

Abstract

The increasing volume of user-generated content on the web has made *sentiment analysis* an important tool for the extraction of information about the human emotional state. A current research focus for sentiment analysis is the improvement of granularity at aspect level, representing two distinct aims: *aspect extraction* and *sentiment classification of product reviews* and sentiment classification of *target-dependent tweets*. *Deep learning* approaches have emerged as a prospect for achieving these aims with their ability to capture both syntactic and semantic features of text without requirements for high-level feature engineering, as is the case in earlier methods. In this article, we aim to provide a comparative review of deep learning for aspect-based sentiment analysis to place different approaches in context.

1. INTRODUCTION

The evolution of web technologies has enabled new means of communication through user-generated content, in the form of blogs, social networks, forums, website reviews, e-commerce websites, etc. (Rana & Cheah, 2016). Following this exponential growth, there has been strong interest from individuals and organisations in data mining technologies to exploit this source of subjective information. One of the most prolific research areas in computer sciences is sentiment analysis, which aims to identify and extract user opinions (Cambria, Poria, Gelbukh, & Thelwall, 2017).

In their seminal work on Aspect-based Sentiment Analysis (ABSA), Hu et al. (2004) argued that the study of sentiment analysis is possible at three levels - document, sentence and entity or aspect. A focus on the document or sentence level presumes that only one topic is expressed in the document or sentence, which is not the case in many situations. A more thorough analysis, therefore, requires investigation at entity and aspect level to identify entities and related aspects and classify sentiments associated with these entities and aspects. Examples of entities include products, services, topics, issues, persons, organizations or events, which usually have several aspects (Jiménez-Zafra, Martín-Valdivia, Martínez-Cámara, & Ureña-López, 2016). For example, a laptop consists of a CPU, screen and keyboard; each also represents an aspect. Furthermore, as an entity is the hierarchy of all aspects, it is also a general aspect. For the purpose of this paper, ABSA signifies sentiment analysis at entity or aspect level.

This kind of fine-grained analysis has generally relied on machine learning techniques, which, although effective, require large, domain specific datasets and manual training data (Hu & Liu, 2004). Furthermore, an aspect may be represented by different words requiring more than one classification algorithm (Schouten & Frasincar, 2016). More recently, experimental work with machine learning methods has shown promise, with Poria, Cambria and Gelbukh (2016) reporting higher accuracy using deep convolutional neural networks, a feature of deep learning (DL), named for its 'deep' multilayer processing technique that uses successive module layers to build on prior output using a backpropagation algorithm (Lecun, Bengio & Hinton, 2015). In each layer, input is converted to numerical representations, which are subsequently classified. Thus, an increasingly higher level of abstraction is achieved (Goodfellow, Bengio & Courville, 2016). A range of algorithms (i.e. deep neural networks (DNN), recurrent neural networks (RNN), convolutional neural networks (CNN), recursive neural networks (RecNN), etc.) facilitate analysis in different fields with deep neural networks particularly suited to fine-grained work due to the significant number of layers of connected processors, activated either by sensors from the environment or by the weighted computations from preceding neurons (Schmidhuber, 2015). An increase in the level of depth leads to higher capability for selective and invariant representation (i.e. extricating different objects) (Lecun et al., 2015). Applied to Natural Language Processing (NLP) tasks, the advantage of DL lies in its independence from expert knowledge and linguistic resources (Rojas-Barahona, 2016) as well as in its superior performance, demonstrated in the areas of 'name-entity recognition' (Chiu & Nichols, 2015; Lample, Ballesteros, Subramanian, Kawakami, & Dyer,

2016; X. Ma & Hovy, 2016; Shen, Yun, Lipton, Kronrod, & Anandkumar, 2017; Strubell, Verga, Belanger, & McCallum, 2017; Yang, Salakhutdinov, & Cohen, 2016); ‘semantic role labelling’ (Do, Bethard, & Moens, 2017; Marcheggiani, Frolov, & Titov, 2017) and ‘Parts-Of-Speech (POS) tagging’ (Ma et al, 2016; Yang et al., 2016).

Early approaches to DL investigated linguistic features, grammatical relations, machine learning classifiers and topic modelling to identify aspects and polarities (Schouten & Frasincar, 2016). More recently, DL methods have been successfully applied to NLP, which makes it interesting to investigate how DL has performed when set fine-grained tasks such as ABSA.

To the best of our knowledge, this work is the first of its type to investigate application of DL methods to ABSA tasks. Recent surveys on DL have not yet covered the areas of ABSA in-depth, even in the work of Tang, Qin, & Liu, (2015), Rojas-Barahona, (2016), Young, Hazarika, Poria, Cambria et al., (2017), and L. Zhang, Wang, & Liu (2018). Deep learning methods are also absent from surveys on ABSA methods, evident from the work of Rana et al. (2016), and Schouten et al. (2016). This paper, rather than repeating established findings from previous surveys, aims to present and compare more recent developments in DL approaches in general and for ABSA in particular. This review is specifically designed for students and researchers in the field of natural language processing, who would like to investigate deep neural networks as well as recent trends in research in ABSA.

The remainder of the paper is organized as follows: Section 2 defines the tasks of ABSA and evaluation measures; Section 3 and 4 analyse DL models for ABSA, investigating in particular how DL affects the interpretation, architecture and performance of ABSA tasks; Section 5 discusses challenges in terms of ABSA and sentiment analysis; the conclusion in section 6 summarises the current landscapes of ABSA and deep learning methods.

2. ASPECT-BASED SENTIMENT ANALYSIS (ABSA)

2.1 The three tasks of ABSA

Pontiki et al. (2016) have assigned three important subtasks to ABSA: (i) Opinion target extraction (OTE), (ii) Aspect category detection (ACD) and (iii) Sentiment Polarity (SP), whereby OTE is concerned with the extraction of aspect terms (i.e. entity or attribute), ACD with identification of associated entities and attributes and (iii) and SP with the clarification of the sentiment polarity of the aspects.

Fig. 1 represents the three tasks of ABSA: The aim of OTE is to extract the opinion target (also referred as "aspect term"¹ [bookmark0](#)) from sentences – in this case "sushi", or "service. For ACD, given the predefined categories, the task is to identify the entity - the aspect of "sushi" as "food" and an attribute denoting "quality". SP identifies the sentiment of a target aspect - "positive" or "negative". It should be noted that the two latter tasks correlate strongly with each other as only through the combination of "great" and "sushi", can both, aspect category and polarity be recognised.

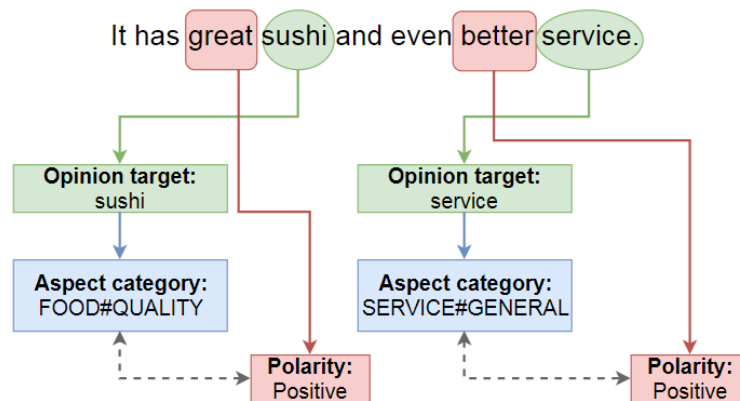


FIGURE 1: Three tasks of ABSA in a sample sentence from SemEval ABSA dataset 2016. The sentence has two opinion targets: sushi & service. The category of "sushi" is "Food", with the attribute being "Quality" and polarity "Positive". The category is "Service", with an attribute of "General" and polarity of "Positive".

¹ In this paper, "opinion target" and "aspect term" are used interchangeably.

In terms of aspects, Hu et al. (2004) identified two types, explicit and implicit, depending on whether or not the aspect words were explicitly stated. Fig. 2 provides an example of an implicit opinion target in the statement "My HP is very heavy". It is clear that polarity and aspect can still be inferred. This implies that OTE focuses only on explicit targets while ACD is concerned with both explicit and implicit aspects.

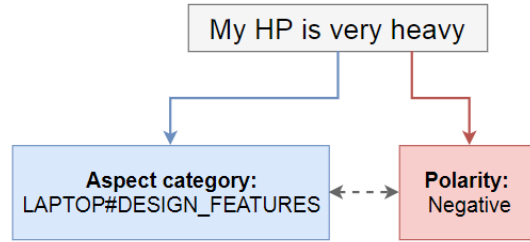


FIGURE 2: The opinion target is not explicitly stated, but the category and polarity of the sentence can still be inferred. Sentence from SemEval data 2016.

2.2 Domain and benchmark datasets

ABSA is mainly applied to customer reviews from websites and e-commerce platforms such as Amazon, Yelp, Taobao and others. These are likely to be product or service reviews and it may be assumed that in each of these only one entity is mentioned but one or more aspects (Saeidi, Bouchard, Liakata & Riedel, 2016). In recent years, systems have been developed for domains such as electronic product reviews (laptop, camera, and phone) and hospitality reviews (restaurant, hotels). A number of benchmark datasets have been made available, including the customer review dataset by Hu et al. (2004) and a number of datasets released by 'International Workshop on Semantic Evaluation' (SemEval 2014-2016) on laptop, camera, restaurant and hotel reviews (Pontiki et al., 2014, 2015, 2016)

Another line of research for ABSA is targeted (or target-dependent) sentiment analysis (Vo & Zhang, 2015), which classifies opinion polarities of a certain target entity mentioned in sentences under scrutiny (normally a tweet). A number of benchmark datasets have been developed for this type such as the Twitter dataset by Dong, Wei, Tan, Tang, Zhou, and Xu (2014). Table 1 below provides a list of publicly available data sets.

TABLE 1: Publicly available datasets for ABSA

N o	Dataset and Author	Domain & Language & Size	Format	Example	URL
1	Customer review data (Hu et al., 2004)	Digital products (EN): 3945 sentences	Text format with tags of aspect terms and polarities (-3, -2, -1, 1, 2, 3)	speaker phone[+2], radio[+2], infrared[+2] ##my favourite features , although there are many , are the speaker phone , the radio and the infrared .	https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
2	SemEval 2014 (Pontiki et al., 2014)	Restaurants (EN): 3841 sentences Laptops (EN): 3845 sentences	XML tag, in which two attributes ("from" and "to") that indicate its start and end offset in the text	<sentence id="81"> <text>Lightweight and the screen is beautiful!</text> <aspectTerms> <aspectTerm term="screen" polarity="positive" from="20" to="26"/> </aspectTerms> </sentence>	http://alt.qcri.org/semeval2014/task4/
3	SemEval 2015 (Pontiki et al., 2015)	Laptop (EN): 450 reviews (2500 sentences)	XML tag of {E#A, polarity}	<sentence id="1004293:0"> <text> Judging from previous posts this used to be a good place, but not any longer. </text>	http://alt.qcri.org/semeval2015/task12/
		Restaurant (EN): 350 reviews (2000 sentences)	XML tag of {E#A, OTE, polarity}	<Opinions> <Opinion target="place" category="RESTAURANT#GENERAL" polarity="negative" from="51" to="56"/> </Opinions>	
		Hotel (EN): 30 reviews (266 sentences) - no training data	XML tag of {E#A, OTE, polarity}	</sentence>	
4	SemEval 2016 (Pontiki et al., 2016)	Laptop (EN): 530 reviews (3308 sentences) Mobile phone (CH): 200 reviews (9521 sentences) Camera (CH): 200 reviews (8040 sentences)	XML tag of {E#A, polarity}	<sentence id="1661043:4"> <text>Decor is charming.</text> <Opinions> <Opinion target="Decor" category="AMBIENCE#GENERAL" polarity="positive" from="0" to="5"/> </Opinions> </sentence>	http://alt.qcri.org/semeval2016/task5/
		Restaurant (DU): 400 reviews (2286 sentences) Mobile phone (DU): 270 reviews (1697 sentences) Restaurant (FR): 455 reviews (2429 sentences) Restaurant (RU): 405 reviews (4699 sentences) Restaurant (ES): 913 reviews (2951 sentences) Restaurant (TU): 339 reviews (1248 sentences) Hotel (AR): 2291 reviews (3309 sentences)	XML tag of {E#A, OTE, polarity}		
5	ICWSM 2010 JDPa Sentiment Corpus for the Automotive Domain (Kessler, Eckert, Clark, & Nicolov, 2010)	Automotive & digital devices: 515 documents (19,322 sentences)	XML tags (<mentionClass> indicate the aspect term)	<classMention id="StructuralSentiment_Instance_40033"> <mentionClass id="Mention.Person">Mention. Person</mentionClass> <hasSlotMention id="StructuralSentiment_Instance_40395" /> </classMention>	https://verbs.colorado.edu/jdpacorporus/
6	Darmstadt Service Review Corpus (Toprak, Jakob, & Gurevych, 2010)	Online university & online service review: 118 reviews (1151 sentences)	MMAx format	<markables xmlns="www.eml.org/NameSpaces/OpinionExpression"> <markable id="markable_38" span="word_118..word_119" referent="empty" annotation_type="holder" mmax_level="opinionexpressio	https://www.ukp.tu-darmstadt.de/fileadmin/user_upload/Group_UKP/data/sentiment-analysis/Darmstadt

No	Dataset and Author	Domain & Language & Size	Format	Example	URL
				<pre> n" isreference="true" /> <markable id="markable_40" span="word_123" annotation_type="opinionexpression" opinionholder="markable_38" mmax_level="opinionexpression" opiniontarget="markable_39" strength="average" polarity="positive" opinionmodifier="empty" /> <markable id="markable_37" span="word_126" annotation_type="target" mmax_level="opinionexpression" isreference="false" /> <markable id="markable_39" span="word_124" referent="markable_37" annotation_type="target" mmax_level="opinionexpression" isreference="true" /> </markables> </pre>	stadtServiceReviewCorpus.zip
7	FiQA ABSA (Maia et al., 2018)	Financial news headlines: 529 samples; financial microblogs: 774 annotated posts	JSON nodes with sentiment score ranged from -1 to 1, "target" indicates opinion target, and "aspect" indicates aspect categories according to different level	<pre> "1": { "sentence": "Royal Mail chairman Donald Brydon set to step down", "info": [{ "snippets": "[set to step down]", "target": "Royal Mail", "sentiment_score": "-0.374", "aspects": ["Corporate/Appointment"] }] </pre>	https://sites.google.com/view/fiqa/home
8	Target-dependent Twitter sentiment classification dataset (Dong et al., 2014)	Twitter comments: training data has 6,248 tweets, and testing data has 692 tweets			http://goo.gl/5Enpu7

2.3 Previous approaches to ABSA tasks

Earlier approaches to identification of OTE and ACD (for example, Hu et al., 2004) were based on frequency of nouns and noun phrases in the text, with the assumption that aspect words were more likely to be repeated. The limitation of this approach is the dependency on the frequency of certain word categories (nouns/noun phrases), which may work well if the text contains high-frequency terms, but may fail if terms are infrequent (Rana et al., 2016).

Others extracted OTE and ACD by exploiting opinion and target relations. Poria, Chaturvedi, Cambria, and Bisio (2016) and Piryani, Gupta, and Singh (2017) focused on rule-based linguistic patterns, including stop words and negation, etc. The assumption here was that it is easier to detect sentiment than aspect words. The authors proposed a set of opinion rules to first identify a sentiment word, and then use grammatical relations to build the syntactic structure of sentences and to detect the aspect. The final step consists of refinement where infrequent words are added and irrelevant aspects are removed. The lexical relation between sentiment words and aspects is the key element in this method, which is able to identify low-frequency aspects (Schouten et al., 2016). However, a drawback is reliance on grammatical accuracy of the sentence and the requirement for manipulation (Poria, Cambria et al., 2016).

Topic modelling has been widely used to perform ACD tasks, with the most popular model being Latent Dirichlet Allocation (LDA) as implemented in Poria et al. (2015), Alam, Ryu, and Lee (2016), García-Pablos,

Cuadros, and Rigau (2018), and Weichselbraun, Gindl, Fischer, Vakulenko and Scharl, (2017). The basis of LDA is the introduction of a latent variable "topic" between the variables "document" and "word", whereby each document contains a random mix of topics, and each topic is constructed through relevant words. While this approach is appropriate to detect aspects at the document level, these may be too broad to capture fine-grained aspects (Schouten et al., 2016). Furthermore, it was also observed that in the majority of studies, such as that by Poria, Chaturvedi, et al. (2016), the topics are unlabelled and require manual evaluation.

For all three tasks, supervised learning approaches, characterised by the use of classifiers built from linguistic resources, predominated (Fernández-Gavilanes et al., 2016). A substantial number of studies in SemEval 2014-2016 chose classifiers such as Maximum Entropy (ME), Conditional Random Field (CRF) and Support Vector Machine (SVM) for aspect detection and polarity. Top performers include CRF models in Chernyshevich (2014), Toh and Su, (2016), and Brun, Perez, and Roux (2016). SVM models were found in Wagner et al. (2014) and Vicente, Saralegi, and Agerri (2017) and ME in Saias (2015). Supervised machine learning approaches were also used at aspect level in sentiment analysis of movie reviews (SVM classifier in Manek, Shenoy, Mohan, and Venugopal (2017), and Naive Bayes in Parkhe and Biswas (2016)). A recent study by Akhtar, Gupta, Ekbal, and Bhattacharyya (2017) presented a cascaded framework based on two steps: first base learning algorithms as classifiers ME, CRF, SVM followed by an ensemble of feature selection and classifier using particle swarm optimization. While the machine learning is simple and quite efficient, it shows certain weaknesses, including the requirement for large datasets, reliance on manual training data, and non-replicable results for other domains (Fernández-Gavilanes et al., 2016). Furthermore, aspects can be represented by different words, which means one classification algorithm is insufficient (Schouten et al., 2016).

2.4 Evaluation measures of ABSA tasks

International Workshops on Semantic Evaluation are promoting the development of aspect-level sentiment analysis (Pontiki et al., 2014, 2016, 2015) providing controlled evaluation methodology and shared datasets for all participants. For the measurement of the efficiency of a classifying model, four main measurements were proposed: Precision (P), Recall (R), F-score (F1) and Accuracy (Acc).

$$P = \frac{TP}{TP + FP} \quad (1)$$

$$R = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = \frac{2PR}{P + R} \quad (3)$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

TP (true positives) and TN (true negatives) are the respective labels and non-labels that are assigned by the system (rather than by humans); FP (false positives) are those labels assigned by the system but not by human annotators, FN (false negatives) are those labels that human annotators assigned and which were not detected by the system.

Precision measures the percentage of labels correctly assigned by the system. Recall measures the percentage of labels found by the system. Accuracy and F-score represent true results (TF and TN).

For OTE and ACD tasks, the F-score is frequently used as the tasks are similar to information retrieval and, to evaluate SP, accuracy is applied.

3 DEEP LEARNING FOR ABSA

Deep learning (DL) is a machine learning method based on learning data representation through algorithms - artificial neural and belief networks - based on multiple layers of modules where input is analysed and classified, with output from one layer fed into the next layer as input. This process is known as backpropagation (Lecun et al. , 2015), whereby activation initiates the backward computation of the gradient of an objective function (Schmidhuber, 2015). Irrelevant of the type of input (i.e. sound, image or text), it is converted to numerical vectors, then clustered into meaningful classification. As each successive layer is corresponding to an increased level of abstraction, DL can be said to represent "nested hierarchies" of simpler concepts (Goodfellow et al., 2016). Another feature is that its depth level can be seen as similar to multi-stage programming, in which each layer is a computer's memory state after executing a set of

instructions (Goodfellow et al., 2016). By increasing the depth level, the system capacity to selectively and invariantly represent is enhanced (Lecun, Bengio, & Hinton, 2015).

Deep neural networks (DNNs) are good examples of DL and are the focus of this paper. Deep neural networks are types of artificial neural networks (algorithms) which include a significant number of layers of “neurons” or connected processors, activated either by sensors from the environment or by the weighted computations from previous neurons (Schmidhuber, 2015). For DL, as for machine learning approaches in general, datasets are often divided into three components: training, validation and test datasets, conforming to general machine learning principles. Lecun et al. (2015) mapped out the training process as a conversion of input into vector scores, regardless of type of input (i.e. images). Initially, an error score appears which needs to be reduced by training the algorithm to more closely conform to the set parameters (weights) for the target word (or image). Adjustments are subsequently made by the machine to reduce the error. The error-adjustment trigger is a ‘gradient vector’, which responds to the manipulation of the parameters, which needs to counter-balance the error.

3.1 Multiple layers of DNN

Deep neural network (DNN) approaches to NLP are distinguished by (i) dense word embeddings; (ii) multiple hidden layers between the input and output; and (iii) output units (Fig. 3).

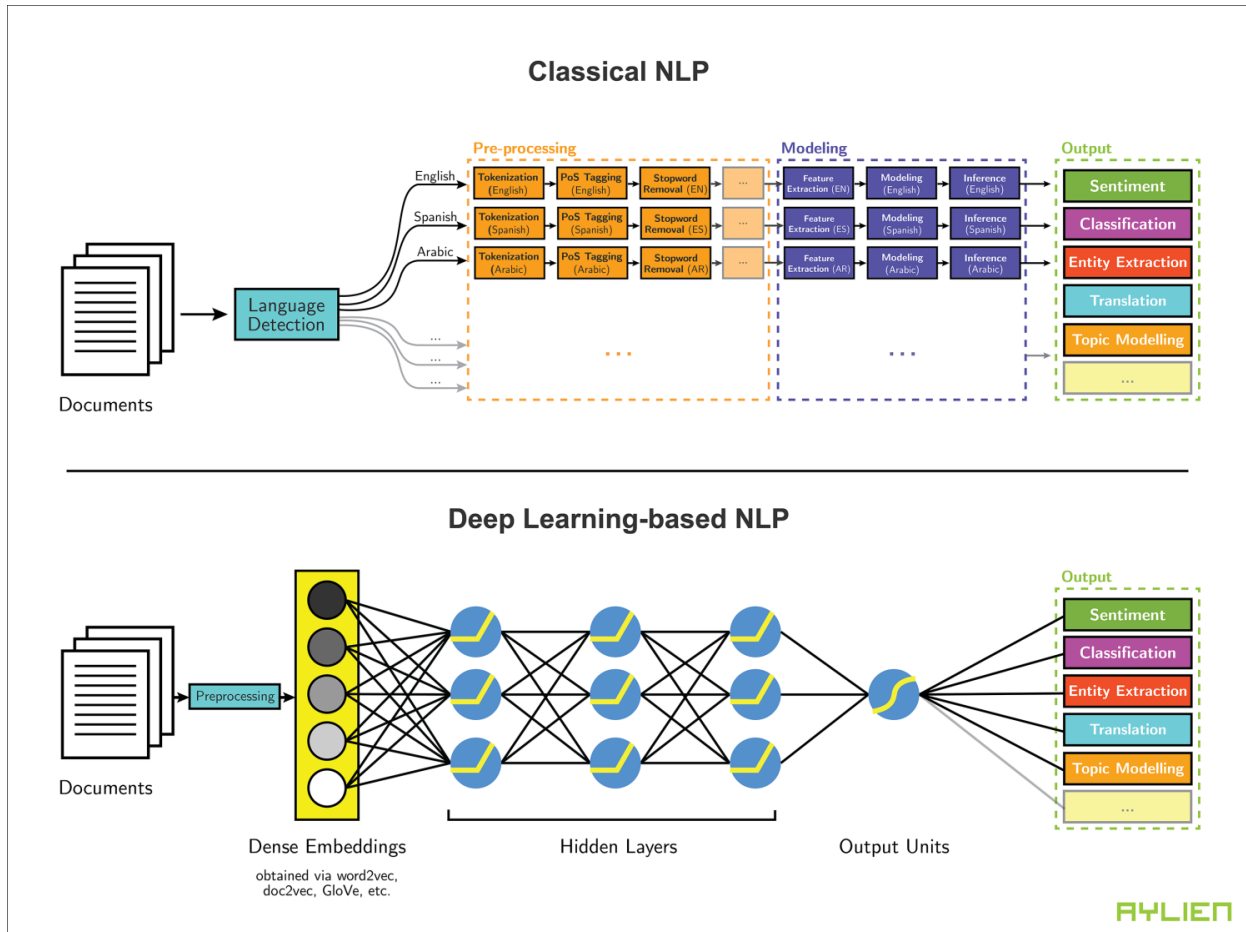


FIGURE 3: Comparison of classic machine learning and deep learning processes for NLP. Deep learning architecture is characterized by dense embeddings and hidden layers - adapted from Thanaki (2017).

Word embeddings are d-dimensional space representations of words, encoded as dense numerical vectors (Rojas-Barahona, 2016). These vectors, Levy and Goldberg (2014) argue, establish the likelihood of a word appearing within a specific word matrix (i.e. with associated words). One of the first word embedding models was that of Bengio et al. (2006) who proposed a neural probabilistic language model with shared lookup table. Thus, given a word and its preceding words, the model looks up its continuous vector, and then

feeds the information into a feed-forward neural network to predict the probable function of the next word. In an attempt to reduce feature engineering, many DNN based studies have used word embeddings as the only feature (such as Liu, Joty, & Meng, 2015). In recent DNN models, word embeddings are typically pre-trained but not task-specific data so that the learning word vectors can capture general syntactical and semantic information (T. Chen, Xu, He, & Wang, 2017; P. Liu et al., 2015; Poria, Cambria, et al., 2016). There are different models for word embeddings, such as Word2Vec (Mikolov, Corrado, Chen, & Dean, 2013) that encode contextual information using continuous Bag-Of-Words (CBOW) and skip-gram models. Word embeddings are discussed further in Section 3.3.1.

The second feature - hidden layers - can be constructed in different forms and architectures, i.e. feed-forward networks and recurrent or recursive networks (Goldberg, 2016). Each hidden layer is composed of multiple neurons, stacked together to compute non-linear outputs (Lecun et al., 2015). Generally, the higher layers evolve through training to exploit the complex compositional nonlinear functions of the lower layers and, hence, capture more abstract representations than the lower layers (Goldberg, 2016).

The computation of hidden features starts with neurons, which take n input to produce a single output. Considering the inputs $x_1, x_2, \dots \in \mathbb{R}$ with n associated parameters (or weights) $\mathbf{w}_1, \mathbf{w}_2, \dots \in \mathbb{R}$ and a bias scalar $b \in \mathbb{R}$, the activation of the neuron is written as $a = \sum_i \mathbf{w}_i x_i + b$. Thus, the output o is calculated with the activation function:

$$o = s(a) = s\left(\sum_i \mathbf{w}_i x_i + b\right) \quad (5)$$

The activation function is a non-linear function, either the sigmoid function (Equation 6), the hyperbolic tangent function (Equation 7), or the rectified linear function (Equation 8).

$$\text{sigmoid}(a) = \frac{1}{1 + e^{-a}} \quad (6)$$

$$\tanh(a) = \frac{e^{2a} - 1}{e^{2a} + 1} \quad (7)$$

$$\text{ReLU}(a) = \max(0, a) \quad (8)$$

For layers of neurons, given layer l with m neurons, each with n -dimensional input vector $x \in \mathbb{R}^n$ with n -dimensional associated weight matrix $W \in \mathbb{R}^{m \times n}$ and a bias scalar $b \in \mathbb{R}^m$ and the activation function s (either *sigmoid*, *tanh* or *ReLU*), the computation of l can be written as:

$$l = s(Wx + b) \quad (9)$$

The third feature - output units - represents the distributed probability over all labels or classes. Supposing the last layer is \mathbf{z} and there are K labels/classes, the probability for the label i can be obtained using the *softmax* function as set out below:

$$y_i = \text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}} \quad (10)$$

For OTE, categories can be represented as similar to sequence tagging of IBO labels ("B" is the start of the aspect term, "I" is the continuation of the aspect term and "O" is not an aspect term) (Table 2).

TABLE 2: An example sentence with labels in IOB format with the opinion target/aspect term as "onion rings".

Words:	The	onion	rings	are	great	!
Labels:	O	B	I	O	O	O

For ACD, for a given category or attribute, the label can be represented as a binary $T = \{\text{category, non-category}\}$. For sentiment polarity, categories might be the set of 4 way polarities as in SemEval tasks $T = \{\text{positive, negative, neutral, conflict}\}$ or simple binary polarities, such as $T = \{\text{positive, negative}\}$. For all three tasks, the output units of the DNN model can return the probability to assign a given label to each input, whereby the label with the highest probability represents the result of the prediction.

To summarise the above discussion, Lecun et al. (2015) suggested that DNNs with distributed representation are able to generalise new combinations of learnt features beyond what has been learned in the training phase. Therefore, in contrast to standard machine learning, DNN models attempt to automatically learn good features or representations (Rojas-Barahona, 2016). Unlike traditional methods, DNN models also

do not require much feature engineering, and if the right model is chosen, it has more robust extraction and representation capacities (Araque, Corcuera-Platas, Sánchez-Rada, & Iglesias, 2017).

In the sections below, major DNN models applied to ABSA tasks will be reviewed, including convolutional neural networks (CNN), recurrent neural networks (RNN), recursive neural networks (RecNN), and hybrid models in Section 3.2 - 3.5. In each section, a review of architecture will be provided, following by the application of models to ABSA tasks.

3.2 Initialization of Input Vectors

Before reaching the first hidden layer of the DNN, the input layer is encoded with a distributed representation, or word embeddings, which represent each word as a low-dimensional, real-valued and continuous vector to encode its semantic and syntactic properties (Tang, Wei, et al., 2016).

3.2.1 Word embeddings vectors

One of the first word embedding models is by Bengio, Schwenk, Senécal, Morin, & Gauvain (2006) who proposed a neural probabilistic language model with a shared lookup table. Given a word and its previous words, the model can look up its continuous vector, feed the vector to a feed-forward neural network and predict the probability function for its next word. Assuming a sequence of T words w_1, w_2, \dots, w_T with n previous words fed into the model, the model can predict the probability p of the words u_t based on finding the model parameter θ that maximises the objective function J :

$$J = \frac{1}{T} \sum_{t=1}^T \log g(w_t, w_{t-1}, \dots, w_{t-n+1}; \theta) + R(\theta) \quad (11)$$

where $R(\theta)$ is the regularisation term, and $g(w_t, w_{t-1}, \dots, w_{t-n+1}; \theta)$ can be estimated by the *softmax* function as $p(w_t | w_{t-1}, \dots, w_{t-n+1})$.

Mikolov et al. (2013) developed the word2vec with two different neural network models for creating word embeddings for training on large corpora: a bag-of-words based model (CBOW) that obtains word context from sequential word context; and the skip-gram model that predicts the word embeddings from neighbouring words (Figure 4).

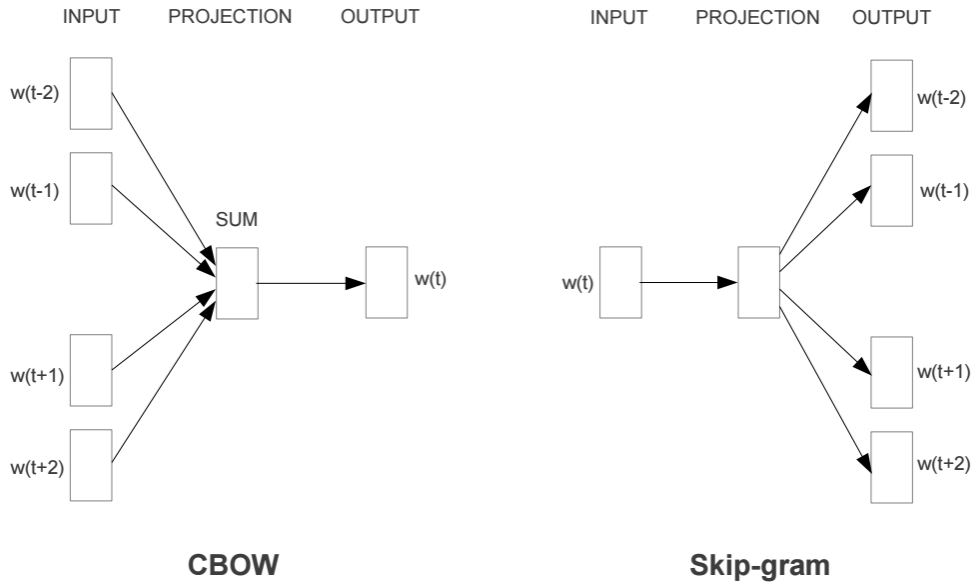


FIGURE 4: CBOW and skip-gram model. Figure from Mikolov et al. (2013).

The CBOW for the target as the word w_t at time step t , the model receives a window of n words around w_t , and the loss function J can be written as:

$$J = \frac{1}{T} \sum_{t=1}^T \log p(w_t | w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n}) \quad (12)$$

In contrast, the skip-gram model uses the centre word w_t to predict the neighbouring words w_{t+j} . In this case, the objective function is:

$$J = \frac{1}{T} \sum_{t=1}^T \sum_{-n \leq j \leq n, j \neq 0} \log p(w_{t+j} | w_t) \quad (13)$$

In addition to word2vec framework, a number of software have been developed for training word embeddings such as GloVe developed by Stanford University or fastText developed by Facebook. Pre-trained word vectors have also been developed such as SENNA (based on the Wikipedia corpus); Google (based on the Google News corpus); Amazon (based on the Amazon corpus); GloVe (based on Wikipedia and Twitter); SSWE (based on Twitter with inclusion of emoticons) (Table 3). Others, such as Poria, Cambria, et al. (2016) and Wang et al. (2016) created their own embeddings by applying word2vec to a selected corpora.

TABLE 3: Pre-trained word embeddings datasets

Pre-trained Word Embeddings	Authors	Dimension	Vocabulary size
SENNA/Wikipedia	Collobert et al. (2011)	50d	130K
word2vec CBOW/Google News	Mikolov et al. (2013)	300d	3M
word2vec CBOW/Amazon	Liu et al. (2015)	50d, 300d	1M
SSWE_h, SSWE_r, SSWE_u	Tang et al. (2014)	50d	137K
GloVe/ Wikipedia 2014 + Gigaword 5	Pennington, Socher, & Manning (2014)	50d, 100d, 200d, 300d	400K
GloVe Twitter	Pennington et al. (2014)	25d, 50d, 100d, 200d	1.2M
Learning word vectors for 157 languages/fastText	Grave, Bojanowski, Gupta, Joulin and Mikolov (2018)	Varied between languages	Varied between languages

There are a number of ways to initialize word embeddings, including random initialization (i.e setting the embedding vectors to random values), or pre-trained (i.e. tuning the vectors so that similar words will obtain similar vectors) (Goldberg, 2016). The recent high performing models typically opt for pre-trained word embeddings and fine-tune them to better initialize the model. As discussed in Liu et al. (2015), the random approach can lead to stochastic gradient descent in the local minima, and if the pre-trained word beddings are employed from readily available resource without tuning, this may not exploit automatic feature learning capacity of DNNs. Experiments from studies such as Liu et al. (2015), Poria, Cambria, et al. (2016), Jebbara and Cimiano (2016) show that the model will be beneficial with the initialization of pre-trained work embeddings and fine-tune them in training, for example, only using pre-trained word embeddings contributed a gain of 6-9% in aspect term extraction (Poria, Cambria, et al., 2016) or 2% in sentiment polarity (Wu, Gu, Sun, & Gu, 2016). Wang et al. (2016) observed average gains of 5% in evaluation criteria thanks to fine-tuning word embeddings according to dependency-based word vectors and specialized features in their RNN models.

Furthermore, pre-training word embeddings in large corpora with similar domain have shown important to the successful implementation of DNN models (Ruder, Ghaffari, & Breslin, 2016). For Liu et al. (2015), Poria, Cambria, et al. (2016), for the task of customer review with less formal texts than Wikipedia and Google news corpus, a word embeddings scheme that contains more opinion specific words such as Amazon have better performance. Meanwhile, in Twitter target ABSA, Tang et al. (2016) showed that a sentiment-specific word embeddings (SSWE) have better performance.

3.2.2 Featuring vectors

As mentioned above, in contrast to previous approaches, deep learning rarely relies on feature engineering, parser, or positional information, but solely on language input (Young et al., 2017). However, in order to generate more salient performance, a number of feature vectors are fed into the DNN, together with the word embeddings. The most common features are summarised below.

Part-Of-Speech (POS) and chunk tags

One observation from Hu et al. (2004), there is high probability that the aspect terms are nouns or noun chunks, which reveals the importance of POS features in OTE. The number of classifications for POS tagging

varies (i.e. 6 tags according to Stanford Tagger in Poria, Cambria, et al. (2016); 4 tags according to Stanford Tagger in Liu et al. (2015), Ye, Yan, Luo, and Chao (2017); or even 45 tag based on Penn Tagger (Jebbara & Cimiano, 2017; Zainuddin, Selamat, & Ibrahim, 2018). In Liu et al. (2015) and Ye et al. (2017), the four POS types are *noun*, *adjective*, *verb* and *adverb*, and the five classes of chunks are: *NP* (noun phrase), *VP* (verb phrase), *PP* (prepositional phrase), *ADJP* (adjective phrase) and *ADVP* (adverb phrase).

Generally, k tags, representing k parts of speech, can be encoded as k -dimensional binary vectors and then concatenated with the word embeddings vectors before being fed to the neural network models. Experiments have shown that POS tagging and word chunks complement word embeddings play a major role in aspect extraction, contributing from 1% (Liu et al., 2015; Poria, Cambria, et al., 2016; Ye et al., 2017) to 4% gain (Feng, Cai, & Ma, 2018).

Commonsense Knowledge

Another feature suggested by Jebbara and Cimiano (2017) and Y. Ma, Peng, and Cambria (2018) to improve both aspect extraction and sentiment classification is common-sense knowledge through SenticNet. This base consists of over 50,000 concepts with associated affective properties (Y. Ma et al., 2018) which are represented by real-value scores consisting of 5 sentics: pleasantness, attention, sensitivity, aptitude, and polarity, which can imply semantic links to aspect and sentiment (Jebbara et al., 2017). An example given by Y. Ma et al. (2018) is the concept "cupcake" has the property "KindOf-food" that can be related to 'restaurant' or 'food quality', but also emotions, e.g., "Arise-joy" that supports sentiment classification.

By including them as 5 feature vectors for each concept, those studies have shown improvement. Y. Ma et al. (2018) suggested the Sentic LSTM significantly outperformed a baseline LSTM. Jebbara and Cimiano (2017) observed that while sentics did not contribute to aspect term extraction, the usage of sentic vectors contributed to 4% gain in the model for sentiment analysis and considerably reduced the training time.

3.3 Training process of DNNs

A neural network is trained through a backpropagation process in which the gradients of all parameters are computed backward and updated with stochastic gradient descent (Goldberg, 2016, 2017).

Let $x = x_1, x_2, \dots, x_n$ be the input, and $y = y_1, y_2, \dots, y_n$ be the output from the machine learning algorithm with the actual labels be $\hat{y} = \hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$, the goal of the algorithm is to estimate a function $y = f(x)$ that matches the inputs with their correct label. The loss function is employed during the training phrase to calculate a numerical score \mathcal{L} that is loss when predicting output y with respect to \hat{y} . In this sense, the parameters of the function (weight matrix $W \in \mathbb{R}^{m \times n}$ and a bias scalar b) are to be set to minimise the loss \mathcal{L} . The loss function for the whole sample is calculated with respect to the parameter θ as the average loss:

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(x_i; \theta), \hat{y}_i) \quad (14)$$

In addition, while minimizing the cost, the model maybe overfitting. Thus, the algorithm combines another function $R(\theta)$ to measure the complexity. Therefore, the goal of the function then is to set the θ to minimize the loss value while keeping a low complexity $R(\theta)$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\theta) = \underset{\theta}{\operatorname{argmin}} \left(\frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(x_i; \theta), \hat{y}_i) + \lambda R(\theta) \right) \quad (15)$$

Following the *softmax* function as above, the categorical cross-entropy loss is used as the loss function:

$$\mathcal{L}(y, \hat{y}) = - \sum_i \hat{y}_i \log(y_i) \quad (16)$$

Because the scores y_i are not negative with the sum of one, the cross-entropy loss produces not only the label prediction but also the distribution.

In some cases, after the features have been obtained from the neural network models, non-neural classifiers are incorporated as the final layer to obtain optimum performance (L. Zhang et al., 2018). The most extensively used classifiers in recent years include Support Vector Machine (SVM) and Conditional Random Fields (CRF) classifiers, with examples in ABSA tasks as CRF in T. Chen et al. (2017), Xu, Lin,

Wang, Yin and Wang (2017), Mai & Le (2018), or SMV in Akhtar, Kumar, Ekbal and Bhattacharyya (2016), and Dong et al. (2014). All these are discriminative models, which learn the most useful features of the input to predict the output, and are trained with different loss functions (Goodfellow et al., 2016).

The SVM model is a classifier that outputs the identity of different classes based on a linear function. The incorporation of the SVM with the neural network model can be implemented selecting the label with the highest score of y expressed as:

$$k = \underset{i}{\operatorname{argmin}} y_i \quad (17)$$

Thus, for the highest scoring label $k = \underset{i}{\operatorname{argmin}} y_i$ and the correct label $\hat{k} = \underset{i}{\operatorname{argmin}} \hat{y}_i$, the SVM loss function is:

$$\mathcal{L}(y, \hat{y}) = \max(0, 1 - (y_{\hat{k}} - y_k)) \quad (18)$$

Comparing the *softmax* layer with SVM, one advantage of the latter is that it is useful under conditions of hard decision rule (i.e. when it is not necessary to estimate the probability of each label) (Goldberg, 2017). Nevertheless, applied to classification task such as sentiment analysis, its performance maybe hindered by its “sparse” and “discrete” features, which makes it difficult to transfer information regarding relationships and coherence of chunks or sentences (Tang, Qin, & Liu, 2015b).

CRF assumes that the output y is connected by undirected edges in an undirected graph (Figure 5). In this sense, the CRF represents the score of a given label sequence (or the clique potential) as a conditional probability that is proportional to the input sequence (T. Chen et al., 2017) as:

$$\operatorname{score}_{CRF}(x, y) = p(y|x) = \frac{1}{Z_x} \prod_{s \in S(y, x)} \phi_s(y_s, x_s) \quad (19)$$

where Z_x is the normalization, $S(y, x)$ the set of cliques of the undirected graph where the outputs are connected and $\phi_s(y_s, x_s)$ is the clique potential. The loss function is calculated as:

$$\mathcal{L}(y, \hat{y}) = -\log \operatorname{score}_{CRF}(x, y) \quad (20)$$

For the CRF model, the labels of each consecutive point can influence others (Goldberg, 2016), which overcomes the disadvantage of *softmax* which features independent labels (Tutubalina & Nikolenko, 2017). Thus, it can be inferred that CRFs can take advantage of the entire sentence sequence to estimate probability for the sentence labelling making CRF a frequent final classification layer of bidirectional RNNs (T. Chen et al., 2017; Irsoy & Cardie, 2014; Lample et al., 2016; P. Liu et al., 2015).

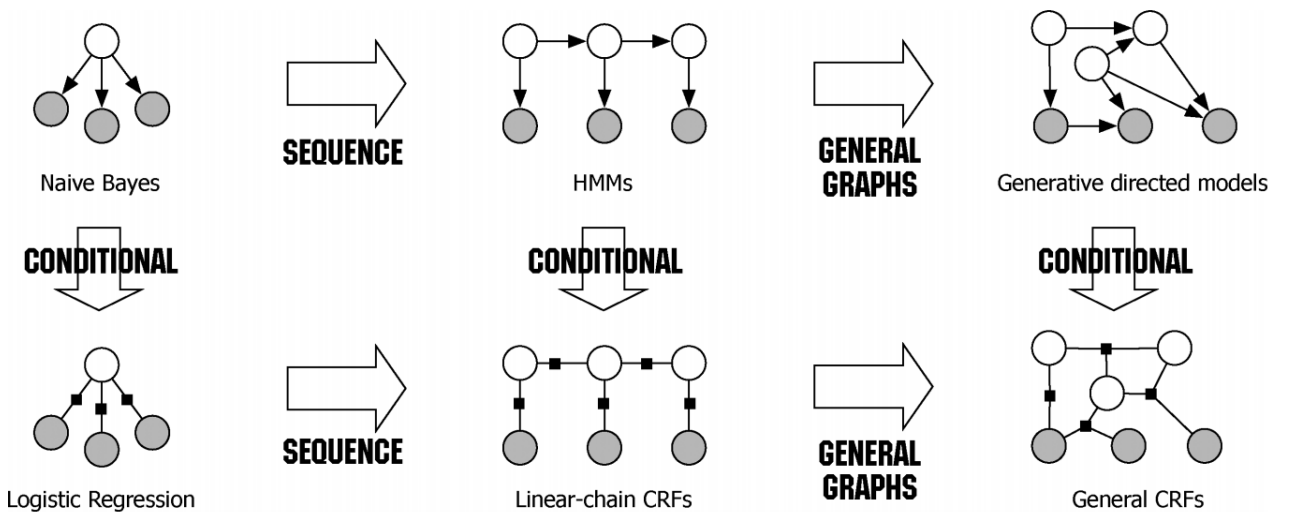


FIGURE 5: Conditional random fields with other models. Taken from (Sutton, 2012)

3.4 Convolutional Neural Network Model (CNN)

3.4.1 Architecture

CNN has become a popular DL model amongst NLP researchers, since the pioneering works of Collobert et al. (2011) and Kim (2014) who advocated the success of CNN in a number of NLP tasks, including sentiment analysis. The main strength of the CNN is its ability to extract the most important n-gram features from the input to create an "informative latent semantic representation" for undertaking further classification tasks (Rojas-Barahona, 2016; Young et al., 2017; L. Zhang et al., 2018).

The basic single layer CNN for sentence modelling may consist of 4 layers as Figure 6 below, according to Kim (2014).

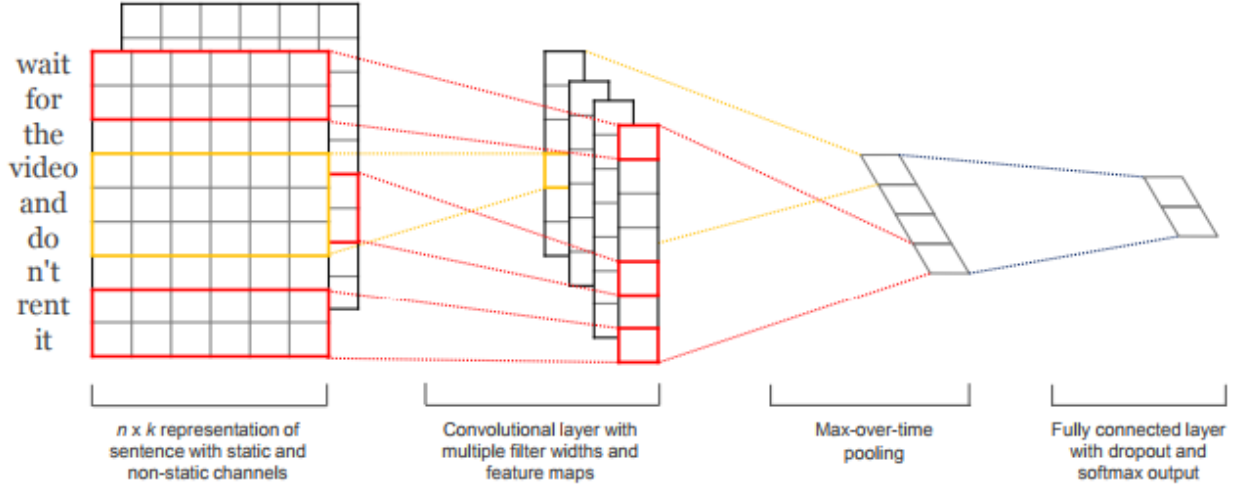


FIGURE 6: Basic CNN model with 4 layers. Adapted from Kim (2014)

The representation of each layer is:

Input layer: representing the sentence of length n as

$$x_{1:n} = x_1 \oplus x_2 \oplus \dots \oplus x_n \quad (21)$$

where $x_i \in \mathbb{R}^d$ be the d -dimensional word vector corresponding to the i -th word in the sentence and \oplus is the concatenation operator.

Convolutional layer: this layer generates the new feature c_i with the filter $w \in \mathbb{R}^{hk}$, using the window of h words from i to $i + h - 1$ as

$$c_i = s(w \cdot x_{i:i+h-1} + b) \quad (22)$$

where $b \in \mathbb{R}$ is the bias term and s is a non-linear activation function, such as *sigmoid*, hyperbolic tangent (*tanh*), or rectified linear (*ReLU*) functions.

So for the sentence, as the possible windows are $\{x_{1:h}; x_{2:h+1} \dots; x_{n-h+1:n}\}$, leading to the feature map as:

$$c = [c_1, c_2, \dots, c_{n-h+1}] \in \mathbb{R}^{n-h+1} \quad (23)$$

Max-pooling layer: this layer selects the maximum value $\hat{c} = \max\{c\}$ as the feature corresponding to one particular filter

Softmax layer: The final feature vector with m filters w is obtained as $\mathbf{z} = [\hat{c}_1, \hat{c}_2, \dots, \hat{c}_m]$, the final output *softmax* layer is obtained using *softmax* function:

$$y_j = \text{softmax}(W\mathbf{z} + b) \quad (24)$$

So the *softmax* layer outputs a probability distribution overall output labels or classes.

The CNN has two important implications from the convolutional and max-pooling layer: first, the convolutional layer is able to capture the most important feature (n-gram) for each feature map; and second, the max-pooling layer can produce a fixed-length output regardless of the size of the filter window h .

3.4.2 Application in consumer review domain

The motivation for utilising the CNN model in ABSA tasks is the assumption that key words may contain the aspect term, and indicate a category or determine polarity, regardless of their position. The CNN is capable of learning to find those features with its architecture (Figure 7) and is, thus, able to extract local patterns from data regardless of their location. This is very useful for identifying fixed-length phrases (Goldberg, 2017). Another advantage is that the CNN is a non-linear model which is expected to better fit the data than linear models such as the CRF and does not require extensive hand-crafted features such as fixed language rules (Poria, Cambria, et al., 2016).

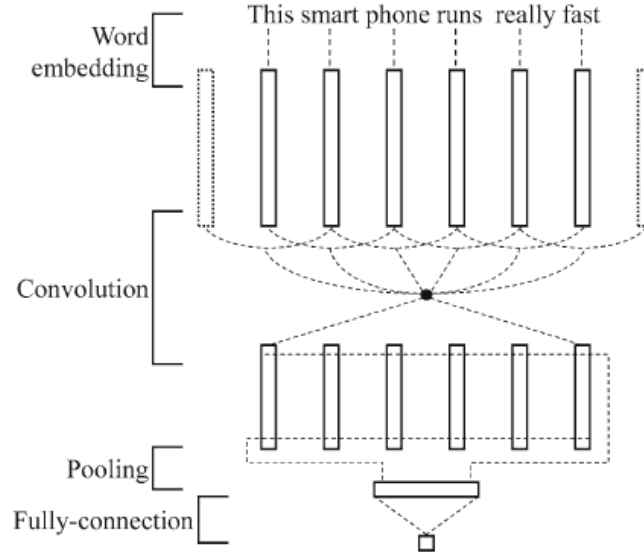


FIGURE 7: An example of CNN architecture for aspect category and sentiment polarity. Adapted from Gu, Gu, & Wu (2017)

The CNN model has been applied to all tasks of ABSA, mainly in *consumer review domain* (Table 4).

TABLE 4: Application of the CNN model in the consumer review domain

No	Study	Domain	Dataset & Language		Model	Performance
Opinion target extraction						
1	Poria, Cambria, et al. (2016)	12 electronic products	Hu and Liu (2004)	English	Deep CNN + Amazon WE + POS + LP	Precision: 82.65 - 92.75% Recall: 85.02 - 88.32% F1: 84.87 - 90.44%
		Laptop	SemEval '14	English		Precision: 86.72% Recall: 78.35% F1: 82.32%
		Restaurant	SemEval '14	English		Precision: 88.27% Recall: 86.10% F1: 87.17%
2	Feng et al. (2018)	Mobile phone	PM from Amazon, Jingdong, and Lynx	Chinese	Deep CNN + WE + POS + dependent syntactic- (explicit aspects)	Precision: 77.75% Recall: 72.61% F1: 75.09%
Aspect category extraction						
3	Toh & Su (2016)	Restaurant	SemEval '16	English	CNN + WE +head word + name list + word cluster	F1: 75.10%
		Laptop	SemEval '16	English		F1: 59.83%
4	Ruder et al. (2016)	Mobile phone	SemEval '16	Dutch	CNN + concatenated vectors	F1: 45.55%
		Hotel	SemEval '16	Arabic		F1: 52.11%
5	Gu et al. (2017)	Smartphone	PM from Amazon	English	Multiple CNNs for each aspect	F1: 72.67 - 83.74%
		Shirt	PM from Taobao	Chinese		F1: 92.26 - 97.34%
6	Wu et al. (2016)	Smartphone	PM from Amazon	English	Multi-task CNN + word2vec/Wikipedia	F1:71.6-81.2%
Sentiment polarity						
7	Gu et al. (2017)	Smartphone	PM from Amazon	English	Single CNN	Acc: 84.87% (binary)
		Shirt	PM from Taobao	Chinese		Acc: 98.26% (binary)
8	Ruder et al. (2016)	Hotel	SemEval '16	Arabic	CNN + aspect tokens	Acc: 82.72%
		Mobile phone	SemEval '16	Dutch	CNN + aspect tokens	Acc: 83.33%
9	Du et al. (2016)	Electronics	PM from Amazon	English	Aspect specific sentiment WE + CNN	Acc: 92.08% (binary)
		Movies and TV		English		Acc: 92.05% (binary)
		CDs and vinyl		English		Acc: 94.38% (binary)
		Clothing, shoes and jewellery		English		Acc: 93.22% (binary)
10	Wu et al. (2016)	Smartphone	PM from Amazon	English	Multi-task CNN+word2vec/Wikipedia	Acc: 84.1% (binary)
11	Xu et al. (2017)	Laptop	PM from Yelp	English	CNN + CRF	Acc: 70.90% (binary, lower than SVM model)
		Restaurant	PM from Yelp	English	CNN + CRF	Acc: 68.34% (binary, lower than SVM model)
12	Akhtar, Kumar, et al. (2016)	12 personal electronic products	PM (Akhtar, Ekbal, & Bhattacharyya, 2016)	Hindi	CNN + SVM	Acc: 65.96% (3-way)

Note: PM indicates that the dataset was primarily collected by authors. 3-way represents the three polarities of positive, negative, neutral.

For the OTE task, a prime example of successful studies is Poria, Cambria, et al. (2016) who adapted CNN architecture from sentence representation to word-based prediction. With the assumption that the tag of each word is dependent on each neighbouring words, they formed a local feature window of five words around each word in a sentence. A deep CNN of seven layers, including one input layer, two convolution layers, two max-pool layers, and a fully connected layer with softmax output, is then applied to each window of words with the prediction to the centre of the window. Experiments show that the deep CNN model, even without any feature engineering or linguistic patterns, still outperformed state-of-the-art models.

For other ABSA tasks, CNN is also a promising approach. Toh and Su (2016) achieved the best performance in SemEval 2016 in ACD with the assembling of two different machine learning systems. As they considered ACD as a multi-class classification problem, they followed a *binary relevance* approach. Particularly, they used multiple binary classifiers trained on a single layer feed-forward neural network then combined the probabilities output from a deep CNN to predict if the text consists of an aspect category. Compared with other features, CNN features contributed the most to performance.

Other papers utilise multiple or multi-tasking CNNs, showing that the CNN can provide other advantages. Examples of multiple CNN can be found in Xu et al. (2017) who incorporated CNN with non-linear CRF to extract the aspect term, then concatenated aspect vector with word embeddings and used another CNN model to determine the sentiment. They achieved a competitive performance in Yelp datasets. Meanwhile, Gu et al. (2017) proposed a cascaded model with two levels of CNNs - CNN aspect mappers and a CNN sentiment classifier. Aspect-mapping CNN and sentiment-classification CNN are organized in a cascaded way. Each mapper determines whether the input sentence belongs to its corresponding aspect. If that is the case, the sentiment classifier predicts sentiment polarity as positive or negative. Apart from the advantages of reduced feature engineering compared to traditional ML methods, the cascaded model also showed that the CNN presented a remarkable reduction of elapsed time, compared to SVM.

Oppositely, an example of multitasking CNN is the work of Ruder et al. (2016) that proposed a CNN approach to undertake both ACD and SP. Similar to Toh and Su (2016), they considered aspect extraction as a multi-label classification problem but approached this through a *probability distribution threshold*. Assuming a sentence S contains K aspect categories, the probability for the sentence to contain an aspect category k is defined as $p(k|S) = \frac{1}{K}$, otherwise, $p(k|S) = 0$. The threshold τ is selected to maximise the F1 score, and the aspect category is selected to satisfy $p(k|S) \geq \tau$. To determine the sentiment towards an aspect, they concatenated an aspect vector with every word embedding and applied a CNN over it. The model also has demonstrated convincing results in the multilingual settings of Spanish, Dutch, and Turkish, showing the strength of DNN as language and domain independence. Another work, Wu et al. (2016), proposed a multitask CNN, which contains aspect mappers and a sentiment classifier sharing word embedding layer whereas other parameters are kept specific in each task. Although this is a promising approach, the experiment showed that multitask CNN performed just slightly better than cascaded CNN.

3.5 Recurrent Neural Network Models (RNN)

Recurrent Neural Networks have become popular in sentiment analysis tasks. The basic of RNN models is that a fixed-size vector represents one sequence (i.e. sentence or document) by feeding each token into a recurrent unit, so it can capture the inherent sequential nature of language (i.e. one word develop its semantic meaning thanks to its previous word) (Goldberg, 2016; Goodfellow et al., 2016). Compared to the CNN models, RNN models have flexible computation steps that the output from RNN is dependent on the previous computations, making it capable of capturing context dependencies in language as well as capable to model various text lengths (Tang, Qin, & Liu, 2016).

3.5.1 Computation of RNN models

The simple RNN model is based on the Elman network (Elman, 1991; Goodfellow et al., 2016) with direct cycles in their hidden connection (Goldberg, 2016; Rojas-Barahona, 2016). This model proposes that the hidden state is dependent on the input and past hidden state, with the same function and the same set of parameters being used at every time step.

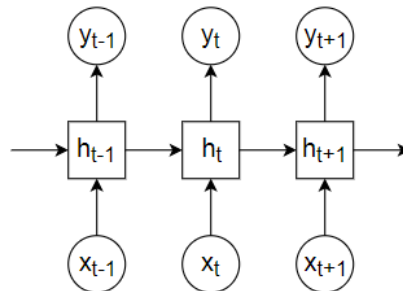


FIGURE 8: Basic RNN model.

Figure 8 shows a basic RNN with a three-layer network of input, hidden state and output. At time t , given

x_t as the input to the network, the hidden state h_t is calculated as;

$$h_t = s(W_{hh}h_{t-1} + W_{xh}x_{t-1}) \quad (25)$$

with W being the weighted matrix function between x_t and h_{t-1} , and s is a non-linear activation function, such as *tanh* or *ReLU*. Therefore, the output can be computed as:

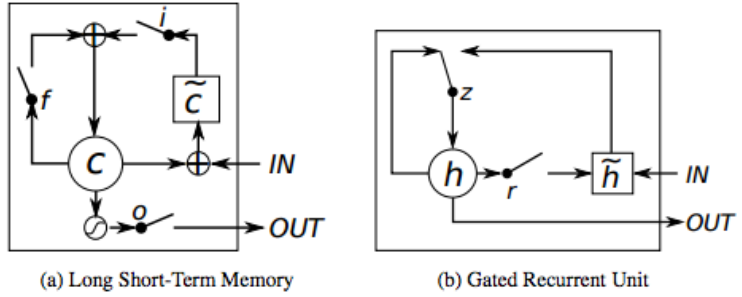
$$y_t = W_{h(y-1)}h_t \quad (26)$$

The RNN model has two important features compared to the feed-forward neural network. First, unlike the CNN has different parameters at each layer, the parameters in RNN are the same in each steps, which then reduces the number of parameters needed to learn (L. Zhang et al., 2018). Second, as the output of one state depends on the previous state, RNN can be said to have the memory of previous computations, making it more superior in processing sequential information compared to the CNN.

However, the simple RNN has a major weakness in terms of the vanishing gradient problems (the gradient comes close to zero) or exploding gradient (the gradient is extremely high) (Goldberg, 2016). As discussed earlier, because the basic role of the gradient is to tune the parameters to improve the gradient, extremes make it difficult to decide in which direction to tweak the parameters, while an exploding gradient causes an unstable learning process (Goodfellow et al., 2016).

However, the simple RNN has limitations caused by the gradient. It may vanish (coming close to zero) or explode (being extremely high). This occurs during the backpropagation process, making it difficult to train and fine-tune the parameters (Goldberg, 2016). This limitation has been improved with the introduction of networks such as long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997; Zaremba & Sutskever, 2014) and gated recurrent units (GRU) (Cho et al., 2014). Table 5 below compares the demonstration and computation of hidden networks between LSTM and GRU. The basis of LSTM is a memory cell that controls the read, write and reset operations of its internal state through output, input and forget gates. At one time t , with the current input x_t and output from the previous state h_{t-1} , the forget gate will decide which information to keep and which to offload, subsequently updating the memory cell. GRU consists of two gates - the reset and the update gate and handles the flow of information, similar to LSTM without the memory unit.

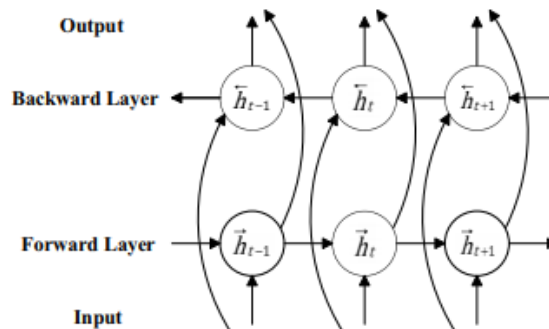
TABLE 5: Comparison of LSTM and GRU. Equations and figures from Chung, Gulcehre, Cho, & Bengio (2014)

	LSTM	GRU
Demonstration	 <p>(a) Long Short-Term Memory (b) Gated Recurrent Unit</p> <p>Figure 1: Illustration of (a) LSTM and (b) gated recurrent units. (a) i, f and o are the input, forget and output gates, respectively. c and \tilde{c} denote the memory cell and the new memory cell content. (b) r and z are the reset and update gates, and h and \tilde{h} are the activation and the candidate activation.</p>	
Computation of hidden state	<p>Input gate: $i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$ (27)</p> <p>Forget gate: $f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$ (28)</p> <p>Output gate: $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$ (29)</p> <p>New memory cell: $\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$ (30)</p> <p>Final memory cell: $c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1}$ (31)</p> <p>Hidden state: $h_t = o_t \odot \tanh(c_t)$ (32)</p>	<p>Update gate: $z_t = \sigma(W_z[x_t] + U_z h_{t-1} + b_z)$ (33)</p> <p>Reset gate: $r_t = \sigma(W_r[x_t] + U_r h_{t-1} + b_r)$ (34)</p> <p>New memory content: $\tilde{h}_t = \tanh(W[x_t] + U(r_t \odot h_{t-1}, x_t) + b)$ (35)</p> <p>Hidden state: $h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$ (36)</p>
Note	<p>x_t as the input to the network at time t</p> <p>h_t is the hidden state at the same time step</p> <p>W, U is the weighted matrix function</p> <p>b is the biases of the LSTM or GRU to be learned during the training</p> <p>\odot denotes element-wise multiplication</p> <p>σ is the logistic sigmoid function</p>	

3.5.2 Bidirectional RNN

The three models presented above focus on using past words to predict the next word. In practice, many studies would like to make predictions based on the future words, and thus, the bidirectional RNN models are proposed, with the incorporation a forward and a backward layer in order to learn information from preceding and following tokens (Graves, 2008; Fan, Qian, Xie, & Soong, 2014). As shown in Figure 7, at each time step t , a hidden forward layer \vec{h} is computed based on the previous hidden state \vec{h}_{t-1} and the current input x_t . Similarly, a hidden backward layer \overleftarrow{h} is computed based on the future hidden state \overleftarrow{h}_{t+1} and current input x_t . The forward and backward context representations \vec{h}_t and \overleftarrow{h}_t are then concatenated into a long vector at the timestep t as:

$$\vec{h}_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (37)$$

**FIGURE 9:** Bidirectional RNN. Taken from (Fan et al., 2014)

3.5.3 Attention mechanism and memory networks

In ABSA, as the aim is to classify sentiment with respect to target aspect terms in the text, it is important for the method to model the interaction between the aspects and the whole sentence. The traditional encoder-decoder framework such as RNN has a potential problem in that the encoder may encode irrelevant information, especially when the input is very information-rich (P. Chen, Sun, Bing, & Yang, 2017; Y. Wang, Huang, Zhao, & Zhu, 2016). One possible solution is to employ an attention mechanism, which allows the model to learn which part of the text to focus on. The general idea of the attention mechanism is to compute an attention weight from each lower level then aggregate the weighted vectors for higher level representation (D. Ma, Li, Zhang, & Wang, 2017). Figure 10 below shows the global attention model on a bidirectional LSTM, following the decoder-encoder model by Bahdanau, Cho and Bengio (2014) in neural machine translation.

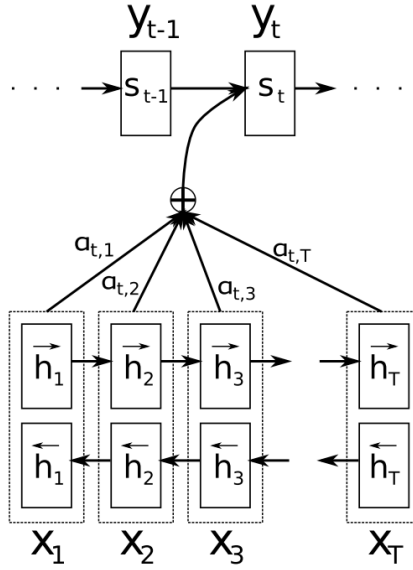


FIGURE 10: Attention mechanism in a bidirectional RNN. Taken from (L. Zhang et al., 2018)

In this model, given the input sentence $S = \{x_1, x_2, \dots, x_T\}$, at time step t , the output y_t is dependent on the decoder state s_t and the set of encoder states $H = \{h_1, h_2, \dots, h_T\}$. The computation of s_t is:

$$s_t = f(s_{t-1}, y_{t-1}, c_t) \quad (38)$$

where c_t is the context vector and c_t is dependent on the set of $H_t = \{h_1, h_2, \dots, h_T\}$. Given the attention weights denoted as $\alpha_t = \{\alpha_{t1}, \alpha_{t2}, \dots, \alpha_{tT}\}$, the context vector is computed as:

$$c_t = \sum_{i=1}^T \alpha_{ti} h_i \quad (39)$$

To compute the attention weight, the model utilises an alignment process, which first computes the attention energies e_{ti} from s_{t-1} and h_i using a feed-forward neural network a as:

$$e_{ti} = a(s_{t-1}, h_i) \quad (40)$$

Variants of attention mechanisms can be computed according to a different function a , such as the additive attention (Bahdanau et al., 2014; Y. Wang et al., 2016) as:

$$e_{ti} = v_a \tanh(W h_i + U s_t) \quad (41)$$

where W, U are weighted matrix and v_a is the weight vector (or aspect embedding vector).

After that, the weight α_{ti} can be computed using the *softmax* function

$$\alpha_{ti} = \text{softmax}(e_{ti}) = \frac{\exp(e_{ti})}{\sum_{j=1}^T \exp(e_{tj})} \quad (42)$$

In the case of ABSA, this also implies that during the decoding period the decoder is conditioned on a "context" vector. This mechanism is most suitable to be applied for the task of sentiment classification, given the aspect terms or aspect categories. It is expected that the models with attention mechanism can focus on the important parts of the sentence in terms of aspects.

Another mechanism that can be applied to resolve the issue of irrelevant information is using external memory such as the Memory Networks model (MemNet) (Sukhbaatar, Szlam, Weston, & Fergus, 2015). Figure 10 shows an extension of the attention mechanism with external memory in a MemNet by Tang, Qin, et al. (2016).

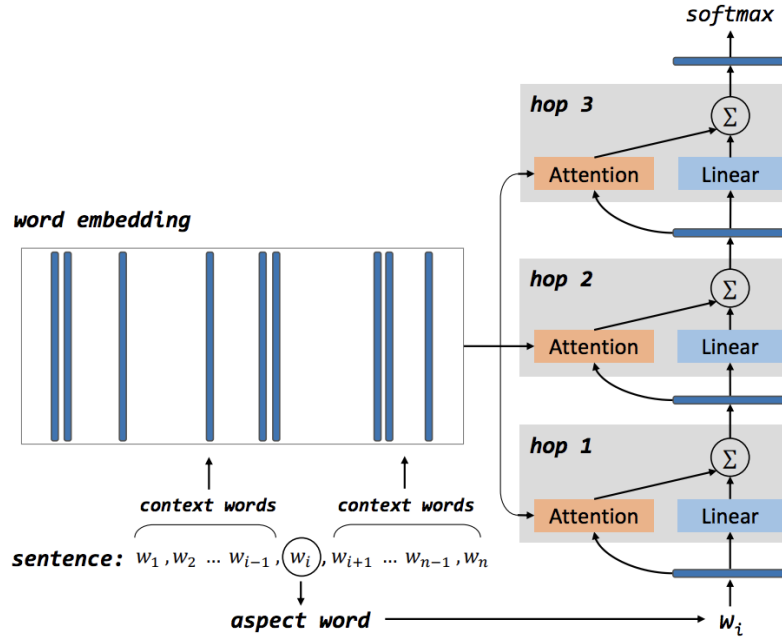


FIGURE 11: A model with three computation layers (hops) model that encodes both context and aspect words, and output of lower hop is recorded as input for higher hop. Taken from Tang, Qin, et al. (2016)

Initially, the sentence is modelled as the composition of n words $\{w_1, w_2, \dots, w_i, \dots, w_n\}$ with the aspect word as w_i . Simultaneously, the context word vectors $\{c_1, c_2, \dots, c_{i-1}, c_{i+1}, \dots, c_n\}$ are stacked as the external memory slices $\{m_1, m_2, \dots, m_{i-1}, m_{i+1}, \dots, m_n\}$. Then, in the first computation layer (hop 1), the aspect vector v_a is selected from the external memory. The output is a continuous vector v computed from the weighted sum of the memory slides as

$$v = \sum_{i=1}^{n-1} \alpha_i m_i \quad (43)$$

where α_i is the weight of m_i to be calculated and $\sum_i \alpha_i = 1$.

The scoring function aims at measuring the semantic similarity of each memory slice m_i to the aspect vector v_a as:

$$e_i = \tanh(W_a[m_i; v_a] + b_a) \quad (44)$$

where W_a is the weighted matrix and b_a is the bias.

This is followed by computation of the weight of m_i using the *softmax* function as:

$$\alpha_i = \text{softmax}(e_i) = \frac{\exp(e_i)}{\sum_{j=1}^{n-1} \exp(e_j)} \quad (45)$$

After the first hop, the attention layer and the linear transformation of the aspect vector are totalled. The sum is stored in the external memory for further information retrieval. This means that the decoder can encode context embeddings from context words and aspect embeddings from aspect words. Thus, the application of attention and memory network mechanisms is useful for sentiment classification of the whole sentence.

3.5.4 Application in the consumer review domain

Compared to other neural network models, RNNs and bidirectional RNNs have had a huge presence in the word-level and sentence-level classification in consumer review domain (Table 6).

TABLE 6: Application of RNN model in the consumer review domain

No	Study	Domain	Dataset & Language		Model	Performance
Opinion target extraction						
1	Toh & Su (2016)	Restaurant	SemEval '16	English	RNN + WE + Name List + DP Name List + Word Cluster	Precision: 75.49% Recall: 69.44% F1: 72.34%
2	T. Chen et al. (2017)	Restaurant	SemEval '16	English	BiLSTM + Google WE + CRF	F1: 72.44%
		Restaurant	SemEval '16	Spanish		F1: 71.70%
		Restaurant	SemEval '16	French		F1: 73.50%
		Restaurant	SemEval '16	Russian		F1: 67.08%
		Restaurant	SemEval '16	Dutch		F1: 64.29 %
		Restaurant	SemEval '16	Turkish		F1: 63.76%
3	Liu et al. (2015)	Laptop	SemEval '14	English	LSTM-RNN+ POS + chunk + Amazon WE	F1: 75.00%
		Restaurant	SemEval '14	English	Bi-Elman-RNN + POS + chunk + Amazon WE	F1: 82.06%
4	Jebbara & Cimiano (2016)	Restaurant Laptop Hotel	ESWC Challenge 2016	English	BiGRU + Amazon WE + POS	Precision: 65.9% Recall: 71.0% F1: 68.4%
5	Tay, Tuan and Hui (2017)	Restaurant	SemEval '14	English	Holo DyMemNN	Precision: 81.87% Recall: 79.73% F1: 79.73%
		Laptop	SemEval '14	English	Holo DyMemNN	Precision: 75.16% Recall: 73.19% F1: 74.03%
6	Al-Smadi, Qawasmeh, Al-Ayyoub, Jararweh, and Gupta (2017)	Hotel	SemEval '16	Arabic	RNN	F1: 48%
					SVM + morphological, N-grams, syntactic, and semantic	F1: 89.8%
7	Yuan, Zhao, Qin, and Liu (2017)	Restaurant	SemEval '14	English	LSTM + Local Context + Senna WE	F1: 80.62% (lower than CRF models)
		Laptop	SemEval '14	English	BiLSTM + Local Context + Senna WE	F1: 74.78% (lower than CRF models)
8	X. Wang et al. (2016)	Restaurant	SemEval '14	English	Uni-directional Elman RNN	F1: 82.12% (lower than the CRF model)
		Laptop	SemEval '14	English		F1: 75.45%
9	Ding, Yu, and Jiang (2017b)	Restaurant	SemEval '14 + '15	English	Hierarchical LSTM	F1: 77.9%
		Laptop	SemEval '15	English		F1: 76.6%
		Digital device	(Kessler et al., 2010)	English		F1: 45.1%
		Web service	(Toprak et al., 2010)	English		F1: 43.8%
10	W. Wang, Pan, & Dahlmeier (2017)	Restaurant	SemEval '14	English	Coupled Multi-layer Attentions (CMLA) based on GRU	F1: 85.29%
		Laptop	SemEval '14	English		F1: 77.80% (lower than RNCRF)
		Restaurant	SemEval '15	English		F1: 70.73%
11	Li & Lam (2017)	Laptop	SemEval '14	English	Memory Interaction Network (MIN) based on LSTM with extended memory	F1: 77.58%
		Restaurant	SemEval '16	English		F1: 73.44%
12	Li, Bing, Li, Lam, & Yang (2018)	Laptop	SemEval '14	English	Truncated History-Attention (THA) and Selective Transformation Network (STN) built on two LSTMs	F1: 79.52%
		Restaurant	SemEval '14	English		F1: 85.61%
		Restaurant	SemEval '15	English		F1: 71.46%
		Restaurant	SemEval '16	English		F1: 73.61%
13	Mai & Le (2018)	Mobile phone review	PM from Youtube	Vietnamese	Bidirectional RNN + CRF	Precision: 68.12% Recall: 75.87% F1: 71.79%
Aspect category extraction						
14	Tamchyna & Veselovská (2016)	Restaurant	SemEval '16	Turkish	Binary classifier (deep LSTM) for each category	F1: 61.03%
		Restaurant	SemEval '16	Russian		F1: 64.83%
15	Tay et al. (2017)	Restaurant	SemEval '14	English	Tensor DyMemNN	F1 (binary): 81.68%
		Reviews	Merge	English	Tensor DyMemNN	F1 (binary): 81.66%

No	Study	Domain	Dataset & Language		Model	Performance
			dataset of SemEval '14-15			
16	Ding, Yu, & Jiang (2017a)	Restaurant	Yelp SG dataset	English	RNN + finetune WE	F1: 72.42%
		Laptop	Amazon Product Reviews (H. Wang, Lu, & Zhai, 2011)	English		F1: 66.17%
17	Chaudhuri & Ghosh (2016)	Hotel	DBS Text Mining Challenge 2015 data	English	Weighted Hierachi Bidirectional RNN (mini-batches)	65% (10 aspects); 55% (20 aspects)
18	Y. Ma et al. (2018)	Reviews	SemEval'15 modified (no explicit aspects)	English	LSTM + Target attention + Sentence-level attention + SenticNet	Macro F1: 76.44% Micro F1: 73.82%
19	Ding et al. (2017b)	Restaurant	SemEval '14 + '15	English	Hierarchical LSTM	F1: 77.9%
		Laptop	SemEval '15	English		F1: 76.6%
		Digital device	JPDA Corpus (Kessler et al., 2010)	English		F1: 45.1%
		Web service	Darmstadt Corpus (Toprak et al., 2010)	English		F1: 43.8%
20	W. Wang et al. (2017)	Restaurant	SemEval '14	English	Coupled Multi-layer Attentions (CMLA) based on GRU	F1: 85.29%
		Laptop	SemEval '14	English		F1: 77.80% (lower than RNCRF)
		Restaurant	SemEval '15	English		F1: 70.73%
21	Li & Lam (2017)	Laptop	SemEval '14	English	Memory Interaction Network (MIN) based on LSTM with extended memory	F1: 77.58%
		Restaurant	SemEval '16	English		F1: 73.44%
Sentiment polarity						
22	Tay, Luu, & Hui (2017)	Restaurant	SemEval '14	English	Aspect Fusion LSTM	Acc: 75.44 (3 way*); 87.78 (binary)
		Laptop	SemEval '14	English		Acc: 68.81 (3 way); 83.58 (binary)
23	Cheng et al. (2017)	Restaurant	SemEval '14	English	BiGRU + aspect attention + sentiment attention	Acc: 91.3 (binary); 85.1 (3-way)
		Restaurant	SemEval '15	English		Acc: 83.4 (binary); 80.5 (3-way)
		Restaurant	SemEval '16	English		Acc: 91.1 (binary); 87.5 (3-way)
		Laptop	SemEval '15	English		Acc: 88.0 (binary); 85.1 (3-way)
24	Y. Wang et al. (2016)	Restaurant	SemEval '16	English	LSTM + aspect attention + aspect embeddings	Acc: 77.2 (3 way); 90.9 (binary)
		Laptop	SemEval '15	English		Acc: 68.9 (3 way); 87.6 (binary)
25	Y. Ma et al. (2018)	Restaurant	SemEval'15 ABSA modified (no implicit target)	English	Sentic LSTM + Target attention + Sentence-level attention	Acc: 76.47% (binary)
26	P. Chen et al. (2017)	Restaurant	SemEval '14	English	Recurrent Attention on Memory (RAM) + attention layers	Acc: 80.59% (3-way)
		Laptop	SemEval '14	English		Acc: 74.65% (3-way)
27	R. Ma et al. (2017)	Restaurant	SemEval '14	English	Feature-based Compositing Memory Networks	Acc: 82.03% (3-way)
		Laptop	SemEval '14	English		Acc: 73.94% (3-way)
28	Tang, Qin, et al. (2016)	Restaurant	SemEval '14	English	Deep memory network	Acc: 80.95% (3-way)
		Laptop	SemEval '15	English		Acc: 72.37% (3-way)
29	Peng et al. (2018)	Notebook	Chinese aspect	Chinese	Aspect target sequence model (ATSM), working	Acc: 75.59% (binary); F1: 60.09%

No	Study	Domain	Dataset & Language	Model	Performance
		Car	dataset	on word level, character and radical level.	Acc: 82.94% (binary); F1: 64.18%
		Camera			Acc: 84.86% (binary); F1: 75.35%
		Phone			Acc: 85.95% (binary); F1: 80.13%

Note: PM indicates that the dataset was primarily collected by the authors. 3-way refers to the three polarities of positive, negative, neutral.

Many RNN-based models took advantage of the bidirectional RNN to recode past and future contexts. An approach from Chaudhuri & Ghosh (2016) used hierarchical bidirectional RNN in ACD in highly skewed data of hotel review domain and obtained superior result over LSTM. The model is composed of six layers including four layers of bidirectional RNNs, one fully connected layer and one softmax layer. Each layer constitutes a hierarchy of classifier. They proposed a mini-batch approach whereby the input dataset is separated randomly into a few words to compute seed classifications; then the remaining words are placed into seed classes to find the highest similarity on average. With the similar task of ACD, Ding et al. (2017b) also used an RNN to model the context of each word as well as the background context. Using continuous vectors to calculate the probabilities of generating different words, they offered an alternative solution for topic models, which was more effective. Jebbara and Cimiano (2016) employed bidirectional GRU for OTE and aspect-specific sentiment extraction. As a first step, a bidirectional GRU is used to extract aspects from a text as a sequence labelling of IOB. In a second step, a bidirectional GRU extracted aspect regarding its context and predicted its sentiment label. Other features include pre-trained semantic word embedding, semantic knowledge extracted from Word-Net and SenticNet.

One of the most successful attempts is to combine RNN with the CRF classification layer, so that the model not only captures the long-term dependency of the entire sentences, but also utilises the dependency of each label on each other. Liu et al. (2015) proposed an application of recurrent neural network (RNN) in OTE with linguistic features of POS, word chunks, which showed better performance than a feature-rich CRF-based system. Inspired by the system of NER by Lample et al. (2016), T. Chen et al. (2017) proposed a bidirectional LSTM-CRF in classifying numbers of targets in the sentence, but the model also achieved state-of-the-art performance in OTE. Overcoming the limitation of a fixed window size in CNN model, this network captured long-term dependencies of context information. The result of the bidirectional LSTM is two fixed-size vectors, which were then concatenated at the fully connected layer. For the IOB tagging, the authors use a CRF layer at last (Figure 12). A similar model by Mai and Le (2018) also showed its effectiveness in Vietnamese.

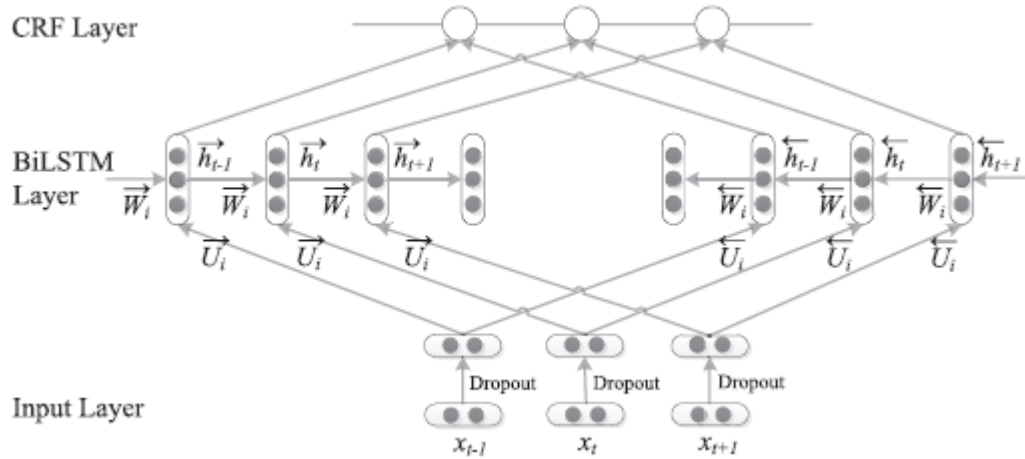


FIGURE 12: Bidirectional LSTM-CRF in opinion target extraction. Taken from T. Chen et al. (2017)

Another promising direction is to utilise attention and memory networks. Approaching the OTE task, Li & Lam (2017) proposed an extended memory framework for LSTM while Ying et al. (2017) proposed a LSTM model in cross-domain aspect term extraction, with the combination of rule-based methods that generated auxiliary label sequence for each sentence. Another study by Li et al. (2018) also incorporated attention in tasks of OTE and ACD with their Truncated History-Attention (THA) and Selective Transformation Network (STN) built on two LSTMs.

Meanwhile, W. Wang et al. (2017) proposed a Coupled Multi-Layer Attention Model (CMLA) based on GRU for co-extracting of aspect and opinion terms. Therefore, learning can be done by encoding/decoding the dual propagations of aspects and opinion terms, and not restricted to grammatical relations (Figure 11). This framework reduces engineering features compared to the CRF and the co-extraction is a worth-noting feature.

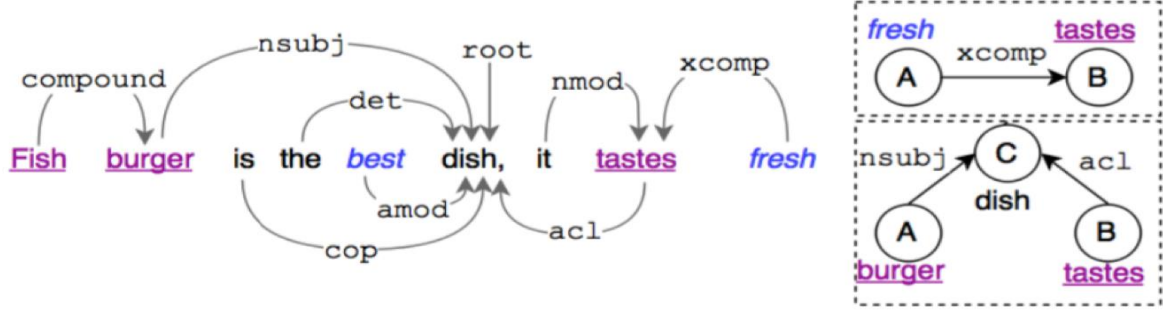


FIGURE 13: Grammatical relations determine aspects and opinions: ‘fish burger’ and ‘tastes’ are obvious aspect terms, with the respective opinions of ‘best’ and ‘fresh’. Considering tastes as an aspect term, fresh can be extracted as an opinion term through a direct relation. Considering ‘fish burger’ as an aspect term, ‘tastes’ can be extracted as another aspect term through the indirection relation. Taken from W. Wang et al. (2017)

Y. Wang et al. (2016) and Y. Ma et al. (2018) proposed a solution with attention weight, in which aspect embeddings are used to decide attention weights for sentiment classification, in addition with sentence representation. Therefore, the model can have different concentration on different parts when different aspects are given (for example in Figure 12). Another work by Cheng et al. (2017) applied attention with bidirectional GRU model to attend the aspect information for one given aspect and extract sentiment for that given aspect. Their work achieved state-of-the-art performance on benchmark datasets. Also in a similar task, Peng et al. (2018) proposed an aspect target sequence model (ATSM) to incorporate adaptive embeddings at word, character and radical level in dealing with multiple-word aspect issues in Chinese.

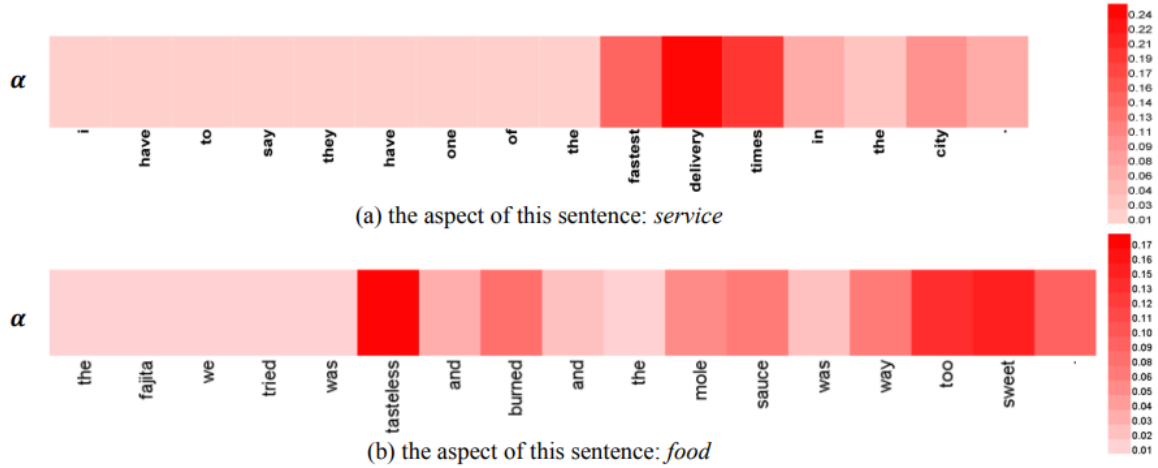


FIGURE 14: Focus of attention on different aspects. Taken from Y. Wang et al. (2016)

On the other hand, Tang, Qin, et al. (2016) adopted a memory network (MemNet) solution, which is based on multiple-hop attention. They included a multiple-attention computation layer on the memory network, which improved lookup for most informational regions. Memory networks also feature in R. Ma et al. (2017) who uses compositing strategies to represent context and features for each word. In deep hops, their model outperforms state-of-the-art approaches of SVM with less feature engineering. Tay, Tuan, et al. (2017) proposed Dyadic Memory Networks (DyMemNN), which incorporates composition techniques that model the dyadic interactions between aspect and words in a document. Their model also achieved competitive performance in OTE and ACD.

3.5.5 Application in targeted sentiment analysis

RNN models are also applied in Twitter domain and new comments (Table 7) rather than CNN model, as explained above on the limitation of CNN model in capturing long-term dependencies. Unlike the customer review domain, the Twitter domain is challenged with the limited length, informal contexts and the use of emoticons (Giachanou & Crestani, 2016). However, studies using RNN models have showed competitive performance in this task.

TABLE 7: Application of the RNN model in targeted sentiment analysis

No	Study	Domain	Datasets	Language	Model	Performance
Opinion category detection						
1	Tay, Tuan, et al. (2017)	Tweets	SemEval 2016 Tweet task	English	Tensor DyMemNN	F1: 72.42% Precision: 72.11% Recall: 72.79%
		Debates	Internet Argument Corpus v2	English	Tensor DyMemNN	F1: 66.17% Precision: 66.53% Recall: 66.07%
2	Y. Ma et al. (2018)	London locations	SentiHood	English	LSTM + Target attention + Sentence-level attention + SenticNet	Acc: 67.43% Macro F1: 78.18% Micro F1: 77.66%
Sentiment polarity						
3	Tang, Qin, Feng, & Liu (2015)	Twitter data	Dong et al. (2014)	English	Target Connection LSTM	Acc: 71.5% Macro F1: 69.5%
4	M. Zhang, Zhang, & Vo (2016)	Twitter data	Dong et al. (2014)	English	Bidirectional GRNN & 3-way gate	Acc: 71.96 (3-way)
5	Y. Ma et al. (2018)	London locations	SentiHood	English	Sentic LSTM + Target attention + Sentence-level attention	Acc: 89.32% (binary)
6	P. Chen et al. (2017)	Twitter data	Dong et al. (2014)	English	Recurrent Attention on Memory (RAM) + attention layers	Acc: 69.36 (3-way)
		Chinese news comments	Primarily collected data	Chinese		Acc: 73.89 (3-way)

An interesting work by Tang, Qin, Feng, et al. (2015) proposed adding attention layers in their bidirectional LSTM. They proposed two models to achieve target-specific sentiment classification: Target Dependent LSTM (TDLSTM) directly uses the hidden outputs of a bidirectional LSTM sentence encoder in panning the target mentions, while Target Connection LSTM (TCLSTM) extends TDLSTM by concatenating each input word vector with a target vector (Figure 13). However, they failed to achieve competitive results possibly due to the small training corpus.

Further works by P. Chen et al. (2017) and Tay, Tuan, et al. (2017) also focused on attention mechanisms for the LSTM to incorporate aspect information into the model. While P. Chen et al. (2017) adopted a multiple-attention mechanism, Tay, Tuan, et al. (2017) introduced a novel association layer with holographic reduced representation.

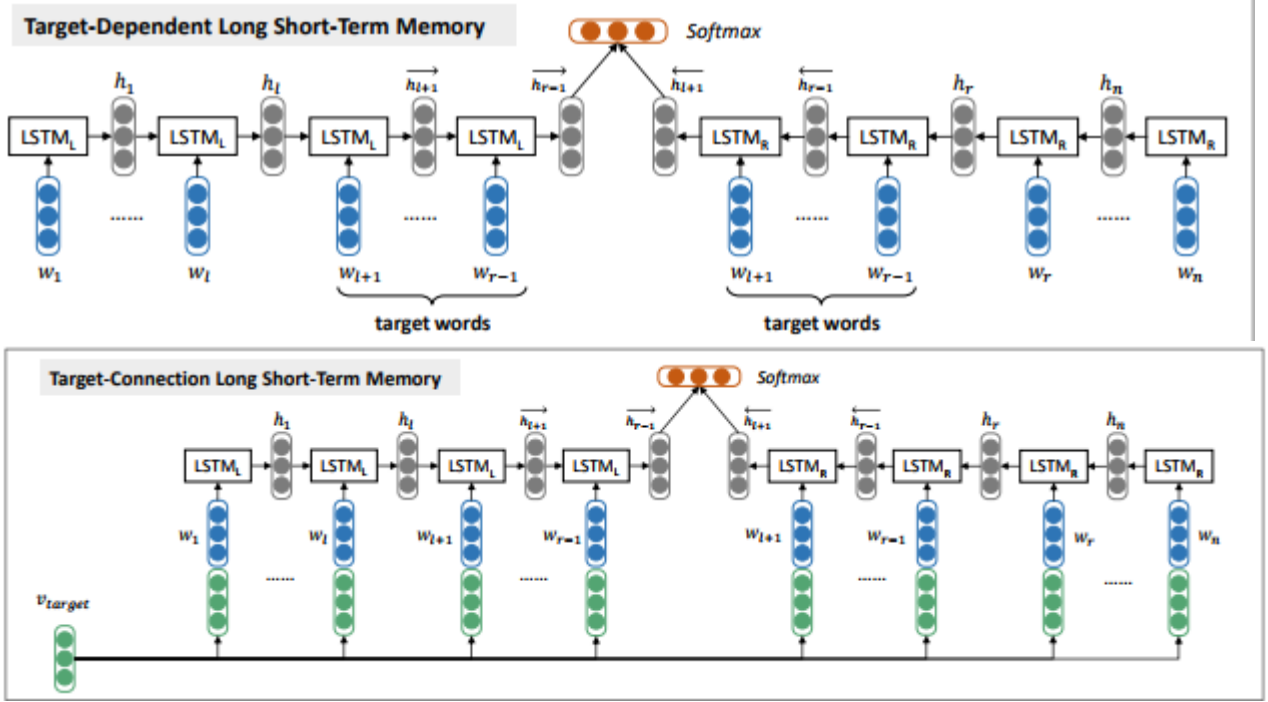


FIGURE 15: Target-Dependent LSTM & Target-Connection LSTM. Source: Tang, Qin, Feng, et al. (2015)

3.6 Recursive Neural Network Model (RecNN)

3.6.1 Architecture

Recursive neural network (RecNN) models are linguistically motivated in that they explore tree structures (e.g., syntactic structures) and aim to learn elegantly compositional semantics. Arguably, natural language demonstrates a natural recursive structure, placing words and phrases in a hierarchical manner. Thus, tree-structured models can better make use of such syntactic interpretations of sentence structure (Socher, Perelygin, & Wu, 2013; Tai, Socher, & Manning, 2015). Generally, in a recursive neural network, the vector representation of each node in the tree structure is calculated from the representation of all its children using a weight matrix W which is shared across the whole network (Socher et al., 2013). For example, giving c_1 and c_2 as n -dimensional vector representation of nodes, their parent will also be an n -dimensional vector calculated using a non-linear function such as \tanh :

$$p_{1,2} = \tanh(W[c_1; c_2]) \quad (46)$$

So in general, a hidden vector for any node n associated with a word vector x_n can be computed as:

$$h_n = \tanh\left(W_v \cdot x_n + b + \sum_{k \in \mathcal{K}_t} W_{r_{nk}} h_k\right) \quad (47)$$

where \mathcal{K}_n denotes the set of children of node n , r_{nk} denotes the dependency relation between node n and its child node k , and h_k is the hidden vector of the child node k .

The tree structures used for RNNs include constituency tree and dependency tree. In a constituency tree, the words is represented at leaf nodes, a phrase is represented at internal nodes the root node represents the whole sentence (Socher et al., 2013). Meanwhile, in a dependency tree, each node including represents a word, connecting with other nodes with dependency connections (Socher et al., 2013). Demonstration of the constituency tree and dependency tree is presented in Figure 16.

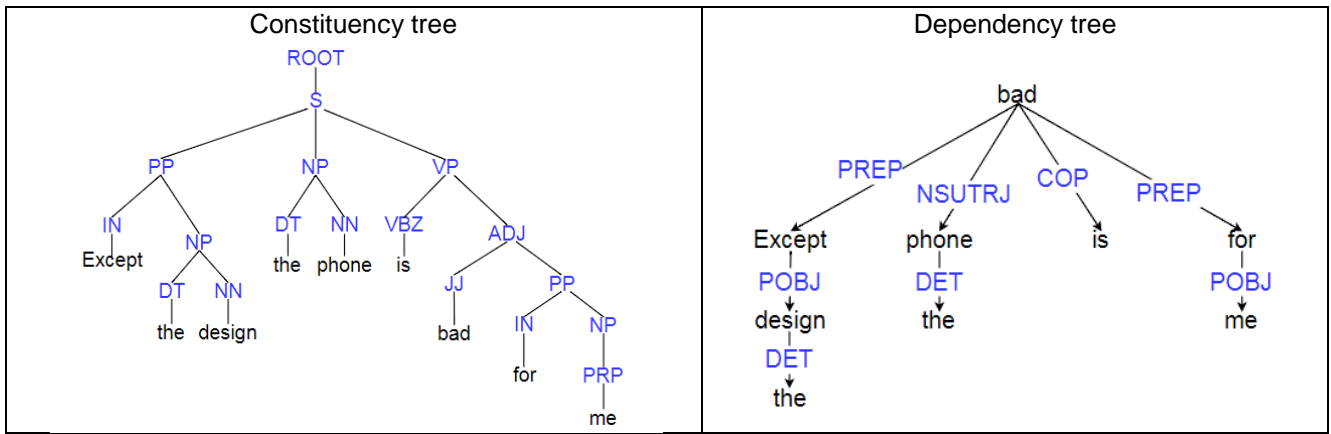


FIGURE 16: Constituency tree and dependency tree. Adapted from Nguyen & Shirai (2015)

3.6.2 Application

Despite the popularity of Recursive Neural Networks (RecNN) in various NLP tasks, its application to ABSA is rather limited (Table 8).

TABLE 8: Application of RecNN in opinion target extraction in customer reviews and targeted Twitter sentiment analysis

No	Study	Domain	Dataset & Language		Model	Performance
Opinion target extraction						
1	Nguyen & Shirai (2015)	Restaurant	SemEval '14	English	PhaseRecNN + CRF + multiple global functions	Precision: 62.40% Recall: 63.94% F1: 62.21%
2	W. Wang, Pan, Dahlmeier, & Xiao (2016)	Laptop	SemEval '14	English	RecNN + CRF + POS + Name list + Sentiment Lexicon	F1: 78.42%
		Restaurant	SemEval '14	English	RecNN + CRF + POS + Name list + Sentiment Lexicon	F1: 84.93% (lower than CRF models)
3	Lakkaraju, Socher, & Manning (2014)	Beer	Stanford Beer Advocate Dataset	English	Joint Multi-Aspect Sentiment Model + Recursive Neural Tensor Network	Accuracy for aspect terms: 77.04%
		Camera	Amazon	English	Joint Multi-Aspect Sentiment Model + Recursive Neural Tensor Network	Accuracy for aspect terms: 81.02%
Sentiment polarity						
4	Dong et al. (2014)	Twitter data	PM	English	Adaptive Recursive Neural Network + uni/bigram features + SVM classifier	Acc: 66.3 (3 way)
5	Vo & Zhang (2015)	Twitter data	Dong et al. (2014)	English	Dependency parser & sentiment lexicon	F1: 69.9% (3-way) Acc: 71.1% (3-way)
6	Zainuddin et al. (2018)	Twitter data	Hate Crime Twitter Sentiment (HCTS)	English	Association rule mining (ARM) + POS + Stanford Dependency Parser (SDP) + Sentiwordnet+PCA + SVM	Acc: 71.62%
			Stanford Twitter Sentiment (STS) dataset	English		Acc: 76.55%
			Sanders Twitter Corpus (STC)	English		Acc: 74.24%
7	B. Wang, Liakata, Zubiaga and Procter (2017)	Twitter data	Dong et al. (2014)	English	TDParse+ (dependency parser and the three sentiment lexica)	F1: 69.8% (3-way) Acc: 72.1% (3-way)
		Tweets from 2015 UK general election campaign	PM	English	TDParse (dependency parser)	Acc: 56.45% F1: 46.09%

Regarding to consumer domain review, a study by Lakkaraju et al. (2014) proposed recursive tensor neural networks in extracting both target and sentiment that is more robust than two single models, and also allows for the representation of multiple aspects within the text.

Two more recent works involving RecNN in the customer review domain include those of Nguyen & Shirai (2015) and W. Wang et al. (2016), both aimed at exploiting the aspects through the dependency and constituent trees of the sentence. While Nguyen & Shirai (2015) just focus on OTE using dependency and constituent trees, W. Wang et al. (2016) expanded the recursive neural network models by Lakkaraju et al. (2014) with a novel framework of RecNN & CRF to co-extract the aspect and opinion terms. This framework consists of a dependency-tree RecNN sentence representation, which feeds input to the CRF for target and opinion co-extraction.

RecNN models developed by Lakkaraju et al. (2014) and Wang et al. (2016) aimed to improve the error-prone two-step approaches (Akhtar et al., 2017; Gu et al., 2017; T. Chen et al., 2017), whereby error 1 leads to error 2. However, while recursive neural network frameworks can resolve this issue, they have not overcome the limitation of RecNN caused by the requirement for a pre-defined tree structure to encode sentences, which limits the scope of its application (Rojas-Barahona, 2016).

Approaching the task of target-dependent Twitter sentiment analysis, Dong et al. (2014) proposed Adaptive Recursive Neural Network that propagates the sentiments of words to target depending on the context and syntactic relationship. This work can be considered as similar to sentiment polarity of aspect term as their annotated dataset contains only one target per tweet.

Nevertheless, contrary to previous studies, B. Wang et al. (2017) argued that simply averaging the attention vector in RNN models might not solve the issues of multiple targets within a text. Their proposed model of using dependency trees overcomes this issue and achieves competitive performance in Twitter target sentiment analysis.

3.7 Hybrid Models

Coping with both advantages and disadvantages of the previously discussed models, many studies attempted to apply hybrid solutions in customer review domains, such as Xue, Zhou, Li, & Wang (2017), Ye et al. (2017), P. Chen, Xu, Yang, & Li (2016)

Xue et al. (2017) noted that the aspect terms and aspect category are closely related, so they proposed a multi-task framework of BiLSTM for OTE and CNN for ACD. The main benefits of this framework is the mutual information sharing of two tasks, in which the CNN can also utilize extra information learned in the BiLSTM to improve its informative features, while the predicted tag from the BiLSTM can also receive the most salient n-gram features via convolutional operations.

Similarly, P. Chen et al. (2016) also combined LSTM and CNN together for sentiment classification but used LST for generating context embedding and CNN for detecting features. Ye et al. (2017) proposed a dependency-tree based convolutional stacked neural network (DTBCSNN) for aspect term extraction, in which the convolution is included in the sentence's dependency parse trees to capture syntactic and semantic features. This can overcome the practical limitations of sequential models (RNNs) which cannot capture the tree-based dependency information. The proposed model does not need any handcraft features and flexible to include extra linguistic patterns. Table 9 below provides an overview of hybrid solutions.

TABLE 9: Application of Hybrid Models

No	Study	Domain	Dataset & Language		Model	Performance
Opinion target extraction						
1	Ye et al. (2017)	Restaurant	SemEval '14	English	Dependency-Tree Based CNN + POS + chunk	F1: 83.97% (lower than the CRF winning model)
		Laptop	SemEval '14	English		F1: 75.66%
2	Xue et al. (2017)	Restaurant	SemEval '14	English	Multi-task learning neural network combines BiLSTM and CNN layers	F1: 83.65% (lower than the top models)
			SemEval '15	English		F1: 67.73%
			SemEval '16	English		F1: 72.95%
Aspect category detection						
3	Xue et al. (2017)	Restaurant	SemEval '14	English	Multi-task learning neural network combines BiLSTM and CNN layers	F1: 88.91%
			SemEval '15	English		F1: 65.97%
			SemEval '16	English		F1: 76.42%
Sentiment polarity						
4	P. Chen et al. (2016)	Phone	COAE2014 task5	Chinese	LSTM + CNN	Acc: 90.91% (3-way polarity)
		Car	COAE2012 task1 and the autohome website	Chinese		Acc: 78.62% (3-way polarity)

4 COMPARISON OF PERFORMANCE ON BENCHMARK DATASETS

The above discussion has provided insights into the different approaches chosen by researchers for NLP tasks. It is evident that outcomes depend not only on model choice but on the semantics to be analysed. Yin et al. (2017) evaluated sentiment classification, question answering and POS tagging and concluded that model choice depended on the global semantics of a classification task and that, therefore, the focus should be on hyper-parameters such as layer size.

This observation is validated by a comparison of models. It is clear that CNN models can be highly effective (Kim, 2014; Poria, Cambria, et al., 2016; Wu et al., 2016) due to their ability to extract local patterns (i.e. the most important n-gram) of a sentence to produce fixed size input. However, this is true only as long as classification of key phrases of limited length is required (Goldberg, 2017). Furthermore, CNN models demand large sets of training data and require a significant amount of fine-tuned parameters (Yin, Kann, Yu, & Schütze, 2017; Young et al., 2017). Further issues arise from the fixed size of the hidden layer which prompts manipulation of input sentence length (CNN models add padding to short sentences and reduce long sentences), making capture of broader contextual information and sentence dependencies impossible (Zhao et al., 2017). Although this limitation can to some extent be overcome by a text window approach, whereby local feature windows of neighbouring words form around each word such as demonstrated by Poria, Cambria, et al. (2016), no information can be captured outside the window. As pointed out by Tu, Lu, Liu, Liu, & Li (2016), this has important implications for the application of CNN to languages with morphologically-rich texts such as Russian and Mandarin. In such cases, a model capable of recognising long-term dependencies such as Recurrent Neural Networks (RNN) or Recursive Neural Networks (RecNN) is called for.

RNNs are powerful because they combine two properties: (i) Distributed hidden states that allows them to efficiently store information from past computations; and (ii) Non-linear dynamics that better fit the non-linear nature of data (Tang, Qin, Feng, et al., 2015; Y. Wang et al., 2016). Significant research suggests that RNN is superior to CNN, citing the example of the LSTM model which does not require large training datasets (Plank, Søgaard, & Goldberg, 2016) and can achieve comparative performance to CNNs with fewer parameters (Hassan & Mahmood, 2017). Therefore, in terms of ABSA tasks, RNNs may perform better than CNNs if the classification is dependent on the semantic relationship of whole sentences.

In the case of RecNNs, a simple architecture and the ability to learn tree-structures of sentences and new words are distinct advantages (Socher et al., 2013; Tai et al., 2015). However, they are heavily dependent on parsers (Socher et al., 2013) and have not yet shown consistent performance in sentence classification (Goldberg, 2017). Further research is clearly required.

Thus, these different models were designed with different objectives in terms of sentence modelling, particularly when analysing CNNs and RNNs. While CNNs try to extract the most important n-grams, RNNs try to create a composition with unbounded context (Young et al., 2017; L. Zhang et al., 2018).

Table 10 presents the summary of model comparison.

TABLE 10: Summary of model comparison

	CNN	RNN	RecNN
Advantages	<ul style="list-style-type: none"> • Ability to extract meaningful local patterns (n-grams) • Non-linear dynamics • Fast computation 	<ul style="list-style-type: none"> • Distributed hidden state that can store past computations • Ability to produce a fixed size vector that takes into account the weighted combination of all words and summarizes the sequence • Do not require large dataset • Require fewer parameters 	<ul style="list-style-type: none"> • Simpler architecture • Ability to learn tree-like structures • Ability to construct representations for any new word
Disadvantages	<ul style="list-style-type: none"> • High demand for data • Fixed size of hidden layer • Failure to capture long-term dependencies 	<ul style="list-style-type: none"> • Chooses the last hidden state to represent the sentence which may lead to incorrect prediction 	<ul style="list-style-type: none"> • Requires parsers/parameters which can be slow and lead to inaccuracies • Models are still in their infancy
Implication for ABSA	<ul style="list-style-type: none"> • Useful if the sentence is supposed to have one opinion/target of fixed length • Not effective for parsing longer sentences 	<ul style="list-style-type: none"> • Useful to capture the semantic meaning • Not useful if the sentence is represented by a key phrase 	<ul style="list-style-type: none"> • The application and training regime still require further research

To provide insights into the large number of proposed methods for ABSA, the below session will classify all methods according to three ABSA tasks: aspect term (or opinionated target) extraction (OTE), methods focusing on aspect category detection (ACD), and methods focusing on aspect-specific sentiment polarity (SP).

For each task, the comparison is presented with a table outlining the attempted DNN methods together with the best-performing methods from SemEval ABSA. Each table contains the method, its domain, the performance as reported by the studies. The performance is reported in the form of Precision, Recall, F1, and Accuracy. It should be noted that: (i) some papers did not provide all the measures; (ii) when multiple models are proposed, the best model will be reported; (iii) due to the difference in experimental settings, the methods should not be compared using the scores; (iv) some researchers aimed to resolve two or three tasks and so will appear more than once in the tables.

4.1 Opinion target extraction

As the majority of studies in this task used data from SemEval 2014 with two domains in English, namely laptop and restaurant, Table 10 evaluates performance for a range of models with respect to this dataset. It is apparent that the majority of approaches were implemented according to RNN and its variants as LSTM or GRU, with high performance in both domains - F1 of over 75 in the laptop domain and over 80 in the restaurant domain. The current best model appears to be the CNN system by Poria, Cambria, et al. (2016) for both domains, showing that a window-approach in CNN can extract relevant opinion targets, and can overcome the issue of long-term dependency. It is also interesting that the attention mechanism can boost performance of RNN-based systems (for examples Li et al., 2018; W. Wang et al., 2017). Fewer attempts were made to apply RecNN with lower performance, suggesting that processing words sequentially may be more informative than a tree structure.

TABLE 11: Performance in opinion target extraction using the SemEval 2014 dataset (restaurant and laptop domains). The best outcomes are highlighted in blue

Domain	No	Study	Model	Performance (F1)
Restaurant	1	Poria, Cambria, et al. (2016)	Deep CNN + Amazon WE + POS + LP	87.17
	2	Li et al. (2018)	LTSMs + Truncated History-Attention (THA) and Selective Transformation Network (STN)	85.61
	3	W. Wang et al. (2017)	Coupled Multi-layer Attentions (CMLA) based on GRU	85.29
	4	W. Wang et al. (2016)	RecNN + CRF + POS + Name list + Sentiment Lexicon	84.93
	5	Toh & Wang (2014)	CRF + POS + dependency tree based features	84.01
	6	Ye et al. (2017)	Dependency-Tree Based CNN + POS + chunk	83.97
	7	Xue et al. (2017)	Multi-task learning neural network combines BiLSTM and CNN layers	83.65
	8	X. Wang et al. (2016)	Uni-directional Elman RNN	82.12
	9	Liu et al. (2015)	Bi-Elman-RNN + POS + chunk + Amazon WE	82.06
	10	Yuan et al. (2017)	LSTM + Local Context + Senna WE	80.62
	11	Tay, Tuan, et al. (2017)	Holo DyMemNN	79.73
	12	Nguyen & Shirai (2015)	PhaseRecNN + CRF + multiple global functions	62.21
Laptop	1	Poria, Cambria, et al. (2016)	Deep CNN + Amazon WE + POS + LP	82.32
	2	Li et al. (2018)	LTSMs + Truncated History-Attention (THA) and Selective Transformation Network (STN)	79.52
	3	Li & Lam (2017)	Memory Interaction Network (MIN) based on LSTM with extended memory	77.58
	4	W. Wang et al. (2016)	RecNN + CRF + POS + Name list + Sentiment Lexicon	78.42
	5	W. Wang et al. (2017)	Coupled Multi-layer Attentions (CMLA) based on GRU	77.8
	6	Ye et al. (2017)	Dependency-Tree Based CNN + POS + chunk	75.66
	7	X. Wang et al. (2016)	Uni-directional Elman RNN	75.45
	8	Liu et al. (2015)	LSTM-RNN+ POS + chunk + Amazon WE	75
	9	Yuan et al. (2017)	BiLSTM + Local Context + Senna WE	74.78
	10	Chernyshevich (2014)	CRF + NER + POS + parsing + semantic analysis + additional reviews	74.55
	11	Tay, Tuan, et al. (2017)	Holo DyMemNN	74.03

Table 12 compares the performance of models across different languages within the SemEval 2016 dataset for the restaurant domain. It is apparent that the performance of models in English is better than for other languages, followed by French and Spanish. It is also interesting to see that the LSTM models (T. Chen et al., 2017; Li et al., 2018) and hybrid models (Xue et al., 2017) show higher performance than the best models in the SemEval competition. There is, therefore, some evidence that the LSTM-based models are more effective in multilingual environments, due to their ability to record past and future contexts of words.

TABLE 12: Performance in opinion target extraction with SemEval 2016 dataset in restaurant domain. Shaded cells highlight the best models in the SemEval competition

No	Study	Language	Model	Performance (F1)
1	Li et al. (2018)	English	LTSMs + Truncated History-Attention (THA) and Selective Transformation Network (STN)	73.61%
2	Li and Lam (2017)	English	Memory Interaction Network (MIN) based on LSTM with extended memory	73.44%
3	Xue et al. (2017)	English	Multi-task learning neural network combines BiLSTM and CNN layers	72.95%
4	T. Chen et al. (2017)	English	BiLSTM + Google WE + CRF	72.44%
5	Toh and Su (2016)	English	RNN + WE + Name List + DP Name List + Word Cluster	72.34%
6	T. Chen et al. (2017)	Spanish	BiLSTM + Google WE + CRF	71.7%
7	Álvarez-López, Juncal-Martínez, Fernández-Gavilanes, Costa-Montenegro and González-Castaño (2016)	Spanish	CRF	68.39%
8	T. Chen et al. (2017)	French	BiLSTM + Google WE + CRF	73.5%
9	Kumar, Kohail, Kumar, Ekbal, and Biemann (2016)	French	CRF + POS	69.64%
10	T. Chen et al. (2017)	Russian	BiLSTM + Google WE + CRF	67.08%
11	Kumar et al. (2016)	Dutch	CRF + POS	64.37%
12	T. Chen et al. (2017)	Dutch	BiLSTM + Google WE + CRF	64.29%
13	T. Chen et al. (2017)	Turkish	BiLSTM + Google WE + CRF	63.76%

It should be emphasised that in this dataset most of the sentences consist only of one target term, and most target terms are expressed by a single word. Therefore, the CNNs can extract the target efficiently. However, the comparison also shows that when RNNs are incorporated with other components such as attention and MemNet, they have comparable power. Such combinations can overcome the weakness of RNNs in capturing key phrases.

4.2 Aspect category detection

Similar to the OTE task, the DNN model has outperformed the best performing supervised machine learning models (Table 12), with their performance in ACD reaching F1 60-70 in English, and over 50 for other languages. In this task, more datasets and more languages have been used, from which can be inferred that the performance in English is much higher than for other languages. In term of the model, CNN seems to have the best performance for this task, with winning models in SemEval ABSA 2016 by Toh et al. (2016) and Ruder et al. (2016) and an outperforming model by Xue et al. (2017). Nevertheless, because of the limited neural network studies in this ACD task, it is difficult to conclude which model achieves the best performance.

TABLE 13: Performance of DNN models in aspect category detection tasks from SemEval 2014-2016

Dataset	No	Study	Domain	Language	Features	Performance
SemEval '14	1	Xue et al. (2017)	Restaurant	EN	Multi-task learning neural network combines BiLSTM and CNN layers	F1: 88.91%
	2	Kiritchenko, Zhu, Cherry, & Mohammad (2014)	Restaurant	EN	Five binary SVMs + n-grams + sentiment lexicon	F1: 88.58%
	3	Tay, Tuan, et al. (2017)	Restaurant	EN	Tensor DyMemNN	F1: 81.68%
SemEval '15	1	Toh and Su (2015)	Restaurant	EN	Sigmoidal feedforward network + bigram + name list + head word + word cluster	F1: 70.38%
			Laptop	EN		F1: 48.41%
	2	Xue et al.	Restaurant	EN	Multi-task learning neural network combines	F1: 65.97%

Dataset	No	Study	Domain	Language	Features	Performance
		(2017)			BiLSTM and CNN layers	
SemEval '16	1	Toh and Su (2016)	Restaurant	EN	CNN + WE +head word + name list + word cluster	F1: 75.10%
			Laptop	EN		F1: 59.83%
	2	Xue et al. (2017)	Restaurant	EN	Multi-task learning neural network combines BiLSTM and CNN layers	F1: 76.42%
	3	Tamchyna et al. (2016)	Restaurant	TU	Binary classifier (deep LSTM) for each category	F1: 61.03%
			Restaurant	RU		F1: 64.83%
	4	Álvarez-López et al. (2016)	Restaurant	ES	SVM + word list	F1: 70.59%
	5	Brun et al. (2016)	Restaurant	FR	Rich linguistic features + CRF (2 stages: 1st detect explicit terms, 2nd detect implicit aspect from the whole sentence)	F1: 61.21%
	6	Çetin, Yıldırım, Özbey and Eryiğit (2016)	Restaurant	DU	Linear classification with Logistic Regression	F1: 60.15%
	7	Ruder et al. (2016)	Mobile phone	DU	CNN + concatenated vectors	F1: 45.55%
			Hotel	AR		F1: 52.11%

4.3 Sentiment polarity of aspect-based consumer reviews

The task of aspect level sentiment polarity is more challenging than general task of sentiment analysis because the model needs to incorporate the impacts of context words towards the target or aspect. A general approach for using DNN in this task is through representing context, generating a target representation, and then identifying the important sentiment words for the target. In polarity classification, although many deep learning techniques have been proposed, there has not yet been an attempt that uses the RecNN model (Table 13). Similar to the task of OTE, the RNNs have demonstrated their competitive performance, in terms of capturing long-term dependency in sentences and general semantic classification. Furthermore, the best performers are the RNNs that incorporate attention or memory networks. This shows that with an attention weight aggregated from a lower level, the models can learn how to concentrate on different parts of the sentence to classify target and opinion words and the link between them.

TABLE 14: Performance in opinion target extraction with SemEval 2014 dataset (restaurant and laptop domains). Cells in shading indicate the best model in SemEval competition

Domain	No	Study	Task: Sentiment analysis	
			Features	Performance
Laptop	1	Kiritchenko et al. (2014)	SVM + n-grams + parse trees, + sentiment lexica	Acc: 80.95%
	2	P. Chen et al. (2017)	Recurrent Attention on Memory (RAM) + attention layers	Acc: 74.65% (3-way)
	3	R. Ma et al. (2017)	Feature-based Compositing Memory Networks	Acc: 73.94% (3-way)
	4	Wagner et al. (2014)	SVM + n-grams + parse trees, + sentiment lexica	Acc: 70.48%
	5	Y. Wang et al. (2016)	LSTM + aspect attention + aspect embeddings	Acc: 68.9% (3 way); 87.6% (binary)
	6	Tay, Luu, et al. (2017)	Aspect Fusion LSTM	Acc: 68.81% (3 way); 83.58% (binary)
Restaurant	1	Cheng et al. (2017)	BiGRU + aspect attention + sentiment attention	Acc: 85.1% (3-way); 91.3% (binary)
	2	R. Ma et al. (2017)	Feature-based Compositing Memory Networks	Acc: 82.03% (3-way)
	3	Tang, Qin, et al. (2016)	Deep memory network	Acc: 80.95% (3-way)
	4	P. Chen et al. (2017)	Recurrent Attention on Memory (RAM) + attention layers	Acc: 80.59% (3-way)
	5	Kiritchenko et al. (2014)	SVM + n-grams + parse trees, + sentiment lexica	Acc: 80.15%
	6	Y. Wang et al. (2016)	LSTM + aspect attention + aspect embeddings	Acc: 77.2% (3 way); 90.9% (binary)

7	Tay, Luu, et al. (2017)	Aspect Fusion LSTM	Acc: 75.44% (3 way); 87.78% (binary)
8	Wagner et al. (2014)	SVM + n-grams + parse trees, + sentiment lexica	Acc: 70.48%

4.4 Sentiment polarity of targeted text

Table 14 shows various performance indicators of models based on a Twitter dataset by Dong et al. (2014). Compared to the performance indicators reported in Section 4.3, it is interesting to observe that the accuracy of this domain is lower than in the customer review domain, which is largely due to the characteristics of tweets – short, highly expressive, high use of sarcasms, and less grammatical correctness than review texts (Giachanou & Crestani, 2016). It also shows that the CNN model has not yet been utilized, possibly because of its weaknesses in processing this type of data. Overall, the performance of RNN and RecNN are similar, with accuracy ranging from 69 to 72. While the CNN and RNN may work better in a grammatically correct context, overall this indicates that for the identification of sentiment polarity of targeted text, the tree structure and parser represent a promising approach.

TABLE 14: Performance on Twitter dataset by Dong et al. (2014)

No	Study	Task: Sentiment polarity	
		Features	Performance
1	B. Wang et al. (2017)	TDParse+ (dependency parser and three sentiment lexica)	F1: 69.8 (3-way) Acc: 72.1 (3-way)
2	M. Zhang et al. (2016)	Bidirectional GRNN & 3-way gate	Acc: 71.96 (3-way)
3	Vo & Zhang (2015)	Dependency parser & sentiment lexicon	F1: 69.9 (3-way) Acc: 71.1 (3-way)
4	Tang, Qin, Feng, et al. (2015)	Target Dependent LSTM	Acc: 70.8 (3-way)
		Target Connection LSTM	Acc: 71.5 (3-way)
5	P. Chen et al. (2017)	Recurrent Attention on Memory (RAM) + attention layers	Acc: 69.36 (3-way)
6	Dong et al. (2014)	Adaptive Recursive Neural Network + uni/bigram features + SVM classifier	Acc: 66.3 (3 way)

5 CHALLENGES

From the above discussion, it is clear that DL methods are still in their infancy. There are cases where the performance of DL methods is not as effective as expected. An example comes from Xu, Liu, Wang, & Yin, (2018) who attempted to approach three ABSA tasks with CNN models but achieved lower outcomes than with the SVM approach. Yuan et al. (2017) found that a purely window-based neural network produces outcomes that are comparable to an LSTM-RNN approach, and concluded that local context rather than long-term dependencies were important for aspect extraction. A study by Al-Smadi et al. (2017) for Arabic hotel reviews demonstrated that the SVM approach outperforms other deep RNN approaches for all ABSA tasks. All this illustrates that there are still significant challenges in terms of the application of DL methods to sentiment analysis in general and to ABSA in particular.

5.1 Domain adaptation

One major challenge for ABSA is the current focus on consumer reviews, which raises the issue of domain adaptation, namely whether the trained parameters in one domain can be applied to another domain. It is apparent that the sentiment of a word can only be determined given its domain or context. For example, “small” contains a positive sentiment in the electronics domain in “the phone is small and convenient” but it has negative sentiment in a restaurant review when it states, “the portion is small”. Given numerous domains, domain adaptation is clearly important to exploit the knowledge from one domain and increase the effectiveness of the analysis (Dragoni & Petrucci, 2017).

With respect to ABSA tasks, it is clear from the above findings that while one method may perform well in one domain, there is no guarantee of similar performance others. For example, Ding et al. (2017b) reported a much lower score in cross-domain performance compared to in-domain. Most noticeably, the performance of models varies significantly between domains. Evidence comes from the SemEval 2014 dataset, where performance in the restaurant domain is reported to be higher than in the laptop domain, for all studies and tasks (Li et al., 2018; Poria, Cambria, et al., 2016).

One possible explanation is that the prevalence of aspect phrases is higher within the laptop domain (i.e. 36.55% versus 24.56%), making it more difficult to predict than in the case of single-word aspects (Poria, Cambria, et al., 2016). Furthermore, consumer reviews in general are highly product-oriented, which means that most of the aspects or opinions are expressed with nouns or noun phrases, while in reality, aspects and opinion can be represented in different formats, and the co-existence of opinionated texts and non-opinionated texts is frequent (De Clercq, 2016). In unsupervised machine learning approaches, the issue can be resolved by incorporating a domain-specific lexicon (Liu, 2015), which could also be considered for DNN models. Ruder et al. (2017) have attempted to overcome the issue through domain adaptation, simplified by domain similarity metrics to guide the selection of appropriate training data. Another way to overcome the domain adaption is to pre-train the word embeddings in a large similar corpora (Ruder et al., 2016), which has shown promise as discussed in session 3.1.1. Similarly, Dragoni et al. (2017) suggested that the domain adaptation will be more effective if the word embeddings is created from an opinion-based corpus rather than a general purpose one (such as Wikipedia). They proposed NeuroSent, a tool for calculating the linguistic overlaps between different domains for conjecturing sentiment polarity.

5.2 Multilingual application

Lo, Cambria, Chiong, & Cornforth (2017) argued that multilingual sentiment analysis has certain challenges, including word sense ambiguity, language-specific structure and translation errors. Peng et al. (2018) have illustrated this in the case of Chinese, where each sub-word may encode semantics. Thus, the verb 'shine' contains 'sun' and 'moon' as sub-elements. This is radically different from English where only character N-grams (i.e. "pre", "sub") contain semantics. Therefore, it requires a higher effort to encode and decode the former type of language.

Despite the fact that DNN models require less language-specific features (Tamchyna et al., 2016), this review has highlighted that ABSA has not yet achieved its potential in a multilingual environment. The first issue stems from the fact that there are insufficient resources for many languages to construct NLP models. This is particularly the case for low-resource languages, which lack of large monolingual or parallel corpora such as Hindi or Tegulu. Is observed that word vectors in those type of languages obtain lower quality than others (Grave et al., 2018). It is also clear that there are yet no benchmark ABSA datasets on different languages. Apart from SemEval 2016, there is merely a small number of product review datasets in Chinese (Feng et al., 2018; Gu et al., 2017), in Hindi (Akhtar, Kumar, et al., 2016), and in Vietnamese (Mai & Le, 2018).

It is, therefore, hardly surprising that there are few successful attempts at using DL methods on ABSA with different languages, with the exception of Ruder et al. (2016) and T. Chen et al. (2017) for French, Spanish, Russian, Dutch and Turkish. From SemEval 2016, it is evident that the performance of models varies between languages, with higher scores recorded in English and Chinese but lower ones in French, Spanish, Dutch, and Russian (Pontiki et al., 2016).

Ruder et al. (2016) have suggested incorporation of different embeddings trained on a range of corpora in different languages. Peng et al. (2018) successfully incorporated radical, character and word embeddings into Chinese to overcome the issue of multi-word aspect representation. Therefore, it is expected that training the models to associate with different surface forms could help to reduce the performance differences.

5.3 Technical requirements

Best performing ABSA systems generally use manually labelled data and language specific resources for training on a particular domain and language (Pontiki et al., 2014, 2015, 2016). Particularly the DL based systems require a significant amount of labelled data for training (Araque et al., 2017; T. Chen et al., 2017). For example, one major issue in Tang, Qin, Feng, et al. (2015) is the failure to produce consistent results, possibly due to a small training corpus (B. Wang et al., 2017).

Another issue is related to computational resources and time. Despite the improvement in technology that reduces computational time in training for DNN models, reported by Gu et al. (2017) and Al-Smadi et al. (2017), when compared to conventional machine learning, time is still a current issue for DNN models. For example, P. Chen et al. (2016) reported an acceptable time span of four hours to process over 3000 sentences.

5.4 Linguistic complications

As discussed in Schouten et al. (2016), there are still challenges in terms of language that have not yet been satisfactorily addressed in current studies. They include the issue of comparative sentences, where it is difficult to detect whether any aspect is preferred. Similar problems exist with conditional sentences (irrealis

phenomenon), where it is difficult to extract sentiment from an unknown/unreal situation. Also highly complex is analysis from sentences that contain negation and valence-shifting, where the polarity can be flipped, and sentiment value can be decreased or increased.

It is also challenging to extract implicit aspects, which can only be read between the lines (Rana et al., 2016). The same text may be read differently in a different situation/ A classic example comes from Pang & Lee (2008) "go read the book" expresses positive sentiments in the case of a book review, but implies a negative sentiment as a film review. One consideration is to undertake "co-reference", reflecting aspects with pronouns or synonymous phrases; however, not much research exists as yet (De Clercq, 2016).

NLP methods also need to catch up with the evolution of user-generated content, which is quite different from standard text. It is characterized by its "noisiness" from highly expressive tokens such as emoticons, flooding (repetition of some characters such as "loooooo") as well as misspellings, grammatical errors, abbreviations and more use of sarcasm, irony, humour and metaphor, particularly for twitters (De Clercq, 2016; Giachanou & Crestani, 2016). This makes it more difficult to train with tools that were originally trained from a standard text (De Clercq, 2016).

The difficulties increase with regard to other languages, with Chinese with words that are ambiguous in terms of semantics and syntax (Peng, Cambria, & Hussain, 2017), in Hindi and Arabic through the issue of multi-dialects and lately also for Arabizi - Arabic words with Latin characters (El-Masri, Altrabsheh, & Mansour, 2017).

A promising approach suggested by Schouten et al. (2016) is to evolve to the more concept-centric approach of the knowledge base. Recent works in SenticNet and SSWE suggest that the incorporation of the knowledge base and recent language evolution is promising.

6 CONCLUSION

With the advent of user-generated content as a rich source of subjective information, there have been vigorous attempts to analyze, classify, understand and predict the nature and opinion polarity of written languages to fine-grained levels. This analysis has presented a comprehensive overview of major deep learning approaches and provided a precise comparison of these approaches for sentiment analysis at aspect level. For this analysis, more than 40 approaches were summarised and categorised according to their main architecture and classification tasks. Common approaches include standard and variants of Convolutional Neural Networks (CNN), Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). To boost the performance of models, studies have included pre-trained and fine-tuned word embeddings, and incorporating linguistic factors in the form of part-of-speech and grammatical rules as well as exploring concept-based knowledge.

However, from the review of the state-of-the-art in aspect-level sentiment analysis and presented in this paper, it is clear that ABSA and deep learning are still in the early stages. Given the relationship between aspect and opinion, improved performance can be obtained by joint extraction and classification of aspect, category and sentiment. However, many robust studies opt to perform only aspect extraction or categorization, and those who jointly perform aspect detection and sentiment analysis, have not yet achieved optimal performance. Therefore, there is the need for a combined approach that can undertake both tasks and create more pervasive sentiment analysis at aspect level. Research would further benefit from a more concept-centric approach to connect knowledge bases with deep learning methods.

APPENDIX: LIST OF ABBREVIATIONS

ABSA:	Aspect-Based Sentiment Analysis
ACD:	Aspect Category Detection
CNN:	Convolutional Neural Networks
CRF:	Conditional Random Field
DL:	Deep Learning
DNN:	Deep Neural Networks
GRU:	Gated Recurrent Unit
LSTM:	Long-Short Term Memory
ME:	Maximum Entropy
MemNet:	Memory Networks
NER:	Name Entity Recognition
NLP:	Natural Language Processing
OTE:	Opinion Target Extraction
POS:	Part Of Speech
RAE:	Recursive Auto-encoders
RecNN:	Recursive Neural Networks
RNN:	Recurrent Neural Networks
SemEval:	International Workshop on Semantic Evaluation
SP:	Sentiment Polarity
SVM:	Support Vector Machine
WE:	Word Embeddings

REFERENCES

- Akhtar, M. S., Ekbal, A., & Bhattacharyya, P. (2016). Aspect based sentiment analysis in hindi: Resource creation and evaluation. In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference (LREC)*.
- Akhtar, M. S., Gupta, D., Ekbal, A., & Bhattacharyya, P. (2017). Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis. *Knowledge-Based Systems*, 125, 116–135. <https://doi.org/10.1016/j.knosys.2017.03.020>
- Akhtar, M. S., Kumar, A., Ekbal, A., & Bhattacharyya, P. (2016). A Hybrid Deep Learning Architecture for Sentiment Analysis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 482–493). Retrieved from <http://aclweb.org/anthology/C16-1047>
- Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2017). Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. *Journal of Computational Science*. <https://doi.org/10.1016/j.jocs.2017.11.006>
- Alam, M. H., Ryu, W. J., & Lee, S. K. (2016). Joint multi-grain topic sentiment: Modeling semantic aspects for online reviews. *Information Sciences*, 339, 206–223. <https://doi.org/10.1016/j.ins.2016.01.013>
- Álvarez-López, T., Juncal-Martínez, J., Fernández-Gavilanes, M., Costa-Montenegro, E., & González-Castaño, J. F. (2016). GTI at SemEval-2016 Task 5: SVM and CRF for Aspect Detection and Unsupervised Aspect-Based Sentiment Analysis. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 306–311. Retrieved from <http://aclweb.org/anthology/S16-1049>
- Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77, 236–246. <https://doi.org/10.1016/j.eswa.2017.02.002>
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *ArXiv Preprint ArXiv:1409.0473* (2014).
- Bengio, Y., Schwenk, H., Senécal, J. S., Morin, F., & Gauvain, J. L. (2006). Neural probabilistic language models. *Studies in Fuzziness and Soft Computing*, 194, 137–186. https://doi.org/10.1007/10985687_6
- Brun, C., Perez, J., & Roux, C. (2016). XRCE at SemEval-2016 Task 5: Feedbacked Ensemble Modelling on Syntactico-Semantic Knowledge for Aspect Based Sentiment Analysis. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, 282–286.
- Cambria, E., Poria, S., Gelbukh, A., & Thelwall, M. (2017). Sentiment Analysis Is a Big Suitcase. *IEEE Intelligent Systems*, 32(6), 74–80. <https://doi.org/10.1109/MIS.2017.4531228>
- Çetin, F. S., Yildirim, E., Özbey, C., & Eryiğit, G. (2016). TGB at SemEval-2016 Task 5 : Multi-Lingual Constraint System for As- pect Based Sentiment Analysis. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, (Slot 1), 27–35. Retrieved from <http://link.springer.com/10.1007/978-3-642-19460-3>
- Chaudhuri, A., & Ghosh, S. K. (2016). Sentiment Analysis of Customer Reviews Using Robust Hierarchical Bidirectional Recurrent Neural Network. In R. Silhavy, R. Senkerik, Z. K. Oplatkova, P. Silhavy, & Z. Prokopova (Eds.), *Advances in Intelligent Systems and Computing* (Vol. 464).

<https://doi.org/10.1007/978-3-319-33625-1>

- Chen, P., Sun, Z., Bing, L., & Yang, W. (2017). Recurrent Attention Network on Memory for Aspect Sentiment Analysis. *Emnlp*, 452–461. Retrieved from <http://aclweb.org/anthology/D17-1047>
- Chen, P., Xu, B., Yang, M., & Li, S. (2016). Clause sentiment identification based on convolutional neural network with context embedding. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016* (pp. 1532–1538). <https://doi.org/10.1109/FSKD.2016.7603403>
- Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, 72, 221–230. <https://doi.org/10.1016/j.eswa.2016.10.065>
- Cheng, J., Zhao, S., Zhang, J., King, I., Zhang, X., & Wang, H. (2017). Aspect-level Sentiment Classification with HEAT (HiErarchical ATtention) Network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17* (pp. 97–106). <https://doi.org/10.1145/3132847.3133037>
- Chernyshevich, M. (2014). IHS R & D Belarus : Cross-domain Extraction of Product Features using Conditional Random Fields. *Semeval*, (SemEval), 309–313.
- Chiu, J. P. C., & Nichols, E. (2015). Named Entity Recognition with Bidirectional LSTM-CNNs. *ArXiv Preprint ArXiv:1511.08308*, (2003). Retrieved from <http://arxiv.org/abs/1511.08308>
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *ArXiv Preprint ArXiv:1406.1078*. <https://doi.org/10.3115/v1/D14-1179>
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv Preprint ArXiv:1412.3555*.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12, 2493–2537. <https://doi.org/10.1.1.231.4614>
- De Clercq, O. (2016). The many aspects of fine-grained sentiment analysis: An overview of the task and its main challenges. In *HUSO 2016* (pp. 23–28). Retrieved from <https://biblio.ugent.be/publication/8500953/file/8500962>
- Ding, Y., Yu, C., & Jiang, J. (2017a). A neural network model for semi-supervised review aspect identification. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10235 LNAI, pp. 668–680). https://doi.org/10.1007/978-3-319-57529-2_52
- Ding, Y., Yu, J., & Jiang, J. (2017b). Recurrent Neural Networks with Auxiliary Labels for Cross-Domain Opinion Target Extraction. *Aaai*, 3436–3442.
- Do, Q. N. T., Bethard, S., & Moens, M.-F. (2017). Improving Implicit Semantic Role Labeling by Predicting Semantic Frame Arguments. *ArXiv Preprint ArXiv:1704.02709*. Retrieved from <http://arxiv.org/abs/1704.02709>
- Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., & Xu, K. (2014). Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification. *Acl-2014*, 49–54.
- Dragoni, M., & Petrucci, G. (2017). A Neural Word Embeddings Approach for Multi-Domain Sentiment Analysis. *IEEE Transactions on Affective Computing*, 8(4), 457–470. <https://doi.org/10.1109/TAFFC.2017.2717879>
- Du, H., Xu, X., Cheng, X., Wu, D., Liu, Y., & Yu, Z. (2016). Aspect-specific Sentimental Word Embedding for Sentiment Analysis of Online Reviews. *International Conference Companion on World Wide Web*, 29–30. <https://doi.org/10.1145/2872518.2889403>
- El-Masri, M., Altrabsheh, N., & Mansour, H. (2017). Successes and challenges of Arabic sentiment analysis research: a literature review. *Social Network Analysis and Mining*. <https://doi.org/10.1007/s13278-017-0474-x>
- Elman, J. L. (1991). Distributed Representations, Simple Recurrent Networks, And Grammatical Structure. *Machine Learning*, 7(2), 195–225. <https://doi.org/10.1023/A:1022699029236>
- Fan, Y., Qian, Y., Xie, F., & Soong, F. K. (2014). TTS synthesis with bidirectional LSTM based Recurrent Neural Networks. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* (pp. 1964–1968).
- Feng, J., Cai, S., & Ma, X. (2018). Enhanced sentiment labeling and implicit aspect identification by integration of deep convolution neural network and sequential algorithm. *Cluster Computing*, pp. 1–19. <https://doi.org/10.1007/s10586-017-1626-5>
- Fernández-Gavilanes, M., Álvarez-López, T., Juncal-Martínez, J., Costa-Montenegro, E., & Javier González-Castaño, F. (2016). Unsupervised method for sentiment analysis in online texts. *Expert Systems with Applications*, 58, 57–75. <https://doi.org/10.1016/j.eswa.2016.03.031>
- García-Pablos, A., Cuadros, M., & Rigau, G. (2018). W2VLDA: Almost unsupervised system for Aspect Based Sentiment Analysis. *Expert Systems with Applications*, 91, 127–137. <https://doi.org/10.1016/j.eswa.2017.08.049>
- Giachanou, A., & Crestani, F. (2016). Like it or not: A survey of Twitter sentiment analysis methods. *ACM*

- Comput Surv*, 49(2), Article 28; 1-41. <https://doi.org/10.1145/2938640>
- Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing, 57, 345–420.
- Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 10(1), 1–309. <https://doi.org/10.2200/S00762ED1V01Y201703HLT037>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. *Nature*, 521(7553), 800. <https://doi.org/10.1038/nmeth.3707>
- Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. Retrieved from <http://arxiv.org/abs/1802.06893>
- Graves, A. (2008). Supervised Sequence Labelling with Recurrent Neural Networks. *ArXiv Preprint ArXiv:1308.0850*, 12(1), 126–140. <https://doi.org/10.1145/2911451.2926729>
- Gu, X., Gu, Y., & Wu, H. (2017). Cascaded Convolutional Neural Networks for Aspect-Based Opinion Summary. *Neural Processing Letters*, 46(2), 581–594. <https://doi.org/10.1007/s11063-017-9605-7>
- Hassan, A., & Mahmood, A. (2017). Deep Learning approach for sentiment analysis of short texts. *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, (September), 705–710. <https://doi.org/10.1109/ICCAR.2017.7942788>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04* (p. 168). <https://doi.org/10.1145/1014052.1014073>
- Irsoy, O., & Cardie, C. (2014). Opinion Mining with Deep Recurrent Neural Networks. *Emnlp-2014*, 720–728. <https://doi.org/10.3115/v1/D14-1080>
- Jebbara, S., & Cimiano, P. (2016). Aspect-Based Sentiment Analysis Using a Two-Step Neural Network Architecture. *Semantic Web Evaluation Challenge*, 153–167. <https://doi.org/10.3233/978-1-61499-672-9-1123>
- Jebbara, S., & Cimiano, P. (2017). Aspect-Based Relational Sentiment Analysis Using a Stacked Neural Network Architecture. *ArXiv Preprint ArXiv:1709.06309*. <https://doi.org/10.3233/978-1-61499-672-9-1123>
- Jiménez-Zafra, S. M., Martín-Valdivia, M. T., Martínez-Cámara, E., & Ureña-López, L. A. (2016). Combining resources to improve unsupervised sentiment analysis at aspect-level. *Journal of Information Science*, 42(2), 213–229. <https://doi.org/10.1177/0165551515593686>
- Kessler, J. S., Eckert, M., Clark, L., & Nicolov, N. (2010). The ICWSM 2010 JDPa Sentiment Corpus for the Automotive Domain. *4th Int'l AAAI Conference on Weblogs and Social Media Data Workshop Challenge (ICWSM-DWC 2010)*. Retrieved from <http://www.cs.indiana.edu/~jaskessl/icwsml0.pdf>
- Kim, Y. (2014). Convolutional Neural Networks for Sentence Classificatio. *Arxiv*, 23–31. <https://doi.org/10.1145/1599272.1599278>
- Kiritchenko, S., Zhu, X., Cherry, C., & Mohammad, S. (2014). NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* (pp. 437–442). <https://doi.org/10.3115/v1/S14-2076>
- Kumar, A., Kohail, S., Kumar, A., Ekbal, A., & Biemann, C. (2016). IIT-TUDA at SemEval-2016 Task 5: Beyond Sentiment Lexicon: Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis. *Proceedings of the 10th International Workshop on Semantic Evaluation, (SemEval)*, 1129–1135.
- Lakkaraju, H., Socher, R., & Manning, C. D. (2014). Aspect Specific Sentiment Analysis using Hierarchical Deep Learning. *NIPS WS on Deep Neural Networks and Representation Learning*, 1–9.
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition. *ArXiv Preprint ArXiv:1603.01360*. <https://doi.org/10.18653/v1/N16-1030>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Li, X., Bing, L., Li, P., Lam, W., & Yang, Z. (2018). Aspect Term Extraction with History Attention and Selective Transformation. *ArXiv Preprint ArXiv:1805.00760*. Retrieved from <http://arxiv.org/abs/1805.00760>
- Li, X., & Lam, W. (2017). Deep Multi-Task Learning for Aspect Term Extraction with Memory Interaction *. In *ACL* (pp. 2886–2892). Retrieved from <http://aclweb.org/anthology/D/D17/D17-1310.pdf>
- Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. <https://doi.org/10.1017/CBO9781139084789>
- Liu, P., Joty, S., & Meng, H. (2015). Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1433–1443). Retrieved from <http://aclweb.org/anthology/D15-1168>
- Lo, S. L., Cambria, E., Chiong, R., & Cornforth, D. (2017). Multilingual sentiment analysis: from formal to informal and scarce resource languages. *Artificial Intelligence Review*, 48(4), 499–527. <https://doi.org/10.1007/s10462-016-9508-4>
- Ma, D., Li, S., Zhang, X., & Wang, H. (2017). Interactive attention networks for aspect-level sentiment

- classification. In *IJCAI International Joint Conference on Artificial Intelligence* (pp. 4068–4074).
- Ma, R., Wang, K., Qiu, T., Sangaiah, A. K., Lin, D., & Liaqat, H. Bin. (2017). Feature-based Compositing Memory Networks for Aspect-based Sentiment Classification in Social Internet of Things. *Future Generation Computer Systems*. <https://doi.org/10.1016/j.future.2017.11.036>
- Ma, X., & Hovy, E. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *ArXiv Preprint ArXiv:1603.01354*. <https://doi.org/10.18653/v1/P16-1101>
- Ma, Y., Peng, H., & Cambria, E. (2018). Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM. *Aaai*. Retrieved from <http://sentic.net/sentic-lstm.pdf>
- Mai, L., & Le, B. (2018). Aspect-Based Sentiment Analysis of Vietnamese Texts with Deep Learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10751 LNAI, pp. 149–158). https://doi.org/10.1007/978-3-319-75417-8_14
- Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M., & Balahur, A. (2018). WWW'18 Open Challenge: Financial Opinion Mining and Question Answering. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France*. ACM, New York, NY, USA.
- Manek, A. S., Shenoy, P. D., Mohan, M. C., & Venugopal, K. R. (2017). Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. *World Wide Web*, 20(2), 135–154. <https://doi.org/10.1007/s11280-015-0381-x>
- Marcheggiani, D., Frolov, A., & Titov, I. (2017). A Simple and Accurate Syntax-Agnostic Neural Model for Dependency-based Semantic Role Labeling. *ArXiv Preprint ArXiv:1701.02593*. Retrieved from <http://arxiv.org/abs/1701.02593>
- Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, 1–12. <https://doi.org/10.1162/153244303322533223>
- Nguyen, T. H., & Shirai, K. (2015). PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September), 2509–2514. Retrieved from <http://aclweb.org/anthology/D15-1298>
- Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Processing*, 1(2), 91–231. <https://doi.org/10.1561/1500000001>
- Parkhe, V., & Biswas, B. (2016). Sentiment analysis of movie reviews: finding most important movie aspects using driving factors. *Soft Computing*, 20(9), 3373–3379. <https://doi.org/10.1007/s00500-015-1779-1>
- Peng, H., Cambria, E., & Hussain, A. (2017). A Review of Sentiment Analysis Research in Chinese Language. *Cognitive Computation*. <https://doi.org/10.1007/s12559-017-9470-8>
- Peng, H., Ma, Y., Li, Y., & Cambria, E. (2018). Learning multi-grained aspect target sequence for Chinese sentiment analysis. *Knowledge-Based Systems*, 148, 55–65. <https://doi.org/10.1016/j.knosys.2018.02.034>
- Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). <https://doi.org/10.3115/v1/D14-1162>
- Pirayani, R., Gupta, V., & Singh, V. K. (2017). Movie Prism: A novel system for aspect level sentiment profiling of movies. *Journal of Intelligent and Fuzzy Systems*. <https://doi.org/10.3233/JIFS-169272>
- Plank, B., Søgaard, A., & Goldberg, Y. (2016). Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. *ArXiv Preprint ArXiv:1604.05529*. <https://doi.org/10.18653/v1/P16-2067>
- Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., ... Eryiğit, G. (2016). SemEval-2016 Task 5: Aspect Based Sentiment Analysis. *Proceedings of the 10th International Workshop on Semantic Evaluation*, 19–30. <https://doi.org/10.18653/v1/S16-1055>
- Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., & Androutsopoulos, I. (2015). SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado*. (pp. 486–495). <https://doi.org/10.1007/978-3-642-40837-3>
- Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2014). SemEval-2014 Task 4: Aspect Based Sentiment Analysis. *Proceedings of the 8th International Workshop on Semantic Evaluation*, (SemEval), 27–35. <https://doi.org/10.15520/ajcsit.v4i8.9>
- Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect Extraction for Opinion Mining with a Deep Convolutional Neural Network. *Knowledge-Based Systems*, 108, 42–49. <https://doi.org/10.1016/j.knosys.2016.06.009>
- Poria, S., Cambria, E., Gelbukh, A., Bisio, F., & Hussain, A. (2015). Sentiment Data Flow Analysis by Means of Dynamic Linguistic Patterns. *IEEE Computational Intelligence Magazine*, 10(4), 26–36. <https://doi.org/10.1109/MCI.2015.2471215>
- Poria, S., Chaturvedi, I., Cambria, E., & Bisio, F. (2016). Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis. In *Proceedings of the International Joint Conference on Neural Networks* (Vol. 2016–October, pp. 4465–4473). <https://doi.org/10.1109/IJCNN.2016.7727784>

- Rana, T. A., & Cheah, Y.-N. (2016). Aspect extraction in sentiment analysis : comparative analysis and survey. *Artificial Intelligence Review*, 46(4), 459–483. <https://doi.org/10.1007/s10462-016-9472-z>
- Rojas-Barahona, L. M. (2016). Deep learning for sentiment analysis. *Language and Linguistics Compass*, 10(12), 701–719. <https://doi.org/10.1111/Inc3.12228>
- Ruder, S., Ghaffari, P., & Breslin, J. G. (2016). INSIGHT-1 at SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Retrieved from <http://arxiv.org/abs/1609.02748>
- Saeidi, M., Bouchard, G., Liakata, M., & Riedel, S. (2016). SentiHood: Targeted Aspect Based Sentiment Analysis Dataset for Urban Neighbourhoods. *ArXiv:1610.03771*. Retrieved from <http://arxiv.org/abs/1610.03771>
- Saias, J. (2015). Sentiu: Target and Aspect based Sentiment Analysis in SemEval-2015 Task 12. *Proceedings of the 9th International Workshop on Semantic Evaluation, (SemEval)*, 767–771.
- Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Schouten, K., & Frasincar, F. (2016). Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3), 813–830. <https://doi.org/10.1109/TKDE.2015.2485209>
- Shen, Y., Yun, H., Lipton, Z. C., Kronrod, Y., & Anandkumar, A. (2017). Deep Active Learning for Named Entity Recognition. *ArXiv Preprint ArXiv:1707.05928*, 1–15. <https://doi.org/10.18653/v3>
- Socher, R., Perelygin, A., & Wu, J. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the ...*, 1631–1642. <https://doi.org/10.1371/journal.pone.0073791>
- Strubell, E., Verga, P., Belanger, D., & McCallum, A. (2017). Fast and Accurate Sequence Labeling with Iterated Dilated Convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Retrieved from <http://arxiv.org/abs/1702.02098>
- Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-To-End Memory Networks. In *Advances in neural information processing systems* (pp. 2440–2448). <https://doi.org/v5>
- Sutton, C. (2012). An Introduction to Conditional Random Fields. *Foundations and Trends® in Machine Learning*, 4(4), 267–373. <https://doi.org/10.1561/22000000013>
- Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. *Proceedings of ACL*, 1556–1566. <https://doi.org/10.1515/popets-2015-0023>
- Tamchyna, A., & Veselovská, K. (2016). UFAL at SemEval-2016 Task 5: Recurrent Neural Networks for Sentence Classification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (pp. 367–371). Retrieved from <https://aclweb.org/anthology/S16/S16-1059.pdf>
- Tang, D., Qin, B., Feng, X., & Liu, T. (2015). Effective LSTMs for Target-Dependent Sentiment Classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Retrieved from <http://arxiv.org/abs/1512.01100>
- Tang, D., Qin, B., & Liu, T. (2015a). Deep learning for sentiment analysis: Successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(6), 292–303. <https://doi.org/10.1002/widm.1171>
- Tang, D., Qin, B., & Liu, T. (2015b). Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1422–1432). <https://doi.org/10.18653/v1/D15-1167>
- Tang, D., Qin, B., & Liu, T. (2016). Aspect Level Sentiment Classification with Deep Memory Network. *ArXiv Preprint ArXiv:1605.08900*. Retrieved from <http://arxiv.org/abs/1605.08900>
- Tang, D., Wei, F., Qin, B., Yang, N., Liu, T., & Zhou, M. (2016). Sentiment Embeddings with Applications to Sentiment Analysis. In *IEEE Transactions on Knowledge and Data Engineering* (Vol. 28, pp. 496–509). <https://doi.org/10.1109/TKDE.2015.2489653>
- Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning Sentiment-Specific Word Embedding. *Acl*, 1555–1565. <https://doi.org/10.3115/1220575.1220648>
- Tay, Y., Luu, A. T., & Hui, S. C. (2017). Learning to Attend via Word-Aspect Associative Fusion for Aspect-based Sentiment Analysis. *ArXiv Preprint ArXiv:1712.05403*. Retrieved from <http://arxiv.org/abs/1712.05403>
- Tay, Y., Tuan, L. A., & Hui, S. C. (2017). Dyadic Memory Networks for Aspect-based Sentiment Analysis. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17* (pp. 107–116). <https://doi.org/10.1145/3132847.3132936>
- Thanaki, J. (2017). *Python Natural Language Processing*. Packt Publishing Ltd.
- Toh, Z., & Su, J. (2015). NLANGP: Supervised Machine Learning System for Aspect Category Classification and Opinion Target Extraction. *Proceedings of the 9th International Workshop on Semantic Evaluation, 14(SemEval)*, 496–501.
- Toh, Z., & Su, J. (2016). NLANGP at SemEval-2016 Task 5: Improving Aspect Based Sentiment Analysis using Neural Network Features. *Proceedings of SemEval-2016, 2015(Subtask 1)*, 282–288. Retrieved from <http://www.aclweb.org/anthology/S16-1045>
- Toh, Z., & Wang, W. (2014). DLIREC: Aspect Term Extraction and Term Polarity Classification System. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, (SemEval),

235–240.

- Toprak, C., Jakob, N., & Gurevych, I. (2010). Sentence and Expression Level Annotation of Opinions in User-Generated Discourse. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 1(July), 575–584. Retrieved from <http://www.aclweb.org/anthology/P10-1059>
- Tu, Z., Lu, Z., Liu, Y., Liu, X., & Li, H. (2016). Coverage-based Neural Machine Translation. *Arxiv*, 1–19. <https://doi.org/10.1145/2856767.2856776>
- Tutubalina, E., & Nikolenko, S. (2017). Combination of Deep Recurrent Neural Networks and Conditional Random Fields for Extracting Adverse Drug Reactions from User Reviews. *Journal of Healthcare Engineering*, 2017. <https://doi.org/10.1155/2017/9451342>
- Vicente, I. S., Saralegi, X., & Agerri, R. (2017). EliXa: A Modular and Flexible ABSA Platform. *ArXiv Preprint ArXiv:1702.01944*. Retrieved from <http://arxiv.org/abs/1702.01944>
- Vo, D. T., & Zhang, Y. (2015). Target-dependent twitter sentiment classification with rich automatic features. In *IJCAI International Joint Conference on Artificial Intelligence* (Vol. 2015–Janua, pp. 1347–1353).
- Wagner, J., Arora, P., Cortes, S., Barman, U., Bogdanova, D., Foster, J., & Tounsi, L. (2014). DCU: Aspect-based Polarity Classification for SemEval Task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* (pp. 223–229).
- Wang, B., Liakata, M., Zubiaga, A., & Procter, R. (2017). TDParse : Multi-target-specific sentiment recognition on Twitter. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (Vol. 1, pp. 483–493). Retrieved from <http://www.aclweb.org/anthology/E17/E17-1046.pdf>
- Wang, H., Lu, Y., & Zhai, C. (2011). Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11* (p. 618). <https://doi.org/10.1145/2020408.2020505>
- Wang, W., Pan, S. J., & Dahlmeier, D. (2017). Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms. *Aaai*, 3316–3322. Retrieved from <http://arxiv.org/abs/1702.01776>
- Wang, W., Pan, S. J., Dahlmeier, D., & Xiao, X. (2016). Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)*, 616–626.
- Wang, X., Liu, Y., Sun, C., Liu, M., & Wang, X. (2016). Extended dependency-based word embeddings for aspect extraction. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9950 LNCS, pp. 104–111). https://doi.org/10.1007/978-3-319-46681-1_13
- Wang, Y., Huang, M., Zhao, L., & Zhu, X. (2016). Attention-based LSTM for Aspect-level Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 606–615).
- Weichselbraun, A., Gindl, S., Fischer, F., Vakulenko, S., & Scharl, A. (2017). Aspect-based extraction and analysis of affective knowledge from social media streams. *IEEE Intelligent Systems*, 32(3), 80–88. <https://doi.org/10.1109/MIS.2017.57>
- Wu, H., Gu, Y., Sun, S., & Gu, X. (2016). Aspect-based Opinion Summarization with Convolutional Neural Networks. In *Proceedings of the International Joint Conference on Neural Networks* (Vol. 2016–Octob, pp. 3157–3163). <https://doi.org/10.1109/IJCNN.2016.7727602>
- Xu, L., Lin, J., Wang, L., Yin, C., & Wang, J. (2017). Deep Convolutional Neural Network based Approach for Aspect-based Sentiment Analysis. *Advanced Science and Technology Letters*, 143(Ast), 199–204. <https://doi.org/10.14257/astl.2017.143.41>
- Xu, L., Liu, J., Wang, L., & Yin, C. (2018). Aspect Based Sentiment Analysis for Online Reviews. In J. Park, V. Loia, G. Yi, & Y. Sung (Eds.), *Advances in Computer Science and Ubiquitous Computing* (Lecture No, Vol. 474). Springer Singapore.
- Xue, W., Zhou, W., Li, T., & Wang, Q. (2017). MTNA: A Neural Multi-task Model for Aspect Category Classification and Aspect Term Extraction On Restaurant Reviews. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2, 151–156.
- Yang, Z., Salakhutdinov, R., & Cohen, W. (2016). Multi-Task Cross-Lingual Sequence Tagging from Scratch. *ArXiv Preprint ArXiv:1603.06270*. Retrieved from <http://arxiv.org/abs/1603.06270>
- Ye, H., Yan, Z., Luo, Z., & Chao, W. (2017). Dependency-tree based convolutional neural networks for aspect term extraction. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10235 LNAI, pp. 350–362). https://doi.org/10.1007/978-3-319-57529-2_28
- Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative Study of CNN and RNN for Natural Language Processing. *ArXiv Preprint ArXiv:1708.02709*. Retrieved from <http://arxiv.org/abs/1702.01923>
- Young, T., Hazarika, D., Poria, S., & Cambria, E. (2017). Recent Trends in Deep Learning Based Natural Language Processing. *Arxiv*, 1–24. Retrieved from <http://arxiv.org/abs/1708.02709>
- Yuan, J., Zhao, Y., Qin, B., & Liu, T. (2017). Local contexts are effective for neural aspect extraction. In *Communications in Computer and Information Science* (Vol. 774, pp. 244–255). https://doi.org/10.1007/978-981-10-6805-8_20
- Zainuddin, N., Selamat, A., & Ibrahim, R. (2018). Hybrid sentiment classification on twitter aspect-based

- sentiment analysis. *Applied Intelligence*, 48(5), 1218–1232. <https://doi.org/10.1007/s10489-017-1098-6>
- Zaremba, W., & Sutskever, I. (2014). Learning to Execute. *ArXiv Preprint ArXiv:1410.4615*, 1–25. [https://doi.org/10.1016/S0893-6080\(96\)00073-1](https://doi.org/10.1016/S0893-6080(96)00073-1)
- Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. <https://doi.org/10.1002/widm.1253>
- Zhang, M., Zhang, Y., & Vo, D. (2016). Gated Neural Networks for Targeted Sentiment Analysis. *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, 3087–3093. Retrieved from <http://zhangmeishan.github.io/targeted-sentiment.pdf>
- Zhao, W., Guan, Z., Chen, L., He, X., Cai, D., Wang, B., & Wang, Q. (2017). Weakly-supervised Deep Embedding for Product Review Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering*. <https://doi.org/10.1109/TKDE.2017.2756658>