to the task of ABSC. In [51], a technique is proposed that can be used to process explicit aspects. The trees in [51] are built towards the target word or phrase corresponding to the explicit aspect. This means that the model learns to forward-propagate the sentiment towards the aspect. Additionally, instead of using only one function $f(\cdot)$, the proposed model implements multiple types of combination functions from which the model adaptively selects based on input vectors and linguistic characteristics. This RecNN specifically designed for ABSC is shown to significantly outperform previous SVM models [101]. Nguyen and Shirai [145] expand on this idea by constructing a model that incorporates both dependency and constituent trees. Furthermore, they also expand upon the use of multiple combination functions. This model is shown to outperform the models presented in [51]. However, based on the reported results, the RecNN models we found still do not perform as well as other deep learning models.

Convolutional neural network (CNN) models [116] are yet another popular deep learning model for text analysis methods, such as sentiment analysis and ABSA [179, 221]. Originally, CNN models were used for processing images using three distinct layer types: convolutional layers, pooling layers, and fully-connected layers. A basic CNN model illustrating these layers is presented in Figure 4 in the context of language processing. The convolutional layer is generally described as a filter that slides across the input data matrix and produces linear combinations of the values within the window. Pooling layers can be used to reduce the dimensionality of the output and improve generalization by summarizing the information obtained from the convolutional layer. A global pooling layer is typically also used before the final layers to transform the features matrix into a single vector. This vector is then processed using fully-connected layers to produce the correct output.

The previously explained model layers form a general model architecture that can be used for many tasks. As such, CNN models have also been used for ABSC. In [142], CNN models are shown to produce superior performances

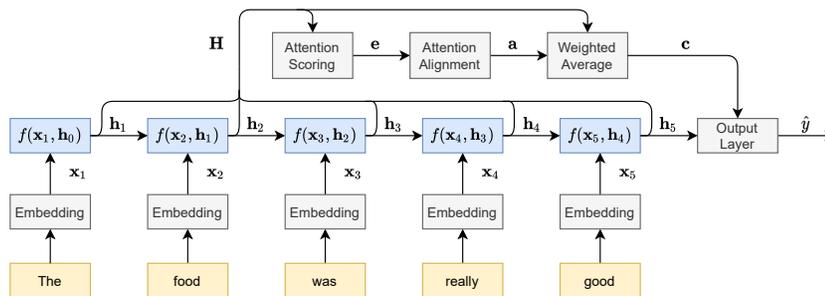


Fig. 5. An illustration of a basic attention-based RNN model for ABSC.

compared to SVM models for ABSC. In [167], a CNN specifically designed for ABSC is implemented by including target embeddings so that the aspect can be modeled explicitly. The aspect embedding is concatenated with the input vectors to be used as input for the CNN. The reported results indicate that this approach allows for performances on par with the top models for the used datasets. A different approach presented in [221] also uses aspect embeddings but in the form of a non-linear gating mechanism inserted between the convolutional and pooling layers. This model produces highly accurate predictions, outperforming many other models, including a random forest [79], RecNN models [51, 145], and even several attention-based deep learning models which are discussed in Subsection 4.2.4. In [227], this gated CNN is further expanded and improved using a linguistic regularization expansion of the loss function. Although this model comes close to being the best performing approach for the SemEval-2014 dataset, it was still outperformed by a hybrid model [106], which is discussed in Subsection 4.3.1.

4.2.4 Attention-Based Deep Learning. The attention mechanism [15] is a highly effective extension to the deep learning models discussed in the previous subsection. In this section, we discuss attention-based deep learning (AB-DL) models for ABSC. The performances of various AB-DL models are displayed in Table 5. We illustrate the attention mechanism using Figure 5 in which the example RNN model presented in Figure 2 is extended with a basic attention architecture. Consider the input representation matrix $X \in \mathbb{R}^{d_x \times n_x}$ for which the columns are the vectors $x_1, \dots, x_{n_x} \in \mathbb{R}^{d_x}$. RNN units are used to encode the sequence of words to produce the hidden states $h_1, \dots, h_{n_x} \in \mathbb{R}^{d_h}$ that correspond to the n_x words. For clarity, we concatenate the hidden states to create the hidden state matrix $H \in \mathbb{R}^{d_h \times n_x}$. Due to the sequential nature of the processing of the input vectors, the last hidden state h_{n_x} (h_5 in Figure 5) should contain information from the entire sequence. However, as previously mentioned, RNN models struggle to learn long-term dependencies, which can be addressed using the attention mechanism.

The basic attention architecture consists of three operations: attention scoring, attention alignment, and a weighted averaging operation. In the most general form, the attention mechanism requires three inputs: **key** vectors, **value** vectors, and a **query** vector. These attention concepts and the corresponding notation were introduced in [42] and further popularized in [201]. The keys and values are generally derived from the data matrix on which attention is calculated. In the example presented in Figure 5, the data matrix upon which we calculate the attention is the matrix of hidden vectors $H = [h_1, \dots, h_{n_x}]$. Key vectors can be derived by linearly transforming the hidden state vectors using a trainable weights matrix $W_K \in \mathbb{R}^{d_k \times d_h}$, which produces the key vectors $k_1, \dots, k_{n_x} \in \mathbb{R}^{d_k}$. A similar process can be performed to obtain the value vectors $v_1, \dots, v_{n_x} \in \mathbb{R}^{d_v}$ using the trainable weights matrix $W_V \in \mathbb{R}^{d_v \times d_h}$. However, for simplicity's sake, the example in Figure 5 simply uses the original hidden state vectors as both the

Table 5. Overview of prominent attention-based deep learning ABSC models and their reported performances.

Reference	Attention Characteristics	Domain(s)	Accuracy	Precision	Recall	F ₁
Wang et al. (2016) [210]	Additive + Global + LSTM	Reviews (Electronics) [159] Reviews (Restaurants) [159]	0.687 0.772	- -	- -	- -
Yang et al. (2017) [223]	General + Global + LSTM	Social Media (Various) [51]	0.726	-	-	0.722
Liu et al. (2018) [120]	Additive + Global + GRU	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51] Reviews (Restaurants) [157, 158] (reimplemented in [204])	0.751 0.809 0.715 0.806	- - - -	- - - -	- - - -
Zhang et al. (2019) [228]	Additive + Global + CNN	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51]	0.754 0.795 0.713	- - -	- - -	- - -
Gan et al. (2020) [66]	Additive + Global + CNN	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51]	0.736 0.814 0.742	- - -	- - -	- - 0.732
He et al. (2018) [84]	Activated General + Syntax-Based + LSTM	Reviews (Electronics) [159] Reviews (Restaurants) [157-159]	0.719 0.823	- -	- -	0.692 0.683
Ma et al. (2017) [123]	Activated General + Global + Co-attention + LSTM	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51] (reimplemented in [222])	0.721 0.786 0.698	- - -	- - -	- - -
Gu et al. (2018) [75]	Activated General + Global + Co-attention + Bi-GRU	Reviews (Electronics) [159] Reviews (Restaurants) [159]	0.741 0.812	- -	- -	- -
Fan et al. (2018) [63]	General + Global + Co-attention + Bi-LSTM	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51]	0.754 0.813 0.725	- - -	- - -	0.725 0.719 0.708
Yang et al. (2019) [222]	Additive + Global + Co-attention + LSTM	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51]	0.735 0.788 0.715	- - -	- - -	- - -
Zheng and Xia (2018) [229]	Activated General + Global + Rotatory + Bi-LSTM	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51] Reviews (Restaurants) [157, 158] (reimplemented in [204])	0.752 0.813 0.727 0.827	- - - -	- - - -	- - - -
Tang et al. (2016) [188]	Additive + Global + Multi-hop + LSTM	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51] (reimplemented in [222])	0.722 0.810 0.685	- - -	- - -	- - -
Fan et al. (2018) [62]	Additive + Multi-Hop + Position-Based + Bi-GRU	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51]	0.764 0.783 0.721	- - -	- - -	0.721 0.684 0.708
Majumder et al. (2018) [127]	Additive/Multiplicative + Global + Multi-Hop + GRU	Reviews (Electronics) [159] Reviews (Restaurants) [159]	0.738 0.800	- -	- -	- -
Wallaart and FrasinCAR (2019) [204]	Activated General + Global + Rotatory + Multi-Hop + Bi-LSTM	Reviews (Restaurants) [157, 158]	0.831	-	-	-
Gao et al. (2019) [67]	Transformer	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51]	0.784 0.846 0.773	- - -	- - -	0.744 0.796 0.744
Xu et al. (2019) [218]	Transformer	Reviews (Electronics) [159] Reviews (Restaurants) [159]	0.781 0.850	- -	- -	0.751 0.770
Zeng et al. (2019) [226]	Transformer	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51]	0.825 0.871 0.773	- - -	- - -	0.796 0.817 0.758
Sun et al. (2019) [34]	Transformer	Social Media (Neighborhoods) [168]	0.933	-	-	-
Karimi et al. (2020) [105]	Transformer	Reviews (Electronics) [159] Reviews (Restaurants) [159]	0.794 0.860	- -	- -	0.765 0.792
Xu et al. (2020) [220]	Transformer + Co-attention	Reviews (Electronics) [159] Reviews (Restaurants) [157-159] Social Media (Various) [51]	0.781 0.843 0.766	- - -	- - -	0.732 0.712 0.722
Ansar et al. (2021) [12]	Transformer	Reviews (Movies) [192, 194] Social Media (Neighborhoods) [168]	0.952 0.924	- -	- -	- -
Su et al. (2021) [183]	Transformer	Reviews (Electronics) [159] Reviews (Restaurants) [159] Social Media (Various) [51]	0.826 0.879 0.776	- - -	- - -	0.793 0.823 0.765

keys and values. The keys are used in combination with the query vector $\mathbf{q} \in \mathbb{R}^{d_q}$ in the first step of the attention calculation: attention scoring, which involves calculating an attention score corresponding to each feature vector. This score directly determines how much attention is focused on each word. The attention score for the t th word depends on the corresponding key vector \mathbf{k}_t , the query vector \mathbf{q} , and a score function. The score function can take a variety of forms, but it is generally meant to determine a relation between the key and query vectors. One of the most common score functions is the **additive** score function [15], which is used in many ABSC models [210]:

$$\mathbf{e}_t = \mathbf{w}^T \times \text{act}(\mathbf{W}_k \times \mathbf{k}_t + \mathbf{W}_q \times \mathbf{q} + \mathbf{b}), \quad (8)$$

where $\mathbf{e}_t \in \mathbb{R}^1$ is the attention score belonging to the t th word, $\mathbf{w} \in \mathbb{R}^{d_w}$, $\mathbf{W}_k \in \mathbb{R}^{d_w \times d_k}$, $\mathbf{W}_q \in \mathbb{R}^{d_w \times d_q}$, and $\mathbf{b} \in \mathbb{R}^{d_w}$ are trainable weight matrices and vectors, and d_w is a predefined dimension parameter. Other types of score functions are often based on a product, such as the **multiplicative** [122], **scaled-multiplicative** [201], **general** score function [122], and **activated general** [123] score functions. One can interpret attention scoring as determining which words contain the most important information with regards to determining the correct sentiment classification. The higher the score, the more important the information contained in the corresponding vector. Based on the scoring calculation, the query \mathbf{q} is essential in determining what information is important. The simplest method of defining this query is by setting it as a constant vector. This allows the model to learn a general way of defining which words to focus on. However, since aspects can differ widely in their characteristics, a general query may not be flexible enough. A more popular technique is to use a vector representation of the aspect as the query. As seen in [223], one can take the hidden vector representation of the target (a pooled version of \mathbf{h}_1 and \mathbf{h}_2 in Figure 5) as the query. In this case, the attention score calculation determines which information is most important with regard to the aspect itself, which enhances the model's ability to focus on the most important words in the sequence. This is especially important when a sentence contains multiple aspects. The purpose of the attention scores is to be used as weights in a weighted average calculation. However, the weights are required to add up to one for this purpose. As such, an alignment function $\text{align}()$ is used on the attention scores $e_1, \dots, e_{n_x} \in \mathbb{R}^1$:

$$\mathbf{a}_t = \text{align}(e_t; \mathbf{e}), \quad (9)$$

where $\mathbf{a}_t \in \mathbb{R}^1$ is the attention weight corresponding to the t th word, and we define the vector $\mathbf{e} \in \mathbb{R}^{n_x}$ as the vector containing all attention scores. A highly popular alignment function is known as **soft** or **global** alignment which applies a softmax function to the attention scores. Other types of alignment are, for example, the **hard** [219] and **local** [122] alignment functions, which provide a more focused attention alignment. Additionally, in [84], the **syntax-based** alignment function is introduced that is specifically designed for ABSC. It employs global alignment but scales the attention weights according to how far the corresponding words are from the target in a dependency tree. The attention weights $a_1, \dots, a_{n_x} \in \mathbb{R}^1$ are used in combination with the value vectors $\mathbf{v}_1, \dots, \mathbf{v}_{n_x} \in \mathbb{R}^{d_v}$ to calculate a so-called **context** vector $\mathbf{c} \in \mathbb{R}^{d_v}$. For clarity, in Figure 5 the attention weights are represented by the vector of attention weights $\mathbf{a} \in \mathbb{R}^{n_x}$. The context vector can be used in combination with the final hidden state in the output layer to obtain a label prediction. The context vector can be calculated as follows:

$$\mathbf{c} = \sum_{t=1}^{n_x} \mathbf{a}_t \times \mathbf{v}_t. \quad (10)$$

989 This context vector summarizes the most important information contained in the context corresponding to the aspect.
990 Since the attention mechanism can pick and choose information from any of the hidden states, information can be
991 retrieved independently of where the it is positioned in the sequence. This significantly improves the model’s ability to
992 capture long-term dependencies. In addition, the attention weights are a direct indication of the importance of certain
993 words. As such, attention models offer some explanation by analysing which words the model tends to focus on. This
994 provides some interpretability for models which are typically considered black-box models. Although, the usefulness of
995 the interpretation of attention weights is controversial [97, 138, 214].
996
997

998 The example we used incorporates the hidden states from an RNN model into the attention calculation. However,
999 attention can also be applied in other models such as a CNN. In the case of a CNN, the sets of feature maps that are
1000 obtained after the convolutional layer can be used as input for the attention mechanism. The rest of the attention
1001 calculation remains the same. The score and alignment functions are basic aspects that are required for every attention
1002 model. While the basic attention model we discussed can be used for the task of ABSC, extensions are often implemented
1003 for further improvements or to process other types of inputs. For example, **co-attention** [121] can be used to calculate
1004 attention between multiple features matrices if there are multiple model inputs. This can be the case for ABSC if the
1005 input text is split up into parts, such as the target words and the words left and right of the target. A similar approach is
1006 called **rotatory** [229], which rotates attention between the inputs. A more general attention extension is **multi-head**
1007 attention [201], which employs so-called attention ‘heads’ to calculate multiple different types of attention in parallel. In
1008 contrast, **multi-hop** attention [196] is an extension that allows for multiple attention calculates in sequence. Multi-head
1009 and multi-hop attention are vital parts of the transformer model [201]. This model is based purely on the attention
1010 mechanism and does not use a separate base model like an RNN or CNN. This model uses **self-attention** to calculate
1011 attention between the feature vectors and use the extracted information to iteratively transform the input. Typical of
1012 this model type is the use of a scaled-multiplicative score function and a global alignment function since these functions
1013 are computationally highly efficient. The transformer model has been proven to be highly effective for many tasks,
1014 including ABSC. Gao et al. [67] present a transformer model with ABSC extensions for properly processing the target.
1015 A transformer model incorporating the previously mentioned co-attention mechanism is proposed in [220]. Zeng et al.
1016 [226] implement two transformer models in parallel to allow the model to process the global context and local context
1017 concerning the aspect separately. All of these transformer models and extensions achieve significant improvements
1018 over previous models, including other attention models. When only considering predictive performance, transformer
1019 models tend to dominate most ABSC tasks and other modern language processing benchmarks, such as GLUE [206]
1020 and SuperGLUE [205]. Yet, these models require large amounts of training data and computational resources to train
1021 from scratch, even compared to other deep learning models. A portion of current research is focused on alleviating
1022 the resources necessary for training transformer models via more efficient architectures, such as the linformer [209].
1023 Nevertheless, transformer models can produce impressive results, but may not be suitable for problems where the
1024 required resources are not available.
1025
1026
1027
1028
1029
1030
1031

1032 4.3 Hybrid

1033 When limited data is available, machine learning models may not provide satisfactory results [179]. ABSC is a domain
1034 where datasets are typically considered to be small in size [85] (see Table 2), which makes training large language
1035 models a significant challenge. This can be an even worse problem when attempting ABSC in some of its more niche
1036 sub-domains, such as sentiment analysis of texts for less popular products or topics. This problem is compounded by
1037 the fact that there is a vast number of different languages. English is a language for which there typically are extensive
1038
1039
1040

resources available. Yet, for other languages, like Arabic or Chinese, new datasets are slowly being introduced (see SemEval-2016 [157]) but the amount of data is still limited [50]. As such, large language models for ABSC can be hard to train. A solution to this problem is to incorporate additional knowledge from knowledge bases to compensate for the lack of data. We refer to models that incorporate both machine learning and knowledge bases as hybrid models. There are various ways in which knowledge bases can be combined with machine learning. A common approach is to use a knowledge base to define features that the machine learning algorithm uses to predict the sentiment. Another is to implement both a knowledge-based classifier and a machine learning classifier in sequence or parallel. In this section, we discuss these various approaches and categorize them using three categories: dictionary-enhanced, ontology-enhanced, and discourse-enhanced models. The performances of different hybrid models are presented in Table 6.

4.3.1 Dictionary-Enhanced Machine Learning. Dictionary-enhanced machine learning methods are some of the more common hybrid approaches. Dictionaries are rich knowledge bases that can easily be used to define features for a machine learning algorithm. In [200], sentiment scores from the *SentiWordNet* [14] dictionary are included as features to improve the performance of an SVM model. In both [47] and [110], the same technique using *SentiWordNet* is also used. Additionally, *NRC-Canada* [106], the top-performing model of the SemEval-2014 ABSC task, is based on a similar concept. However, the authors generate their own sentiment lexicons from unlabeled review corpora. In [203], a different approach is presented. First, word embeddings are defined using *word2vec* [133]. Then, embeddings for words that are not in the sentiment lexicons are filtered out. As such, only words that contain useful sentiment information are kept. Since an SVM model is used for classification, the authors employ several pooling functions to summarize the features from the dense feature vectors to be used by the SVM model.

In [16], it is argued that attention models typically overfit when trained using small datasets. As such, [16] proposes an attention-based LSTM model that incorporates lexicon features to improve the flexibility and robustness of attention-based deep learning models when trained with insufficient data. In [4], a hybrid model is presented specifically for resource-poor languages, such as Hindi. The proposed model combines trained CNN features and lexicon features. It is shown to produce better performances than the tested baselines that do not utilize the knowledge bases. In [96], a Bi-GRU model is proposed for ABSC for reviews from an Indonesian online marketplace. It is shown that performance is improved by incorporating lexicon features into the model.

Lexicons and other knowledge bases can enhance machine learning models, but machine learning can also improve the production of knowledge bases. An important example of this is *SenticNet* [30], which is a knowledge base that uses linguistic patterns, first-order logic, and deep learning to discover relationships between entities, concepts, and primitives. Such ensembles of symbolic and subsymbolic tools can be highly useful for sentiment analysis and ABSA. For example, [125] proposed the Sentic LSTM model, which is an attention-based LSTM that fully incorporates sentic knowledge into the deep learning architecture. It is shown that this ensemble application of symbolic and subsymbolic AI produces better results than symbolic and subsymbolic AI separately.

4.3.2 Ontology-Enhanced Machine Learning. Ontologies can be used both as a knowledge base for sentiment information and to define a structure between concepts. In [45], a restaurant-specific ontology is used to define features for a review-level ABSC SVM model. The authors use both the concepts and the sentiment information from the ontology to define features. Essentially, a “bag-of-concepts” is defined that includes a binary feature for each concept in the ontology. However, the value of a concept feature depends on whether it is related to the aspect itself, meaning that implicit information in the text can be encoded into the feature vector. Secondly, features are defined to count how often sentiment polarity words from the ontology occur in the text.

Table 6. Overview of prominent hybrid ABSC models and their reported performances. “AB-DL = Attention-Based Deep Learning”

Reference	Category	Domain(s)	Accuracy	Precision	Recall	F ₁
Varghese and Jayasree (2013) [200]	Dictionary-Enhanced SVM	Reviews (Movies) [60] Reviews (Music) [155]	0.853 0.865	- -	- -	- -
Kiritchenko et al. (2014) [106]	Dictionary-Enhanced SVM	Reviews (Electronics) [159] Reviews (Restaurants) [159]	0.705 0.802	- -	- -	- -
Vo et al. (2015) [203]	Dictionary-Enhanced SVM	Social Media (Various) [51]	0.711	-	-	0.699
Devi et al. (2016) [47]	Dictionary-Enhanced SVM	Reviews (Electronics) [10, 57, 64]	0.881	0.872	0.898	0.884
Fachrina and Widyantoro (2017) [61]	Dictionary-Enhanced SVM	Reviews (Various) [61]	-	-	-	0.858
Akhtar et al. (2016) [4]	Dictionary-Enhanced CNN	Reviews (Various) [3] Reviews (Electronics) [159] Reviews (Restaurants) [159]	0.660 0.680 0.772	- - -	- - -	- - -
Ma et al. (2018) [125]	Dictionary-Enhanced RNN	Reviews (Restaurants) [158] Social Media (Neighborhoods) [168]	0.765 0.893	- -	- -	- -
Ilmania et al. (2018) [96]	Dictionary-Enhanced RNN	Reviews (Various) [61]	-	-	-	0.848
Bao et al. (2019) [16]	Dictionary-Enhanced AB-DL	Reviews (Restaurants) [159]	0.829	-	-	-
Schouten et al. (2017) [176]	Ontology-Enhanced SVM	Reviews (Restaurants) [157]	-	-	-	0.753
De Heij et al. (2017) [44]	Ontology-Enhanced SVM	Reviews (Restaurants) [158]	-	-	-	0.808
De Kok et al. (2018) [45]	Ontology-Enhanced SVM	Reviews (Restaurants) [157]	-	-	-	0.812
Schouten and Frasinca (2018) [175]	Ontology-Enhanced SVM	Reviews (Restaurants) [157, 158]	0.842	-	-	-
García-Díaz et al. (2020) [68]	Ontology-Enhanced RNN	Social Media (Diseases) [68]	0.542	-	-	-
Kumar et al. (2020) [114]	Ontology-Enhanced CNN	Reviews (Hotels) [21]	0.885	0.943	0.856	0.860
Meškelė and Frasinca (2019) [130]	Ontology-Enhanced AB-DL	Reviews (Restaurants) [157] Reviews (Restaurants) [157, 158] (reimplemented in [131])	0.863 0.834	- -	- -	- -
Wallaart and Frasinca (2019) [204]	Ontology-Enhanced AB-DL	Reviews (Restaurants) [157, 158]	0.843	-	-	-
Trusca et al. (2020) [197]	Ontology-Enhanced AB-DL	Reviews (Restaurants) [157, 158]	0.844	-	-	-
Meškelė and Frasinca (2020) [131]	Ontology-Enhanced AB-DL	Reviews (Restaurants) [157, 158]	0.855	-	-	-
Wang et al. (2018) [208]	Discourse-Enhanced AB-DL	Reviews (Electronics) [158] Reviews (Restaurants) [158]	0.816 0.809	- -	- -	0.667 0.685
Wu et al. (2019) [216]	Discourse-Enhanced AB-DL	Reviews (Electronics) [159] Reviews (Restaurants) [157–159] Social Media (Various) [51]	0.734 0.828 0.698	- - -	- - -	0.691 0.642 0.675

A second approach for ontology-enhanced models is via a two-step method that sequentially employs a knowledge-based method and a machine learning classifier. First, an ontology-based model is employed to attempt to classify the aspect. When a conflicting answer is found for the sentiment label or when there is no information available regarding the sentiment, a backup model is employed in the form of a machine learning classifier. In [204], sentence-level ABSC is performed by implementing a multi-hop rotatory attention model as the backup algorithm after an ontology-based classification model. The results presented indicate that the two-step method performs better compared to either of the classifiers individually. In [130], this two-step approach is used with a lexicalised domain ontology followed by an attention model. The attention model combines a sentence-level content attention mechanism and a bidirectional context attention mechanism. In [131], this model is further extended using *BERT* [48] word embeddings, additional regularization, and adjustments to the training process. In [197], the two-step method using an ontology and a multi-hop rotatory attention model is extended with a hierarchical attention mechanism and deep contextual word embeddings. In [175], both the two-step method and ontology-based features are used for sentence-level ABSC. First, an ontology-based

1145 classifier is used to try to determine a sentiment classification for the aspect. If no definitive answer can be found, an
1146 SVM is employed that uses a bag-of-words feature vector enhanced with ontology sentiment features.

1147 Ontology-enhanced methods can prove useful when training language models for ABSC in niche domains. In [44],
1148 an ontology-enhanced SVM model is tested on the SemEval-2016 dataset. It is shown that a model with ontology
1149 features attains a significantly higher F_1 -score than models that do not have those features. Moreover, the model with
1150 ontology features can obtain equal performance with less than 60% of the training data. In [176], it is shown that
1151 an ontology-enhanced model can be highly robust to changes in dataset size. For the task of aspect extraction, the
1152 performance of the ontology-enhanced SVM model proposed in [176] drops by less than 10% with only 20% of the
1153 original training data, while the performances of the base methods drop significantly. For the task of ABSC, all models
1154 proposed in [176] appear to be robust to dataset size changes. Yet, this can be explained by the fact that all methods
1155 are also enhanced using another knowledge base in the form of sentiment dictionaries. In [68], an ontology-enhanced
1156 ABSC method is employed in the niche domain of infodemiology in the Spanish language. Aspects are extracted from
1157 Twitter posts and are classified using a Bi-LSTM model and an ontology of the infectious disease domain.
1158
1159

1160
1161
1162 **4.3.3 Discourse-Enhanced Machine Learning.** As discussed previously, discourse trees function similarly to ontologies
1163 in the sense that they do not inherently contain sentiment information. However, similarly to ontologies, the structure
1164 of discourse trees can still be useful to combine with a machine learning classifier. Wang et al. [208] use the structure of
1165 a discourse tree to determine how words are processed via an attention-based deep learning model for sentence-level
1166 ABSC. For each clause, a Bi-LSTM and an attention layer are used to produce a clause vector represented by the
1167 context vector output of the attention layer. The clause representations are then processed through another Bi-LSTM
1168 and attention layer to create a hierarchical attention structure. The resulting context vector is used for prediction.
1169 While the individual clauses defined by the discourse tree are incorporated in this model, the relational structure
1170 of the tree is not. In [216], the relations between discourse clauses are incorporated through the use of conjunction
1171 rules. First, each clause identified by the discourse tree is processed separately using Bi-LSTM layers. However, clause
1172 representations are produced simply by averaging. These clause representations are then used in the output layer, a
1173 layer to extract the relations between clauses using a Bi-LSTM, and a layer where conjunction rules are used to extract
1174 additional features. Conjunctions indicate how clauses are connected. Clauses connected by words such as “and” are
1175 called coordinate conjunctions and often indicate a shared sentiment between the clauses. When words such as “but”
1176 are used, an adversative conjunction is present that generally indicates an opposite sentiment. This information is used
1177 to selectively summarize the clause representations. Using these techniques, the rhetorical structure of the context can
1178 be incorporated into the model.
1179
1180
1181
1182
1183

1184 5 RELATED TOPICS

1185 5.1 Aspect Detection & Aggregation

1186 In Section 1, ABSA was explained to consist of three steps: detection, classification, and aggregation. This survey
1187 discusses the classification step, also known as ABSC. Yet, the three steps cannot be completely separated from each
1188 other since these issues are not independent [174]. For example, it may be important to consider the design of the
1189 classification and aggregation steps jointly, since information extracted during the classification step can be useful
1190 during the aggregation step. Namely, one can use features such as sentence importance to create a weighted average,
1191 which typically outperforms simple averages for aggregation [18]. Similarly, information from the aspect detection
1192 step can be useful in the classification step. For example, one can use features extracted during the aspect extraction
1193
1194
1195
1196

1197 phase to predict the sentiments [25]. Yet, models have also been proposed that fully jointly perform the detection and
1198 classification steps. For example, in [234] a dictionary-based model using *WordNet* [134] is proposed that extracts and
1199 pairs aspects and opinion words to find aspect-level sentiments. In [172], the detection and classification steps are
1200 completely fused in a single end-to-end LSTM model. The proposed model jointly learns the two tasks and produces
1201 improved performance for both tasks. In [211], a capsule attention model is proposed that also jointly learns the
1202 detection and classification tasks in an end-to-end manner. This model produces state-of-the-art results for both tasks.
1203
1204

1205 5.2 Sarcasm & Thwarting

1206

1207 Thwarting and sarcasm are two complex linguistic phenomena that pose significant challenges for sentiment analysis
1208 models. Thwarting is the concept of building expectations, only to then contrast them. In other words, the overall
1209 sentiment of a document differs from the sentiment expressed throughout the majority of the text [161]. As such,
1210 methods that simply rely on aggregating the sentiment expressed by individual words can easily miss the greater
1211 context of the thwarted text. Additionally, developing a model for detecting thwarting is a difficult task due to the
1212 lack of available training data [161]. In the limit, detecting thwarting can also be considered to be highly similar to the
1213 difficult task of recognizing sarcasm [161]. Sarcasm is the act of deliberately ridiculing or questioning subjects by using
1214 language that is counter to its meaning [104]. It is particularly common in social media texts, where sarcasm analysis
1215 often also involves emoticons and hashtags [129]. Recognizing sarcasm in a text is a task that is often even difficult for
1216 humans [129] and requires significant world knowledge, which is difficult to include in most sentiment models [161].
1217
1218

1219 The two problems of thwarting and sarcasm make sentiment analysis non-trivial [11]. Yet, they have received little
1220 attention in the literature on ABSC. For ABSC, thwarting or sarcasm means that the sentiment expressed towards the
1221 aspect via the language used throughout the majority or the entirety of the record is opposite from the true sentiment.
1222 In the literature on general sentiment analysis, some attempts to address these problems have been proposed. Although,
1223 sentiment analysis is often considered a separate task from thwarting or sarcasm detection. For example, in [161], a
1224 domain-ontology of camera reviews is used in combination with an SVM model to identify thwarted reviews of products.
1225 In [112], a multi-head attention-based LSTM model is used to detect sarcasm in social media texts. Yet, thwarting and
1226 sarcasm can also be handled in the context of sentiment analysis. For example, in [135], the problems of thwarting and
1227 sarcasm in sentiment analysis are addressed using cognitive features obtained by tracking the eye movements of human
1228 annotators. It is theorized that the cognitive processes used to recognize thwarting and sarcasm are related to the eye
1229 movements of readers. It is shown that the gaze features significantly improve the performance of sentiment analysis
1230 classifiers, such as an SVM. Furthermore, it is specifically shown that the gaze features help address the problem of
1231 analysing complex linguistic structures, such as thwarting and sarcasm. In [59], the tasks of sentiment analysis and
1232 sarcasm detection are considered jointly via a multi-task model. A transformer model is used to identify both sentiment
1233 and sarcasm in Arabic tweets. It is shown that the proposed multi-task model outperforms its single-task counterparts.
1234 Such methods could also be used for ABSC. Cognitive features could be combined with aspect features to address
1235 thwarting and sarcasm for the task of ABSC. Similarly, the task of ABSC can be considered jointly with the task of
1236 sarcasm detection. Yet, more specialized ABSC datasets would be required for such problems.
1237
1238
1239
1240
1241

1242 5.3 Emotions

1243

1244 Emotions are an interesting subject in conjunction with ABSC. Sentiment analysis and emotion analysis are two highly
1245 related topics. In sentiment analysis, we typically assign polarity labels or scores to texts, but in emotion analysis, a
1246 wide range of emotions (e.g., “joy”, “sadness”, “anger”) are to be considered [28]. For example, in [195], movie reviews
1247
1248

are analysed and assigned emotion scores based on the dimensions of the hourglass of the emotions [31]. Like sentiment analysis, emotion analysis can be done at multiple levels [81]. The equivalent of ABSC for emotion analysis is called aspect-based emotion analysis. For example, in [156], emotions towards aspects in social media posts are classified using word embedding centroids, lexicons, and emoticons. More extensive work on the usage of emoticons in sentiment analysis is available from [89, 90]. In [184], several models are tested to detect emotions towards aspects in restaurant reviews. Information about emotions can also help with the task of ABSC. For example, in [113], features from the *SenticNet* [30] knowledge base are used to enhance a model for ABSC. In [125], an attention-based LSTM model is proposed that incorporates emotion information from the *AffectiveSpace* knowledge base [29] for improved performance in ABSC tasks. A similar technique is used in [124]. Additionally, emotion analysis and sentiment analysis can also be performed jointly. For example, in [213], a model is proposed that can extract aspect-level affective knowledge that includes sentiment polarities and emotion categories. In [212], an extensive framework for multi-level sensing of sentiments and emotions is proposed that incorporates knowledge bases and sarcasm handling.

6 CONCLUSION

In this survey, we have provided an overview of the current state-of-the-art models for ABSC. We have explained the process of ABSC according to its three main phases: input representation, aspect sentiment classification, and output evaluation. The input representation phase involves representing a body of text by a numeric vector or matrix such that a classification model can identify the correct polarity label of an aspect. In the output evaluation phase, the quality of these polarity label predictions is assessed using performance measures. The quality of the predictions is determined by the architecture of the classification model. We have discussed a variety of state-of-the-art ABSC models using a proposed taxonomy and summarizing tables that serve as overviews of model performances. These ABSC models have been discussed and compared using intuitive explanations, technical details, and reported performances. We have also discussed a variety of important topics related to ABSC.

ABSC is a relatively new task that has quickly gained popularity and is rapidly changing. A noticeable evolution in the field of ABSC concerns the used datasets. The authors of the early ABSC works often scraped and compiled their own datasets from the Web. While this phenomenon makes sense due to the vast amounts of public reviews available online, it makes performance comparisons difficult due to most models being tested on different datasets. It was only after researchers started adopting the Twitter dataset from [51] and the review datasets from the SemEval challenge [157–159] that actual comparative analyses became more feasible. While this is a significant development for the domain of ABSC, a consequence is that ABSC models are mostly only implemented for restaurant reviews, electronics reviews, and Twitter data. Other types of data, such as hotel reviews, go mostly ignored even though the SemEval-2016 datasets [157] contain a set of hotel reviews. To further advance the field of ABSC, it is desirable to test models in other domains. This requires more high-quality public datasets to be made available and adopted. Examples are the FiQA-2018 [126] and SentiHood [168] datasets. Another example is the book review dataset produced by Álvarez-López et al. [9]. This dataset contains a subset of book reviews taken from the INEX Amazon/LibraryThing book corpus [109] that was hand-annotated at the aspect level. Yet, this dataset has rarely been adopted by other researchers.

New datasets are also required for languages other than English. As discussed, English is a language for which there is a relatively sizeable number of datasets available for the task of ABSC. This is not the case for most other languages, which makes training language models a difficult task. If it is too costly to obtain more training data, simpler models like SVMs are typically the better option. However, the incorporation of knowledge bases may help compensate for the lack of data, as evidenced by the hybrid models discussed in Subsection 4.3. As has been shown in

1301 various works [130, 197, 208], knowledge bases are effective in enhancing state-of-the-art models to achieve higher
1302 performances. We believe that the further exploration of knowledge-enhanced methods will help improve on the
1303 current state-of-the-art. However, the incorporation in current research is mostly relatively basic. For example, in [208],
1304 only the clauses extracted by the discourse tree are incorporated in the model. The model presented in [216] improves
1305 upon this by including conjunction rules, but this solution fails to exploit the many other discourse relations that can
1306 contain useful information for sentiment classification. Similarly, we have mainly seen domain ontologies in hybrid
1307 models as part of a two-step method where the ontology is separate from the machine learning model. Yet, ontologies
1308 contain many concepts and relations that can provide important features for the machine learning model. As such,
1309 we would advocate for further incorporation of knowledge bases and their structures. Furthermore, while we have
1310 only discussed three types of knowledge bases (dictionaries, ontologies, and discourse trees), we expect to see further
1311 exploration of new knowledge bases to be incorporated for ABSC. Additionally, improvements can be made in the
1312 construction of knowledge bases. This is exemplified by the recent research on the semi-automatic construction of
1313 domain ontologies for ABSC [46, 175, 190, 235]. Another problem is that proper knowledge bases for resource-poor
1314 languages may be scarce as well [3]. As such, the development of new knowledge resources other than labeled ABSC
1315 datasets is an important next step.

1320 Often, the problem is not a lack of data, but specifically a lack of labeled data. Unsupervised or weakly supervised
1321 methods can be highly useful in situations where labeled data is too expensive to obtain. For example, in [233], a weakly
1322 supervised model is proposed for joint aspect extraction and sentiment classification. The proposed model involves a
1323 sentiment dictionary learned via an auto-encoder using attention. Furthermore, weakly supervised systems can even be
1324 used to produce new training data via labeling mechanisms based on expert knowledge [163]. An alternative solution
1325 to the problem of a lack of data is the use of cross-lingual and multi-lingual models. Knowledge from a language with
1326 extensive resources available can be transferred to models for other languages to compensate for the lack of data [17].
1327 For example, in [5], an attempt is made to solve the problem of data sparsity in French and Hindi datasets via the use of
1328 a deep learning model built on top of bilingual word embeddings. These bilingual word embeddings were produced
1329 using English-French and English-Hindi parallel corpora created via standard machine translation methods. Training
1330 models that are more language-agnostic is an important step for language models in general. Another important step is
1331 the development of domain-agnostic models. The problem of data scarcity for ABSC can be alleviated by transferring
1332 knowledge from other language domains where data is more readily available in large quantities [34, 218]. Pre-training
1333 large language models like BERT on large language datasets from other domains and then fine-tuning the model
1334 parameters using a small domain dataset is a popular technique to handle this problem and can become more and more
1335 useful as larger and more general language models continue to emerge [24].

1340 Early ABSC approaches were systems based almost purely on knowledge bases. As larger labeled datasets became
1341 available, machine learning models such as SVMs became the standard for ABSC. Soon after, deep learning methods
1342 started becoming more popular, but often performed on par with the machine learning methods that used high-quality
1343 handcrafted features. Yet, with the introduction of attention, deep learning methods started rapidly outpacing other
1344 approaches. Until a new revolutionary innovation comes along, we foresee attention-based deep learning models to be
1345 the future for ABSC. New attention models are rapidly being developed in and outside the field of ABSC. For example,
1346 multi-dimensional attention [180] is a general extension of the attention mechanism that allows for a more fine-grained
1347 attention computation. Yet, it has barely been explored for ABSA. Similarly, while multi-head attention is a general
1348 attention extension, it is typically only used in transformer-based architectures [201]. Admittedly, the transformer model
1349 is a highly successful model producing state-of-the-art performances in ABSC [67, 105, 218, 220, 226]. Transformer
1350
1351
1352

models will undoubtedly remain relevant as new transformer-based architectures are proposed that can be used for ABSC, such as the *transformer-XL* [41] and the *reformer* model [107]. Attention models provide an inherent type of interpretability via the attention weights, which is an important aspect that is missing from many black-box algorithms like deep learning models. Another example of more explainable models is the ensembles of symbolic and subsymbolic AI. Models like the Sentic LSTM [125] may be the start for new explainable models for ABSA [185].

Not only can the field of ABSC evolve through advancements in models and datasets, but also via research on the applications of ABSC. For example, most ABSC methods focus either on implicit or explicit aspects. Yet, texts can often contain multiple implicit and explicit aspects [52]. This poses a significant challenge for real-words applications of ABSA methods. As such, further research on methods that can handle both implicit and explicit aspects is required. Furthermore, when considering real-world applications of ABSA, users are typically interested in the sentiment expressed towards aspects aggregated over a review or sets of reviews. Yet, most ABSC methods are focused on sentence-level ABSC, after which the sentiment is aggregated at the review level. However, it has been shown that pure review-level ABSC can outperform sentence-level ABSC with aggregation [45]. As such, more research on the application of review-level ABSC can also help move the field forward. An interesting new application for ABSC is the implementation of ABSA-based search engines. *Smith* [36] is a specialized opinion-based search engine that returns restaurants based on sentiments expressed towards certain aspects in online reviews. Such specialized search engines and other new applications will drive the field of ABSC forward in the future. Another direction that is to be explored stems from the manner in which the task of ABSC is formulated. Currently, the task of ABSC is generally defined in the absence of a time element, meaning that sentiments are considered to be static over time. Although, sentiments towards products are known to typically change over time [136]. This concept has been considered in some sentiment analysis research. For example, in [207], the overall sentiment expressed toward the presidential candidates in the 2012 U.S. election is tracked over time based on Twitter posts. Nonetheless, ABSA over time is an interesting problem that has not seen much attention. However, this development requires new datasets to be produced that are suitable for this task.

REFERENCES

- [1] Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. 2014. SAMAR: Subjectivity and sentiment analysis for Arabic social media. *Computer Speech & Language* 28, 1 (2014), 20–37.
- [2] Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. Aspect based sentiment analysis: category detection and sentiment classification for Hindi. In *17th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2016) (LNCS)*, Vol. 9624. Springer, 246–257.
- [3] Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. Aspect based sentiment analysis in Hindi: resource creation and evaluation. In *10th International Conference on Language Resources and Evaluation (LREC 2016)*. ELRA, 2703–2709.
- [4] Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. A hybrid deep learning architecture for sentiment analysis. In *26th International Conference on Computational Linguistics (COLING 2016)*. 482–493.
- [5] Md Shad Akhtar, Palaash Sawant, Sukanta Sen, Asif Ekbal, and Pushpak Bhattacharyya. 2018. Solving data sparsity for aspect based sentiment analysis Using cross-linguality and multi-linguality. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*. ACL, 572–582.
- [6] Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yaser Jararweh, and Omar Qawasmeh. 2019. Enhancing aspect-based sentiment analysis of Arabic hotels' reviews using morphological, syntactic and semantic features. *Information Processing & Management* 56, 2 (2019), 308–319.
- [7] Mohammad Al-Smadi, Omar Qawasmeh, Mahmoud Al-Ayyoub, Yaser Jararweh, and Brij Gupta. 2018. Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels reviews. *Journal of Computational Science* 27 (2018), 386–393.
- [8] H. Alani, Sanghee Kim, D.E. Millard, M.J. Weal, W. Hall, P.H. Lewis, and N.R. Shadbolt. 2003. Automatic ontology-based knowledge extraction from Web documents. *IEEE Intelligent Systems* 18, 1 (2003), 14–21.
- [9] Tamara Álvarez-López, Milagros Fernández-Gavilanes, Enrique Costa-Montenegro, Jonathan Juncal-Martínez, Silvia García-Méndez, and Patrice Bellot. 2017. A book reviews dataset for aspect based sentiment analysis. In *8th Language & Technology Conference (LTC 2017)*. 49–53.
- [10] Amazon.com, Inc. 2016. <http://www.amazon.com>

- 1405 [11] Michelle Annett and Grzegorz Kondrak. 2008. A comparison of sentiment analysis techniques: Polarizing movie blogs. In *21st Conference of the*
 1406 *Canadian Society for Computational Studies of Intelligence (Canadian AI 2008) (LNCS)*, Vol. 5032. Springer, 25–35.
- 1407 [12] Wazib Ansar, Saptarsi Goswami, Amlan Chakrabarti, and Basabi Chakraborty. 2021. An efficient methodology for aspect-based sentiment analysis
 1408 using BERT through refined aspect extraction. *Journal of Intelligent & Fuzzy Systems* 40, 5 (2021), 9627–9644.
- 1409 [13] Arena Com, Ltd. 2011. <http://www.gsmarena.com>
- 1410 [14] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and
 1411 opinion mining. In *7th International Conference on Language Resources and Evaluation (LREC 2010)*, Vol. 10. ELRA, 2200–2204.
- 1412 [15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd*
 1413 *International Conference on Learning Representation (ICLR 2015)*.
- 1414 [16] Lingxian Bao, Patrik Lambert, and Toni Badia. 2019. Attention and lexicon regularized LSTM for aspect-based sentiment analysis. In *57th Annual*
 1415 *Meeting of the Association for Computational Linguistics (ACL 2019)*. ACL, 253–259.
- 1416 [17] Jeremy Barnes, Patrik Lambert, and Toni Badia. 2016. Exploring distributional representations and machine translation for aspect-based cross-lingual
 1417 sentiment classification. In *26th International Conference on Computational Linguistics (COLING 2016)*. ACL, 1613–1623.
- 1418 [18] Mohammad Ehsan Basiri, Arman Kabiri, Moloud Abdar, Wali Khan Mashwani, Neil Y Yen, and Jason C Hung. 2020. The effect of aggregation
 1419 methods on sentiment classification in Persian reviews. *Enterprise Information Systems* 14, 9-10 (2020), 1394–1421.
- 1420 [19] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on*
 1421 *Neural Networks* 5, 2 (1994), 157–166.
- 1422 [20] Amlaan Bhoi and Sandeep Joshi. 2018. Various approaches to aspect-based sentiment analysis. *arXiv preprint arXiv:1805.01984* (2018).
- 1423 [21] Booking.com. 2020. <https://www.booking.com>
- 1424 [22] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.
- 1425 [23] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and Regression Trees*. CRC press.
- 1426 [24] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry,
 1427 Amanda Askell, et al. 2020. Language models are few-shot learners. In *34th Annual Conference on Neural Information Processing Systems (NIPS*
 1428 *2020)*, Vol. 33. Curran Associates, Inc., 1877–1901.
- 1429 [25] Caroline Brun, Diana Nicoleta Popa, and Claude Roux. 2014. XRCE: Hybrid classification for aspect-based sentiment analysis. In *8th International*
 1430 *Workshop on Semantic Evaluation (SemEval 2014)*. ACL, 838–842.
- 1431 [26] Christopher JC Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and Knowledge Discovery* 2, 2 (1998),
 1432 121–167.
- 1433 [27] Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning.
 1434 *Journal of Artificial Intelligence Research* 63 (2018), 743–788.
- 1435 [28] Erik Cambria. 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems* 31, 2 (2016), 102–107.
- 1436 [29] Erik Cambria, Jie Fu, Federica Bisio, and Soujanya Poria. 2015. AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis.
 1437 In *29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, Vol. 29. AAAI Press, 508–514.
- 1438 [30] Erik Cambria, Yang Li, Frank Z. Xing, Soujanya Poria, and Kenneth Kwok. 2020. *SenticNet 6: Ensemble application of symbolic and subsymbolic AI*
 1439 *for sentiment analysis*. ACM, 105–114.
- 1440 [31] Erik Cambria, Andrew Livingstone, and Amir Hussain. 2012. The hourglass of emotions. In *Cognitive behavioural systems*. LNCS, Vol. 7403.
 1441 Springer, 144–157.
- 1442 [32] Erik Cambria, Björn Schuller, Yunqing Xia, and Catherine Havasi. 2013. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent*
 1443 *Systems* 28, 2 (2013), 15–21.
- 1444 [33] Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *2014 Conference on Empirical*
 1445 *Methods in Natural Language Processing (EMNLP 2014)*. ACL, 740–750.
- 1446 [34] Sun Chi, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *2019*
 1447 *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*. ACL,
 1448 380–385.
- 1449 [35] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of neural machine translation: Encoder-
 1450 decoder approaches. In *8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST 2014)*. ACL, 103–111.
- 1451 [36] Jaehoon Choi, Donghyeon Kim, Donghee Choi, Sangrak Lim, Seongsoo Kim, Jaewoo Kang, and Youngjae Choi. 2015. Smith search: Opinion-based
 1452 restaurant search engine. In *24th International Conference on World Wide Web*. ACM, 187–190.
- 1453 [37] CNET, Inc. 2016. <http://www.cnet.com>
- 1454 [38] Anni Coden, Dan Gruhl, Neal Lewis, Pablo N. Mendes, Meena Nagarajan, Cartic Ramakrishnan, and Steve Welch. 2014. Semantic lexicon expansion
 1455 for concept-based aspect-aware sentiment analysis. In *2014 Semantic Web Evaluation Challenge (SemWebEval 2014) (CCIS)*, Vol. 475. Springer,
 1456 34–40.
- [39] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3 (1995), 273–297.
- [40] Koby Crammer and Yoram Singer. 2001. Pranking with ranking. In *14th Annual Conference on Neural Information Processing Systems (NIPS 2001)*.
 MIT Press, 641–647.

- [41] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. ACL, 2978–2988.
- [42] Michał Daniłuk, Tim Rocktäschel, Johannes Welbl, and Sebastian Riedel. 2017. Frustratingly short attention spans in neural language modeling. In *5th International Conference on Learning Representations (ICLR 2017)*.
- [43] Dayan de França Costa and Nadia Felix Felipe da Silva. 2018. INF-UFG at FiQA 2018 task 1: Predicting sentiments and aspects on financial tweets and news headlines. In *2018 World Wide Web Conference (WWW 2018)*. ACM, 1967–1971.
- [44] Daan de Heij, Artiom Troyanovsky, Cynthia Yang, Milena Zychlinsky Scharff, Kim Schouten, and Flavius Frasinca. 2017. An ontology-enhanced hybrid approach to aspect-based sentiment analysis. In *18th International Conference on Web Information Systems Engineering (WISE 2017) (LNCS)*, Vol. 11570. Springer, 338–345.
- [45] Sophie de Kok, Linda Punt, Rosita van den Puttelaar, Karoliina Ranta, Kim Schouten, and Flavius Frasinca. 2018. Review-aggregated aspect-based sentiment analysis with ontology features. *Progress in Artificial Intelligence* 7, 4 (2018), 295–306.
- [46] Ewelina Dera, Flavius Frasinca, Kim Schouten, and Lisa Zhuang. 2020. SASOBUS: Semi-automatic sentiment domain ontology building using synsets. In *17th Extended Semantic Web Conference (ESWC 2020) (LNCS)*, Vol. 12123. Springer, 105–120.
- [47] DV Nagarjuna Devi, Chinta Kishore Kumar, and Siriki Prasad. 2016. A feature based approach for sentiment analysis by using support vector machine. In *6th International Conference on Advanced Computing (IACC 2016)*. IEEE, 3–8.
- [48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*. ACL, 4171–4186.
- [49] DianPing.com. 2009. <http://www.DianPing.com>
- [50] Hai Ha Do, PWC Prasad, Angelika Maag, and Abeer Alsadoon. 2019. Deep learning for aspect-based sentiment analysis: A comparative review. *Expert Systems with Applications* 118 (2019), 272–299.
- [51] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent Twitter sentiment classification. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*. ACL, 49–54.
- [52] Nikoleta Dosoula, Roel Griep, Rick den Ridder, Rick Slangen, Kim Schouten, and Flavius Frasinca. 2016. Detection of multiple implicit features per sentence in consumer review data. In *12th International Baltic Conference on Databases and Information Systems (DB&IS 2016) (CCIS)*, Vol. 615. Springer, 289–303.
- [53] Mauro Dragoni, Célia da Costa Pereira, Andrea GB Tettamanzi, and Serena Villata. 2016. SMACk: An argumentation framework for opinion mining. In *25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*. IJCAI, 4242–4243.
- [54] Mauro Dragoni, Celia da Costa Pereira, Andrea GB Tettamanzi, and Serena Villata. 2018. Combining argumentation and aspect-based opinion mining: the SMACk system. *AI Communications* 31, 1 (2018), 75–95.
- [55] Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. *Commun. ACM* 63, 1 (2019), 68–77.
- [56] Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence* 77, 2 (1995), 321–357.
- [57] eBay, Inc. 2016. <http://www.ebay.com>
- [58] Magdalini Eirinaki, Shamita Pisal, and Japinder Singh. 2012. Feature-based opinion mining and ranking. *J. Comput. System Sci.* 78, 4 (2012), 1175–1184.
- [59] Abdelkader El Mahdaouy, Abdellah El Mekki, Kabil Essefar, Nabil El Mamoun, Ismail Berrada, and Ahmed Khoumsi. 2021. Deep multi-task model for sarcasm detection and sentiment analysis in Arabic language. In *6th Arabic Natural Language Processing Workshop (WANLP 2021)*. ACL, 334–339.
- [60] Epinions. 2013. <http://www.epinions.com>
- [61] Zulva Fachrina and Dwi H. Widyantoro. 2017. Aspect-sentiment classification in opinion mining using the combination of rule-based and machine learning. In *2017 International Conference on Data and Software Engineering (ICoDSE 2017)*. 1–6.
- [62] Chuang Fan, Qinghong Gao, Jiachen Du, Lin Gui, Ruifeng Xu, and Kam-Fai Wong. 2018. Convolution-based memory network for aspect-based sentiment analysis. In *41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018)*. ACM, 1161–1164.
- [63] Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained attention network for aspect-level sentiment classification. In *2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*. ACL, 3433–3442.
- [64] Flipkart. 2016. <http://www.flipkart.com>
- [65] Jerome H Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* (2001), 1189–1232.
- [66] Chenquan Gan, Lu Wang, Zufan Zhang, and Zhangyi Wang. 2020. Sparse attention based separable dilated convolutional neural network for targeted sentiment analysis. *Knowledge-Based Systems* 188 (2020), 104827.
- [67] Z. Gao, A. Feng, X. Song, and X. Wu. 2019. Target-dependent sentiment classification With BERT. *IEEE Access* 7 (2019), 154290–154299.
- [68] José Antonio García-Díaz, Mar Cánovas-García, and Rafael Valencia-García. 2020. Ontology-driven aspect-based sentiment analysis classification: An infodemiological case study regarding infectious diseases in Latin America. *Future Generation Computer Systems* 112 (2020), 641–657.
- [69] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning* 63, 1 (2006), 3–42.
- [70] Yoav Goldberg. 2017. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers.
- [71] Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *1996 International Conference on Artificial Neural Networks (ICANN 1996)*, Vol. 1. IEEE, 347–352.

- [72] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
- [73] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *11th International Conference on Language Resources and Evaluation (LREC 2018)*. ELRA, 3483–3487.
- [74] Thomas R Gruber. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies* 43, 5-6 (1995), 907–928.
- [75] Shuqin Gu, Lipeng Zhang, Yuexian Hou, and Yin Song. 2018. A position-aware bidirectional attention network for aspect-level sentiment analysis. In *27th International Conference on Computational Linguistics (COLING 2018)*. ACL, 774–784.
- [76] Nicola Guarino and Christopher Welty. 2002. Evaluating ontological decisions with OntoClean. *Commun. ACM* 45, 2 (2002), 61–65.
- [77] Satarupa Guha, Aditya Joshi, and Vasudeva Varma. 2015. SIEL: Aspect based sentiment analysis in reviews. In *9th International Workshop on Semantic Evaluation (SemEval 2015)*. 759–766.
- [78] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *Computing Surveys* 51, 5 (2018), 1–42.
- [79] Deepak Kumar Gupta and Asif Ekbal. 2014. IITP: Supervised machine learning for aspect based sentiment analysis. In *8th International Workshop on Semantic Evaluation (SemEval 2014)*. ACL, 319–323.
- [80] Emma Haddi, Xiaohui Liu, and Yong Shi. 2013. The role of text pre-processing in sentiment analysis. *Procedia Computer Science* 17 (2013), 26–32.
- [81] Nida Manzoor Hakak, Mohsin Mohd, Mahira Kirmani, and Mudasar Mohd. 2017. Emotion analysis: A survey. In *2017 International Conference on Computer, Communications and Electronics (COMPTHELIX 2017)*. IEEE, 397–402.
- [82] SM Shamimul Hasan and Donald A Adjeroh. 2011. Proximity-based sentiment analysis. In *4th International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2011)*. IEEE, 106–111.
- [83] Vasileios Hatzivassiloglou and Kathleen R McKeown. 1997. Predicting the semantic orientation of adjectives. In *8th Conference of the European Chapter of the Association for Computational Linguistics (EACL 1997)*. ACL, 174–181.
- [84] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective attention modeling for aspect-level sentiment classification. In *27th International Conference on Computational Linguistics (COLING 2018)*. ACL, 1121–1131.
- [85] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Exploiting document knowledge for aspect-level sentiment classification. In *56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*. ACL, 579–585.
- [86] Rajalaxmi Hegde and Seema S. 2017. Aspect based feature extraction and sentiment classification of review data sets using incremental machine learning algorithm. In *3rd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB 2017)*. 122–125.
- [87] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE-Wiley.
- [88] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
- [89] Alexander Hogenboom, Daniella Bal, Flavius Frasinca, Malissa Bal, Franciska de Jong, and Uzay Kaymak. 2013. Exploiting emoticons in sentiment analysis. In *28th Annual ACM Symposium on Applied Computing (SAC 2013)*. ACM, 703–710.
- [90] Alexander Hogenboom, Daniella Bal, Flavius Frasinca, Malissa Bal, Franciska De Jong, and Uzay Kaymak. 2015. Exploiting emoticons in polarity classification of text. *Journal of Web Engineering* 14, 1&2 (2015), 22–40.
- [91] Rowan Hoogervorst, Erik Essink, Wouter Jansen, Max Van Den Helder, Kim Schouten, Flavius Frasinca, and Maite Taboada. 2016. Aspect-based sentiment analysis on the web using rhetorical structure theory. In *16th International Conference on Web Engineering (ICWE 2016) (LNCS)*, Vol. 9671. Springer, 317–334.
- [92] John J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *National Academy of Sciences* 79, 8 (1982), 2554–2558.
- [93] Shengluan Hou, Shuhan Zhang, and Chaoqun Fei. 2020. Rhetorical structure theory: A comprehensive review of theory, parsing methods and applications. *Expert Systems with Applications* 157 (2020), 113421.
- [94] Mingqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*. ACM, 168–177.
- [95] Mingqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. In *19th AAAI Conference on Artificial Intelligence (AAAI 2004)*. AAAI Press, 755–760.
- [96] Arfinda Ilmania, Abdurrahman, Samuel Cahyawijaya, and Ayu Purwarianti. 2018. Aspect detection and sentiment classification using deep neural network for Indonesian aspect-based sentiment analysis. In *2018 International Conference on Asian Language Processing (IALP 2018)*. IEEE, 62–67.
- [97] Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*. ACL, 3543–3556.
- [98] Hitkul Jangid, Shivangi Singhal, Rajiv Ratn Shah, and Roger Zimmermann. 2018. Aspect-based financial sentiment analysis using deep learning. In *2018 World Wide Web Conference (WWW 2018)*. ACM, 1961–1966.
- [99] JD.com, Inc. 2013. <http://www.jd.com>
- [100] Soufian Jebbara and Philipp Cimiano. 2016. Aspect-based sentiment analysis using a two-step neural network architecture. In *2016 Semantic Web Evaluation Challenge (SemWebEval 2016) (CCIS)*, Vol. 641. Springer, 153–167.

- [101] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter sentiment classification. In *49th Annual Meeting of the Association for Computational Linguistics (ACL-HLT 2011)*. ACL, 151–160.
- [102] Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *10th European Conference on Machine Learning (ECML 1998) (LNCS)*, Vol. 1681. Springer, 137–142.
- [103] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28, 1 (1972), 11–21.
- [104] Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2018. *Investigations in Computational Sarcasm*. Cognitive Systems Monograph Series, Vol. 37. Springer.
- [105] Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. Adversarial training for aspect-based sentiment analysis with BERT. In *25th International Conference on Pattern Recognition (ICPR 2020)*. IEEE, 8797–8803.
- [106] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *8th International Workshop on Semantic Evaluation (SemEval 2014)*. ACL, 437–442.
- [107] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *8th International Conference on Learning Representations (ICLR 2020)*.
- [108] Efstratios Kontopoulos, Christos Berberidis, Theologos Dergiades, and Nick Bassiliades. 2013. Ontology-based sentiment analysis of Twitter posts. *Expert Systems with Applications* 40, 10 (2013), 4065–4074.
- [109] Marijn Koolen, Toine Bogers, Maria Gäde, Mark Hall, Iris Hendrickx, Hugo Huurdeman, Jaap Kamps, Mette Skov, Suzan Verberne, and David Walsh. 2016. Overview of the CLEF 2016 social book search lab. In *7th International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF 2016) (LNCS)*, Vol. 9822. Springer, 351–370.
- [110] M. H. Krishna, K. Rahamathulla, and A. Akbar. 2017. A feature based approach for sentiment analysis using SVM and coreference resolution. In *2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2017)*. Springer, 397–399.
- [111] Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies* 1, 1 (2009), 1–127.
- [112] Avinash Kumar, Vishnu Teja Narapareddy, Veerubhotla Aditya Srikanth, Aruna Malapati, and Lalita Bhanu Murthy Neti. 2020. Sarcasm detection using multi-head attention based bidirectional LSTM. *IEEE Access* 8 (2020), 6388–6397.
- [113] Akshi Kumar, Simran Seth, Shivam Gupta, and Shivam Maini. 2021. Sentic computing for aspect-based opinion summarization using multi-head attention with feature pooled pointer generator network. *Cognitive Computation* (2021), 1–19.
- [114] Ravindra Kumar, Husanbir Singh Pannu, and Avleen Kaur Malhi. 2020. Aspect-based sentiment analysis using deep networks and stochastic optimization. *Neural Computing and Applications* 32, 8 (2020), 3221–3235.
- [115] Naveen Kumar Laskari and Suresh Kumar Sanampudi. 2016. Aspect based sentiment analysis survey. *IOSR Journal of Computer Engineering* 18, 2 (2016), 24–28.
- [116] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. 1999. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*. LNCS, Vol. 1681. Springer, 319–345.
- [117] Zachary C Lipton, John Berkowitz, and Charles Elkan. 2015. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019* (2015).
- [118] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5, 1 (2012), 1–167.
- [119] Bing Liu and Mingqing Hu. 2004. Opinion mining, sentiment analysis, and opinion spam detection. <https://www.cs.uic.edu/liub/FBS/sentiment-analysis.html> (2004).
- [120] Qiao Liu, Haibin Zhang, Yifu Zeng, Ziqi Huang, and Zufeng Wu. 2018. Content attention model for aspect based sentiment analysis. In *2018 World Wide Web Conference (WWW 2018)*. ACM, 1023–1032.
- [121] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *30th Annual Conference on Neural Information Processing Systems (NIPS 2016)*. Curran Associates, Inc., 289–297.
- [122] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*. ACL, 1412–1421.
- [123] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*. IJCAI, 4068–4074.
- [124] Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*. AAAI Press, 5876–5883.
- [125] Yukun Ma, Haiyun Peng, Tahir Khan, Erik Cambria, and Amir Hussain. 2018. Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis. *Cognitive Computation* 10, 4 (2018), 639–650.
- [126] Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. WWW’18 open challenge: Financial opinion mining and question answering. In *2018 World Wide Web Conference (WWW 2018)*. ACM, 1941–1942.
- [127] Navonil Majumder, Soujanya Poria, Alexander Gelbukh, Md. Shad Akhtar, Erik Cambria, and Asif Ekbal. 2018. IARM: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis. In *2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*. ACL, 3402–3411.

- 1613 [128] William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary*
1614 *Journal for the Study of Discourse* 8, 3 (1988), 243–281.
- 1615 [129] Diana Maynard and Mark Greenwood. 2014. Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis.. In *9th*
1616 *International Conference on Language Resources and Evaluation (LREC 14)*. ELRA, 4238–4243.
- 1617 [130] Donatas Meškele and Flavius Frasincar. 2019. ALDONA: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalised
1618 domain ontology and a neural attention model. In *34th ACM Symposium on Applied Computing (SAC 2019)*. ACM, 2489–2496.
- 1619 [131] Donatas Meškele and Flavius Frasincar. 2020. ALDONAr: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized
1620 domain ontology and a regularized neural attention model. *Information Processing & Management* 57, 3 (2020), 102211.
- 1621 [132] Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International*
1622 *Conference on Learning Representations (ICLR 2013), Workshop Track*.
- 1623 [133] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhersch, and Armand Joulin. 2018. Advances in pre-training distributed word
1624 representations. In *11th International Conference on Language Resources and Evaluation (LREC 2018)*. ELRA.
- 1625 [134] George A Miller. 1995. WordNet: A lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.
- 1626 [135] Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016. Leveraging cognitive features for sentiment analysis.
1627 In *20th SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016)*. ACL, 156–166.
- 1628 [136] Wendy W Moe and David A Schweidel. 2012. Online product opinions: Incidence, evaluation, and evolution. *Marketing Science* 31, 3 (2012),
1629 372–386.
- 1630 [137] Samaneh Moghaddam and Martin Ester. 2010. Opinion Digger: An unsupervised opinion miner from unstructured product reviews. In *19th ACM*
1631 *International Conference on Information and Knowledge Management (CIKM 2010)*. ACM, 1825–1828.
- 1632 [138] Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasani Srinivasan, and Balaraman Ravindran. 2020.
1633 Towards transparent and explainable attention models. In *58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. ACL,
1634 4206–4216.
- 1635 [139] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. 2020. Interpretable machine learning – A brief history, state-of-the-art and challenges.
1636 In *2020 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2020) (CCIS)*,
1637 Vol. 1323. Springer, 417–431.
- 1638 [140] Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. Feature specific sentiment analysis for product reviews. In *13th International Conference*
1639 *on Computational Linguistics and Intelligent Text Processing (CICLing 2012) (LNCS)*, Vol. 7181. Springer, 475–487.
- 1640 [141] Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *2004 Conference on*
1641 *Empirical Methods in Natural Language Processing (EMNLP 2004)*. ACL, 412–418.
- 1642 [142] Budi M Mulyo and Dwi H Widyantoro. 2018. Aspect-based sentiment analysis approach with CNN. In *5th International Conference on Electrical*
1643 *Engineering, Computer Science and Informatics (EECSI 2018)*. IEEE, 142–147.
- 1644 [143] Ambreen Nazir, Yuan Rao, Lianwei Wu, and Ling Sun. 2020. Issues and challenges of aspect-based sentiment analysis: A comprehensive survey.
1645 *IEEE Transactions on Affective Computing* (2020).
- 1646 [144] Thien Nguyen and Kiyooki Shirai. 2015. Aspect-based sentiment analysis using tree kernel based relation extraction. In *16th International Conference*
1647 *on Computational Linguistics and Intelligent Text Processing (CICLing 2015) (LNCS)*, Vol. 9042. Springer, 114–125.
- 1648 [145] Thien Hai Nguyen and Kiyooki Shirai. 2015. PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis. In *2015 Conference*
1649 *on Empirical Methods in Natural Language Processing (EMNLP 2015)*. ACL, 2509–2514.
- 1650 [146] Xinhui Nie, Lizhen Liu, Hanshi Wang, Wei Song, and Jingli Lu. 2013. The opinion mining based on fuzzy domain sentiment ontology tree for
1651 product reviews. *Journal of Software* 8, 11 (2013), 2682–2687.
- 1652 [147] Marek Obitko, Vaclav Snašel, Jan Smid, and V Snašel. 2004. Ontology design with formal concept analysis. In *2nd International Conference on*
1653 *Concept Lattices and Their Applications (CLA 2004) (CEUR Workshop Proceedings)*, Vol. 128. CEUR-WS.org, 1377–1390.
- 1654 [148] Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *7th International Conference on*
1655 *Language Resources and Evaluation (LREC 2010)*, Vol. 10. ELRA, 1320–1326.
- 1656 [149] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2, 1–2 (2008), 1–135.
- 1657 [150] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *2002*
1658 *Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. ACL, 79–86.
- 1659 [151] Nipuna Upeka Pannala, Chamira Priyamanthi Nawarathna, JTK Jayakody, Lakmal Rupasinghe, and Kesavan Krishnadeva. 2016. Supervised
1660 learning based approach to aspect based sentiment analysis. In *16th IEEE International Conference on Computer and Information Technology (CIT*
1661 *2016)*. IEEE, 662–666.
- 1662 [152] Ioannis Pavlopoulos. 2014. Aspect based sentiment analysis. *Athens University of Economics and Business* (2014).
- 1663 [153] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *2014 Conference on Empirical*
1664 *Methods in Natural Language Processing (EMNLP 2014)*. ACL, 1532–1543.
- 1665 [154] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word
1666 representations. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*
(NAACL-HLT 2018). ACL, 2227–2237.
- [155] Pitchfork Media, Inc. 2013. <http://www.pitchfork.com>
- Manuscript submitted to ACM

- 1665 [156] Marco Polignano, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. 2018. An emotion-driven approach for aspect-based opinion mining.
1666 In *9th Italian Information Retrieval Workshop (IIR 2018) (CEUR Workshop Proceedings)*, Vol. 2140. CEUR-WS.org.
- 1667 [157] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub,
1668 Yanyan Zhao, Bing Qin, and Orphée De Clercq. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *10th International Workshop on
1669 Semantic Evaluation (SemEval 2016)*. ACL, 19–30.
- 1670 [158] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based
1671 sentiment analysis. In *9th International Workshop on Semantic Evaluation (SemEval 2015)*. ACL, 486–495.
- 1672 [159] Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4:
1673 Aspect based sentiment analysis. In *8th International Workshop on Semantic Evaluation (SemEval 2014)*. ACL, 27–35.
- 1674 [160] Dragomir R. Radev. 2000. A common theory of information fusion from multiple text sources step one: Cross-document structure. In *1st Annual
1675 Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2000)*, Vol. 10. ACM, 74–83.
- 1676 [161] Ankit Ramteke, Akshat Malu, Pushpak Bhattacharyya, and J. Saketha Nath. 2013. Detecting turnarounds in sentiment analysis: Thwarting. In *51st
1677 Annual Meeting of the Association for Computational Linguistics (ACL 2013)*. ACL, 860–865.
- 1678 [162] Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *1996 Conference on Empirical Methods in Natural Language
1679 Processing (EMNLP 1996)*. ACL, 133–142.
- 1680 [163] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with
1681 weak supervision. In *43rd International Conference on Very Large Data Bases (VLDB 2017)*, Vol. 11. VLDB Endowment, 269–282.
- 1682 [164] Reevoo, Ltd. 2011. <http://www.reevoo.com>
- 1683 [165] Kirk Roberts, Michael A Roach, Joseph Johnson, Josh Guthrie, and Sanda M Harabagiu. 2012. EmpaTweet: Annotating and detecting emotions on
1684 Twitter. In *8th International Conference on Language Resources and Evaluation (LREC 2012)*, Vol. 12. 3806–3813.
- 1685 [166] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. In *2016 Conference
1686 on Empirical Methods in Natural Language Processing (EMNLP 2016)*. ACL, 999–1005.
- 1687 [167] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. Insight-1 at SemEval-2016 task 5: Deep learning for multilingual aspect-based sentiment
1688 analysis. In *10th International Workshop on Semantic Evaluation (SemEval 2016)*. ACL, 330–336.
- 1689 [168] Marzieh Saedi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. 2016. SentiHood: Targeted aspect based sentiment analysis dataset for
1690 urban neighbourhoods. In *26th International Conference on Computational Linguistics (COLING 2016)*. 1546–1556.
- 1691 [169] Gerard. Salton. 1968. *Automatic Information Organization and Retrieval*. McGraw Hill Text.
- 1692 [170] Nuttapon Sanglerdsinlapachai, Anon Plangprasopchok, and Ekawit Nantajeewarawat. 2016. Exploring linguistic structure for aspect-based
1693 sentiment analysis. *Maejo International Journal of Science and Technology* 10, 2 (2016), 142.
- 1694 [171] Bernhard Scholkopf, Alexander J Smola, and Francis Bach. 2018. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and
1695 Beyond*. MIT Press.
- 1696 [172] Martin Schmitt, Simon Steinheber, Konrad Schreiber, and Benjamin Roth. 2018. Joint aspect and polarity classification for aspect-based sentiment
1697 analysis with end-to-end neural networks. In *2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*. ACL, 1109–1114.
- 1698 [173] Kim Schouten and Flavius Frasinca. 2015. The benefit of concept-based features for sentiment analysis. In *2015 Semantic Web Evaluation Challenge
1699 (SemWebEval 2015) (CCIS)*, Vol. 548. Springer, 223–233.
- 1700 [174] Kim Schouten and Flavius Frasinca. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering* 28, 3
1701 (2015), 813–830.
- 1702 [175] Kim Schouten and Flavius Frasinca. 2018. Ontology-driven sentiment analysis of product and service aspects. In *15th Extended Semantic Web
1703 Conference (ESWC 2018) (LNCS)*, Vol. 10843. Springer, 608–623.
- 1704 [176] Kim Schouten, Flavius Frasinca, and Franciska de Jong. 2017. Ontology-enhanced aspect-based sentiment analysis. In *17th International Conference
1705 on Web Engineering (ICWE 2017) (LNCS)*, Vol. 10360. Springer, 302–320.
- 1706 [177] Sam Scott and Stan Matwin. 1998. Text classification using WordNet hypernyms. In *Usage of WordNet in Natural Language Processing Systems*.
- 1707 [178] Terrence J Sejnowski. 2018. *The Deep Learning Revolution*. MIT Press.
- 1708 [179] Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *38th International ACM
1709 SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2015)*. ACM, 959–962.
- 1710 [180] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. DiSAN: Directional self-attention network for RNN/CNN-
1711 free language understanding. In *32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*. AAAI Press, 5446–5455.
- 1712 [181] SIOC. 2021. Semantically Interlinked Online Communities. <http://sioc-project.org/>
- 1713 [182] Rudi Studer, V Richard Benjamins, and Dieter Fensel. 1998. Knowledge engineering: principles and methods. *Data & Knowledge Engineering* 25, 1-2
1714 (1998), 161–197.
- 1715 [183] Jinsong Su, Jialong Tang, Hui Jiang, Ziyao Lu, Yubin Ge, Linfeng Song, Deyi Xiong, Le Sun, and Jiebo Luo. 2021. Enhanced aspect-based sentiment
1716 analysis models with progressive self-supervised attention learning. *Artificial Intelligence* 296 (2021), 103477.
- [184] Andi Suciati and Indra Budi. 2020. Aspect-based sentiment analysis and emotion detection for code-mixed review. *International Journal of Advanced
Computer Science and Applications* 11, 9 (2020), 179–186.
- [185] Yosephine Susanto, Erik Cambria, Bee Chin Ng, and Amir Hussain. 2021. Ten years of sentic computing. *Cognitive Computation* (2021), 1–19.

- 1717 [186] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term
1718 memory networks. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural
1719 Language Processing (ACL-IJCNLP 2015)*. ACL, 1556–1566.
- 1720 [187] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for target-dependent sentiment classification. In *26th International
1721 Conference on Computational Linguistics (COLING 2016)*. ACL, 3298–3307.
- 1722 [188] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *2016 Conference on Empirical
1723 Methods in Natural Language Processing (EMNLP 2016)*. ACL, 214–224.
- 1724 [189] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou. 2016. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on
1725 Knowledge and Data Engineering* 28, 2 (2016), 496–509.
- 1726 [190] Fenna ten Haaf, Christopher Claassen, Ruben Eschauzier, Joanne Tjan, Daniël Buijs, Flavius Frasinca, and Kim Schouten. 2021. WEB-SOBA: Word
1727 embeddings-based semi-automatic ontology building for aspect-based sentiment classification. In *18th Extended Semantic Web Conference (ESWC
1728 2021) (LNCS)*, Vol. 12731. Springer, 340–355.
- 1729 [191] S. Thara and S. Sidharth. 2017. Aspect based sentiment classification: SVD features. In *2017 International Conference on Advances in Computing,
1730 Communications and Informatics (ICACCI 2017)*. 2370–2374.
- 1731 [192] The Internet Movie Database. IMDb.com, Inc. 2008. <http://www.imdb.com>
- 1732 [193] Tun Thura Thet, Jin-Cheon Na, and Christopher SG Khoo. 2010. Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal
1733 of Information Science* 36, 6 (2010), 823–848.
- 1734 [194] Times Movie Reviews, TOI. 2021. <https://timesofindia.indiatimes.com/entertainment/movie-reviews>
- 1735 [195] Kamil Topal and Gultekin Ozsoyoglu. 2016. Movie review analysis: Emotion analysis of IMDb movie reviews. In *2016 IEEE/ACM International
1736 Conference on Advances in Social Networks Analysis and Mining (ASONAM 2016)*. IEEE, 1170–1176.
- 1737 [196] Nam Khanh Tran and Claudia Niedereée. 2018. Multihop attention networks for question answer matching. In *41st ACM SIGIR International
1738 Conference on Research and Development in Information Retrieval (SIGIR 2018)*. ACM, 325–334.
- 1739 [197] Maria Mihaela Trușcă, Daan Wassenberg, Flavius Frasinca, and Rommert Dekker. 2020. A hybrid approach for aspect-based sentiment analysis
1740 using deep contextual word embeddings and hierarchical attention. In *International Conference on Web Engineering (LNCS)*, Vol. 12128. Springer,
1741 365–380.
- 1742 [198] Peter D. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *40th Annual Meeting
1743 on Association for Computational Linguistics (ACL 2002)*. ACL, 417–424.
- 1744 [199] Twitter, Inc. 2014. <http://www.twitter.com>
- 1745 [200] R. Varghese and M. Jayasree. 2013. Aspect based sentiment analysis using support vector machine classifier. In *2013 International Conference on
1746 Advances in Computing, Communications and Informatics (ICACCI 2013)*. 1581–1586.
- 1747 [201] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is
1748 all you need. In *31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*. Curran Associates, Inc., 5998–6008.
- 1749 [202] Viewpoints. 2011. <http://www.viewpoints.com>
- 1750 [203] Duy-Tin Vo and Yue Zhang. 2015. Target-dependent Twitter sentiment classification with rich automatic features. In *24th International Joint
1751 Conference on Artificial Intelligence (IJCAI 2015)*. IJCAI, 1347–1353.
- 1752 [204] Olaf Wallaart and Flavius Frasinca. 2019. A hybrid approach for aspect-based sentiment analysis using a lexicalized domain ontology and
1753 attentional neural models. In *16th Extended Semantic Web Conference (ESWC 2019) (LNCS)*, Vol. 11503. Springer, 363–378.
- 1754 [205] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE:
1755 A stickier benchmark for general-purpose language understanding systems. In *33rd Annual Conference on Neural Information Processing Systems
1756 (NIPS 2019)*, Vol. 32. Curran Associates, Inc.
- 1757 [206] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis
1758 platform for natural language understanding. In *2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. ACL,
1759 353–355.
- 1760 [207] Hao Wang, Doğan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. 2012. A system for real-time Twitter sentiment analysis of 2012
1761 US presidential election cycle. In *ACL 2012 System Demonstrations*. ACL, 115–120.
- 1762 [208] Jingjing Wang, Jie Li, Shoushan Li, Yangyang Kang, Min Zhang, Luo Si, and Guodong Zhou. 2018. Aspect sentiment classification with both
1763 word-level and clause-level attention networks. In *27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*. IJCAI, 4439–4445.
- 1764 [209] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv:2006.04768* (2020).
- 1765 [210] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *2016 Conference
1766 on Empirical Methods in Natural Language Processing (EMNLP 2016)*. ACL, 606–615.
- 1767 [211] Yequan Wang, Aixin Sun, Minlie Huang, and Xiaoyan Zhu. 2019. Aspect-level sentiment analysis Using AS-capsules. In *2019 World Wide Web
1768 Conference (WWW 2019)*. ACM, 2033–2044.
- [212] Zhaoxia Wang, Seng-Beng Ho, and Erik Cambria. 2020. Multi-level fine-scaled sentiment sensing with ambivalence handling. *International Journal
of Uncertainty, Fuzziness and Knowledge-Based Systems* 28, 4 (2020), 683.
- [213] Albert Weichselbraun, Stefan Gindl, Fabian Fischer, Svitlana Vakulenko, and Arno Scharl. 2017. Aspect-based extraction and analysis of affective
knowledge from social media streams. *IEEE Intelligent Systems* 32, 3 (2017), 80–88.
- Manuscript submitted to ACM

- 1769 [214] Sarah Wiegrefe and Yuval Pinter. 2019. Attention is not not Explanation. In *2019 Conference on Empirical Methods in Natural Language Processing*
1770 *and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*. ACL, 11–20.
- 1771 [215] Katarzyna Wójcik and Janusz Tuchowski. 2014. Ontology based approach to sentiment analysis. In *6th International Scientific Conference Faculty of*
1772 *Management Cracow University of Economics (CFM 2014)*.
- 1773 [216] Sixing Wu, Yuanfan Xu, Fangzhao Wu, Zhigang Yuan, Yongfeng Huang, and Xing Li. 2019. Aspect-based sentiment analysis via fusing multiple
1774 sources of textual knowledge. *Knowledge-Based Systems* 183 (2019), 104868.
- 1775 [217] Huan Xu, Constantine Caramanis, and Shie Mannor. 2009. Robustness and regularization of support vector machines. *Journal of Machine Learning*
1776 *Research* 10, 7 (2009), 1485–1510.
- 1777 [218] Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *2019*
1778 *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*. ACL,
2324–2335.
- 1779 [219] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and
1780 tell: Neural image caption generation with visual attention. In *32nd International Conference on Machine Learning (ICML 2015)*, Vol. 37. PMLR,
1781 2048–2057.
- 1782 [220] Qiannan Xu, Li Zhu, Tao Dai, and Chengbing Yan. 2020. Aspect-based sentiment classification with multi-attention network. *Neurocomputing* 388
1783 (2020), 135–143.
- 1784 [221] Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *56th Annual Meeting of the Association for*
1785 *Computational Linguistics (ACL 2018)*. ACL, 2514–2523.
- 1786 [222] Chao Yang, Hefeng Zhang, Bin Jiang, and Keqin Li. 2019. Aspect-based sentiment analysis with alternating coattention networks. *Information*
1787 *Processing Management* 56, 3 (2019), 463–478.
- 1788 [223] Min Yang, Wenting Tu, Jingxuan Wang, Fei Xu, and Xiaojun Chen. 2017. Attention based LSTM for target dependent sentiment classification. In
1789 *31st AAAI Conference on Artificial Intelligence (AAAI 2017)*. AAAI Press, 5013–5014.
- 1790 [224] Steve Yang, Jason Rosenfeld, and Jacques Makutonin. 2018. Financial aspect-based sentiment analysis using deep representations. *arXiv preprint*
1791 *arXiv:1808.07931* (2018).
- 1792 [225] Jianxing Yu, Zheng-Jun Zha, Meng Wang, and Tat-Seng Chua. 2011. Aspect ranking: Identifying important product aspects from online consumer
1793 reviews. In *49th Annual Meeting of the Association for Computational Linguistics (ACL-HLT 2011)*. ACL, 1496–1505.
- 1794 [226] Biqing Zeng, Heng Yang, Ruyang Xu, Wu Zhou, and Xuli Han. 2019. LCF: A local context focus mechanism for aspect-based sentiment classification.
1795 *Applied Sciences* 9 (2019), 3389.
- 1796 [227] Daojian Zeng, Yuan Dai, Feng Li, Jin Wang, and Arun Kumar Sangaiah. 2019. Aspect based sentiment analysis by a linguistically regularized CNN
1797 with gated mechanism. *Journal of Intelligent & Fuzzy Systems* 36, 5 (2019), 3971–3980.
- 1798 [228] Suqi Zhang, Xinyun Xu, Yanwei Pang, and Jungong Han. 2019. Multi-layer attention based CNN for target-dependent sentiment classification.
1799 *Neural Processing Letters* 51 (2019), 2089–2103.
- 1800 [229] Shiliang Zheng and Rui Xia. 2018. Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention.
1801 *arXiv:1802.00892* (2018).
- 1802 [230] Jie Zhou, Jimmy Xiangji Huang, Qin Chen, Qinmin Vivian Hu, Tingting Wang, and Liang He. 2019. Deep learning for aspect-level sentiment
1803 classification: Survey, vision, and challenges. *IEEE Access* 7 (2019), 78454–78483.
- 1804 [231] Lina Zhou and Pimwadee Chaovalit. 2008. Ontology-supported polarity mining. *Journal of the American Society for Information Science and*
1805 *Technology* 59, 1 (2008), 98–110.
- 1806 [232] Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou, and Muhua Zhu. 2009. Multi-aspect opinion polling from textual reviews. In *18th ACM Conference*
1807 *on Information and Knowledge Management (CIKM 2009)*. ACM, 1799–1802.
- 1808 [233] Honglei Zhuang, Fang Guo, Chao Zhang, Liyuan Liu, and Jiawei Han. 2020. Joint aspect-sentiment analysis with minimal user guidance. In *43rd*
1809 *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*. ACM, 1241–1250.
- 1810 [234] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *15th ACM International Conference on Information and*
1811 *Knowledge Management (CIKM 2006)*. ACM, 43–50.
- 1812 [235] Lisa Zhuang, Kim Schouten, and Flavius Frasinca. 2019. SOBA: Semi-automated ontology builder for aspect-based sentiment analysis. *Journal of*
1813 *Web Semantics* 60 (2019), 100544.