ABSTRACT

USING BERT ON ASPECT-BASED SENTIMENT ANALYSIS FOR ARABIC HOTEL REVIEWS

By

Mutaz Mohammad Younes

Aspect-based sentiment analysis in the Arabic language has been challenging over the years, and many researchers proposed deep learning models to achieve decent accuracy in such tasks. This research presents a state-of-the-art model based on Bidirectional Encoder Representations from Transformers that shows significant improvement compared to other results reported by researchers. The proposed models can accurately extract the aspects and their sentiments from the Arabic Hotel reviews. The model used in this research was trained from scratch on a large chunk of Arabic corpus and then post-trained on a domain-specific corpus. The proposed model is evaluated using a reference dataset of Arabic Hotels' reviews. Results show that our model achieves 74.4% F1-Score in aspect extraction and 90% Accuracy in aspect term polarity extraction. In other words, the proposed model outperforms baseline research on both tasks with an improvement of 45% for the aspect term extraction task and 14% for the aspect term polarity task.

Chapter One: Introduction

In the past several years, interest from the researchers in processing the Arabic language has grown significantly (1). The Arabic language is one of the Semitic languages and is considered a morphologically complex language (2, 3). The Arabic language is spoken or written in three ways, the first one is the Modern Standard Arabic (MSA), and the second one is Classical Arabic (CA), the older style of Arabic, and the third one is dialects (1). The Arabic dialects are different from each other; for example, the used words are different for each geographical region, or even the word itself can be used in a different pronunciation.

One of the areas that researchers recently focused on is analyzing the sentiment of online reviews in the Arabic language. Many online reviews and social media interactions are not being investigated yet; such a large number of reviews and interactions can provide powerful insights for companies interested in investing these unstructured data. Customer behavior may be interpreted via sentiment analysis, which can then be utilized to identify ways to enhance sales strategy. (4). For example, successful tourism organizations focus on online reviews to enhance performance and maintain sustainable long-term success. Such companies analyze the feedback from their customers and work to improve their services based on the feedback.

The usage of social media is rapidly increasing (5), which cases an electronic Word of Mouth (eWOM) (6), This increase in social media influences the decision-making strategies in many companies (7). Nowadays, social media is an integral part of our lives; everyone shares their feelings and opinions and discusses their responses to different issues instantly and effortlessly. The incremental use of social media via the internet affects all areas, such as governmental, commercial, products, and services (8). So it offers the opportunity to benefit from these social media websites for various purposes such as communication, advertising, feedback about products, and marketing (9).

Currently, Sentiment Analysis (SA) is considered one of the many branches in data mining (10), and it is an application of text classification tasks (11). SA is the field of study that uses a written text to analyze people's opinions, sentiments, attitudes, and emotions (12). SA is used in many areas, such as social media (13), brand monitoring (14), customer service (15), and market research (16). There are different SA types, fine-grained SA, which is the study that extracts the sentiment utterances on a more fine-grained level (17); Therefore, instead of extracting only positive and negative, it is possible to extract very negative, negative, neutral, positive, and very positive sentiments from the text. The second type is multilingual SA, which determines the polarity of sentences within a multilingual framework (18).

Finally, aspect-based SA (ABSA), ABSA can provide a new set of helpful information for companies. ABSA model gets as input a set of text such as reviews that discuss a particular entity. The model needs to detect the main aspects of the entities and then find the sentences' sentiment for each aspect (19).

1.1 Problem Statement

The ABSA task is divided into three sub-tasks:

1. Aspect term extraction (ATE): this task is about getting the related aspect terms in each review. For example, given the following review, "The hotel was great, but I did not like the food," the extracted aspects would be "hotel" and "food."
2. Aspect term polarity (ATP): this task is about extracting polarity for each aspect term. The polarity term can be positive, negative, or neutral. Given the previous example, the polarity for the word "hotel" is "positive," and the polarity for the word "food" is "negative."
3. Aspect category extraction (ACE): this task is about extracting the category for each aspect term. The annotators provided 37 different categories (e.g., "HOTEL#QUALITY," "LOCATION#GENERAL," "HOTEL#PRICES," "SERVICE#GENERAL"). Following the previous example, the word hotel will fall under "HOTEL#GENERAL," and the word "food" will fall under "FOOD_DRINKS#QUALITY."

1.2 Aim of Study

This research aims to use state-of-the-art natural language processing (NLP) models to enhance the performance of ABSA. It proposes a domain-specific Arabic Bidirectional Encoder Representations from Transformers (BERT) model that outperforms the results of different public Arabic BERT models. Then an implementation for deep learning models for different ABSA tasks is presented, and finally compare the results with the proposed model.

1.3 Research Structure

This research is organized as follows: Section Literature Review overviews the previous research work on ABSA tasks; it is split into four categories, unsupervised methods, machine learning methods, deep learning methods, and ABSA using BERT. Section Background describes the background of the used models and their architectures. Section Methodology describes the methodology. Section Results and Discussions discusses the model results and compares the results with the different models used for ABSA. Finally, section Conclusions and Future Work concludes this research and provides possibilities for future work.

Chapter Two: Literature Review

There was a significant amount of work on ABSA in the last few years. Many research papers used different machine learning and deep learning models and achieved top results in the different ABSA tasks. In recent years, and with BERT coming to the NLP world, many research papers focused on leveraging BERT to gain higher results in the ABSA tasks (20, 21, 22, 23, 24, 25, 26, 27). This section summarizes previous research papers that worked on ABSA tasks: ATE, ATP, and ACE. It has four sections, ABSA using machine learning, ABSA using unsupervised approaches, ABSA using deep learning, and finally ABSA using BERT.

ABSA tasks gained attention in 2014; many researchers focused on leveraging machine learning algorithms to solve the various ABSA tasks. In the beginning, researchers used machine learning algorithms such as SVM, maximum entropy classifier, CRF, logistic regression, and more. Section ABSA Using Machine Learning describes various research papers that used different machine learning algorithms to solve ABSA tasks; you will notice that some papers achieve very accurate results, even higher than the results gained from using BERT or deep learning methods. These papers manually extracted a rich and representative set of features, which allowed the machine learning algorithms to gain a high-level understanding of the data and produce such accurate results.

2.1 ABSA Using Machine Learning

In the ABSA task of Semantic Evaluation (SemEval) 2014, the UWB team (28) adapted an approach combined between the constrained and unconstrained systems; the authors trained the constrained system on the training data and the unconstrained system on additional data. The system for ATE achieved 79.53% as F1-score, which consists of Conditional Random Fields (CRF) with constrained features such as word occurrence, Bag of Words (BoW), Bigrams, Learned Dictionary (LD), Suffixes (S). In comparison, the unconstrained system uses the same features in addition to Word Clusters (WC). The system for ATP achieved an accuracy of 71.02%, consisting of a Maximum Entropy classifier with the BoW, Bag of Bigrams (BoB) as constrained features. And Bag of Clusters (BoC, Sentiment Dictionary (SD), SentiWordnet (SW) for the unconstrained approach. The ACE score was 79.39% F1-score; they adapted Maximum Entropy classifier with constrained features such as BoW, term frequency-inverse document frequency (Tf-IDF), and Latent Dirichlet Allocation (LDA), BoC for the unconstrained approach. Their system for the ACE polarity task obtains an accuracy of 70.20%; their system consists of a Maximum Entropy classifier with a constrained feature set such as BoW, BoB, Tf-IDF. In addition to an unconstrained case such as BoC, LDA, SD, SW. Since the ABSA task gained many researchers' attention on its first release, the competition creators re-launched the competition in 2016. Starting with one of the most exciting papers among the participants authored by Hercig et al. (29) used a Maximum Entropy classifier for both ACE and ATP tasks and used CRF for the ATE task. For ACE, a maximum entropy classifier is used with a threshold to decide which categories to keep from the prediction. In addition to the features they extracted, the authors used Glove, Continuous Bag-of-Words (CBOW), and LDA to extract more features for their models. The authors competed in nineteen constrained experiments and ranked in the first place for nine of them.

Another interesting paper authored by Alvarez-Lopez et al. (30) developed an SVM-based system for ACE and CRFs based system for ATE. They used words, lemmas, POS tags, and bigrams as features for the SVM model. The authors trained a separate binary SVM model for each category; this method allows the models to detect more than one category in the given text. The authors used CRFs with bigrams, lemmas, and words as features for the ATE task. While for the third task, ATP, they used an unsupervised approach based on syntactic dependencies and context-based polarity lexicons. The proposed approach for ACE got the highest result on the English and Spanish restaurants' datasets and the first place on ATE for the Spanish restaurants' dataset, while they got low results using the unsupervised approach for the ATP task. Moving to Jiang et al. (31) who participated in the ATP task and proposed a logistic regression model. They also proposed a new approach for extracting features from the given reviews. Since this task contains the aspect term and the review itself as input, the authors suggested extracting the features from the review fragments related to the given aspect term. This approach contains the following steps: the segmentation step, which splits the sentence into fragments, and the selection step selects one or more fragments from the given review sentence for each aspect. In the segmentation step, they used the punctuation marks and conjunctions to split the review into several parts or fragments, and then they chose the fragment that contained the target as the target fragment. The authors used a logistic regression model for their system and ranked above average among the submitted systems.

Another paper that used SVM to solve ABSA tasks is proposed by Kumar et al. (32). They ranked first in the ATP task for the laptops in the English language and the restaurants' for Spanish and Turkish. Moreover, they ranked first in the ATE task for the restaurants' domain in Dutch and French languages. This research uses an SVM classifier for ACE and ATP tasks; and CRF for ATE identification. They used thresholds ranging between 0.1 and 0.2 to predict if a review has a category or not. The authors used several features for the

ACE task, such as domain dependency graph, distributional thesaurus, Tf-IDF score, a BoW features. Moreover, for the ATE, they used word and local context, POS, headword with its POS, prefix with suffix, frequent aspect term, dependent relations, n-grams, orthographic feature, DT features, expansion score, chink information, lemma, WordNet, named entity information.

Many other papers were created and published after the SemEval 2016 task. The following is a summary of a few of them.

For ATP, the authors (33) conducted a comparison between SVM and Naive Bayes with the Gini Index method in the movie reviews domain. The dataset is from five websites: Bollywoodhungama, Rediff, Times of India, Rottentomatoes, and Mouthshut. The study is on two datasets: a large movie review dataset V1.0 with 94.46% accuracy and an extensive movie review data set SAR14 with 97.32% accuracy.

Mohammed et al. (34) compared the performance of two pre-trained models, fastText Arabic Wikipedia model and AraVec-Web model, using support vector machine for both ATE and ATP tasks. The experiments use Arabic airline customer service tweets; most tweets were in the Saudi dialect rather than MSA. Their results achieved an accuracy of 62% using AraVec, 70% using fastText in the ATE task, 86% using AraVec, and 89% using fastText in the ATP task.

Al-Smadi et al. (35) compared the performance of RNN and SVM models on the ABSA task. The SVM model uses several features, such as n-grams, semantic features, and syntactic features. For the RNN model, word2vec model was used to extract the embeddings so the model can use them as inputs. The performance of the SVM outperformed the performance of RNN in all subtasks. The results are SVM approach 93.4% F1-score for the first subtask, 89.8% F1-score for the second subtask, and 95.4% accuracy for the third subtask, whereas for the RNN approach, 48% F1-score for the first subtask, 48% F1-score for the second subtask, and 87% accuracy for the third subtask. On the other hand, (36) achieved high results in ABSA tasks without using deep learning methods. The suggested method outperformed the baseline by 53% in the first subtask, about 59% in the second subtask, and around 19% in the third subtask. The high result is justified by the high quality of the used features since the authors chose features that offer useful information to the classifier. The authors started by preprocessing the data to extract the features. First, they removed spacial and Latin characters and extracted the morphological, syntactic, and semantic features. Then, the authors used Weka (37) 2.7 to train a set of machine learning models as follows, Naïve Bayes, Bayes Networks, Decision Tree (J48), K-Nearest Neighbor, and Support-Vector Machine. The best model was SVM, and it achieved the following results: 93.4% F1-score in the first subtask, 89.8% in the second subtask, and 95.4% accuracy score in the third subtask.

Sana et al. (38) suggested a hybrid model for ATE and ATP tasks. The paper used the Arabic hotel reviews dataset from SemEval 2016. The hybrid model consists of features extraction, AdaBoost classifier, and linguistic method. The authors used MADAMERA to extract features such as numeric, lexical, semantic, and syntactic. Then they used AdaBoost classifier to detect aspects and the linguistic method to rectify the machine learning classifier's output and identify sentiments towards these aspects. The hybrid model obtains an accuracy of 97%. Caroline et al. (39) applied a robust deep syntactic parser combined with SVM to detect aspect term and category and their polarities. They achieved the highest results as the XRCE team in Semeval-2014 Task 4 in the restaurants' domain. They obtained 83% as F1-Measure for ATE and 82% for ACE, while the ATP model obtained an accuracy of 77% and 78% for aspect category polarity detection.

2.2 ABSA Using Unsupervised Methods

ABSA can also work with unsupervised approaches. Using supervised approaches does not seem to gain high results, but it removes the barrier of focusing on domain-specific tasks, and it can work with a more general type of data.

García-Pablos et al. (40) proposed a model that requires minimal supervision for multi-domain and multilingual ABSA. Their system leverages large quantities of unlabeled text and uses a minimal set of seed words, and it is based on a topic modeling approach with word embedding and a maximum entropy classifier. The authors compared their system with other LDA-based systems and achieved slightly higher results. Moreover, they outperformed the baseline models for ABSA 2016 task. Pavlopoulos et al. (19) published three new datasets for aspect-based sentiment classification; the published datasets are in the following domains, laptops, hotels, and restaurants. The authors proposed a new evaluation metric for ATE. They also showed how to improve an unsupervised ATE method using continuous space vector representations of words and phrases. The authors implemented four methods for the ATE task, the first method returns the most frequent distinct nouns (dubbed FREQ), the second method adds pruning mechanisms and extra steps to detect more aspects (dubbed H&L), the third method is an extension of the previous method, but it uses an extra pruning step (dubbed H&L + W2V). Finally, a similar extension of the FREQ is (dubbed FREQ+W2V). The (H&L + W2V) method got the highest results. Results were reported in average weighted precision and were as follows, 66.8% for restaurants' data, 53.3% for hotel data, and 38.9% for laptops data.

2.3 ABSA Using Deep Learning

Deep learning approaches are typically more powerful than traditional machine learning models; thus, many researchers try to address the various ABSA challenges using various deep learning algorithms such as CNNs, LSTMs, and others. Using a deep learning approach, one of the researchers ranked first on the English datasets for the SemEval-2016 tasks ACE and ATE. TOG et al. (41) used various lexicon features, syntactic features, cluster features, and features that were extracted using a deep learning approach. The proposed method consists of multiple single-layer feed-forward networks where each network works as a binary classifier that only detects one aspect. While for the second task, they used CRF to train sequential labeling classifiers. Some reviews contained more than one aspect, so the authors used a threshold to detect these aspects and their categories. So far, the previous papers have talked about using deep learning models in various ways to solve the ABSA tasks, but the following paper adds a CRF layer to the neural network to take advantage of the power of CRF layers. Wenya et al. (42) proposed a model that integrates neural networks and CRF to extract the aspect and opinion terms. Using both the restaurants' and the laptops' datasets from Semeval competition, the authors compared their model with several baseline models such as CRF, LSTM, LSTM+ hand-crafted features, WDEmb+B+CRF model proposed by Yin et al. (43), and more. The proposed model outperforms the baseline models. For ATE, the model achieves an 84.93% F1-score in the restaurants' dataset and a 78.42% F1-score in the laptops dataset. At the same time, the ATP achieves an 84.11% F1-score for the restaurants' dataset and a 79.44% F1-score for the laptops dataset.

Several researchers applied the LSTM models for this task. Ruder et al. (44) used a hierarchical bidirectional Long Short-Term Memory (BLSTM) architecture to model the interdependencies of sentences in a review in ABSA task. The authors tested their proposed model on different datasets in English, Spanish, French, Russian, Dutch, Turkish, Arabic, and Chinese. Although their hierarchical BLSTM model uses only pre-trained embedding as features, it outperformed models that relied on heavy feature engineering or used enormous external resources on five datasets. Y Ma et al. (45) added the attention mechanism to their work in order to develop a more advanced model. They developed an attention-based neural architecture model for two subtasks of ABSA: ACE and ATP. Their model adapted an LSTM model called sentic LSTM with target-level attention and sentence-level attention. Sentic LSTM achieved good results on both subtasks by evaluating the model on SentiHood (46) and a Semeval subtask in 2015 (47). Furthermore, Al-Smadi et al. (48) proposed a BLSTM approach solving two subtasks of ABSA task (49). For the first subtask, the authors used character-level BLSTM along with a CRF classifier (BLSTM-CRF). Using this method, they overcame the baseline's result by 39% for the ATE subtask. For the second subtask, they used aspect-based LSTM in which the aspect is considered attention expressions to support the ATP model, and this method achieved a 6% increase to the baseline result for the ATP subtask.

As we said earlier, researchers used different deep learning models such as NN, LSTMs. However, other researchers used the well-known CNN models known for their power and usability with image processing. Wei Xue et al. (50) applied convolutional neural networks (CNN) with a gating mechanism on SemEval datasets. They used the restaurants' and laptops' domains to address both the aspect-category SA and the aspect-term SA problems. This implementation is more simple than the RNN based models and also allows parallelized processing. On the aspect-term task, the gated convolutional network with the aspect embedding model achieved 77.28% accuracy on the restaurants' dataset and 69.14% accuracy on the laptops dataset. Ishaq et al. (51) proposed an ABSA method by tuning the CNN using a genetic algorithm. Using extracted semantic features and Word2Vec representation for three datasets (hotel, automobiles, and movie reviews), 95.5%, accuracy was achieved. INSIGHT-1 got first place in the SemEval-2016 Task-5 competition on the Arabic hotel reviews dataset. The authors concatenated aspect embedding with every word embedding and fed the output to CNN for both ATP and ACE tasks (52). Bo Wang et al. (53) used a neural network approach for ACE by leveraging a constituency parse tree and a CNN model. They implemented a model consisting of a two-layer neural network for ACE based on the constituency parse tree and a CNN model for ATP. The authors developed the model on laptops' and restaurants' domains using the SemEval-2015 Task 12 dataset. The combined model achieved decent results compared with other deep learning models in that period: 51.3% F1-score for the ACE task and 79.3% accuracy for the ATP task.

2.4 ABSA Using BERT

With the introduction of BERT to the NLP world, many research papers started leveraging the power of BERT when analyzing text datasets. ABSA researchers recently started utilizing BERT with their ABSA models; this comprehensive model achieved top results in ABSA tasks.

Xu et al. (54) proposed a post-training method to adjust the weights of BERT to make it better for specific domains and tasks. The authors used the BERT model and post-trained it on Amazon laptops reviews (55) and Yelp Dataset Challenge reviews (56) to increase its performance for laptops-related tasks. For the restaurants' domain, they used Yelp reviews from restaurants' categories. The results show decent improvement using this new approach. The authors compared the BERT model with and without post-training; the post-trained BERT model improved the result by 4.98% for ATE in the laptops domain and 3.87% for ATE in the restaurants' domain. While for ATP, the improvements are 2.78% for the laptops domain and 3.41% for the restaurants' domain.

Mickel and Oskar (57) proposed three models: aspect classification model based on sentence pair classification from BERT, to check the relatedness between the aspect and the text, and then output a label that indicates whether it is related or unrelated. ATP model to predict the sentiment label of a text-based on the aspect. Furthermore, a combined model to predict the aspects and their sentiments. They evaluated the models using SemEval-2016 tasks, and the evaluation shows that the combined model outperforms previous state-of-the-art results for aspect-based sentiment classification.

Li et al. (58) implement a model for End-to-End ABSA, they add E2E-ABSA layer, which is a neural layer on the top of BERT representation. They implement different neural layers to be the E2E-ABSA layers, such as a linear, RNN, self-attention, and CRF. The experimental results approve that BERT + GRU achieves the highest result of 61.12% F1-score.

Jafarian et al. (27) presented a model using BERT for ABSA on the Pars-ABSA dataset in the Persian language. The model uses multilingual pre-trained BERT that supports the Persian language and receives input that consists of two sentences; the review sentiment and the aspect terms as auxiliary sentences. The authors experimented with four types of auxiliary sentences; the best one was the NLI-M auxiliary sentence. The model outperforms other models for an ABSA task in the Persian language. It got an accuracy of 91%.

Junqi et al. (59) compared the induced trees from different pre-trained embedding such as BERT and RoBERTa with several models for aspect-level sentiment classification tasks. Based on their experiments, the best model was using induced trees from fine-tuned RoBERTa with MLP. the experiments evaluated six benchmark datasets, two of them from SemEval 2014 task 4 on restaurants' and laptops' domains. The results reported inaccuracy score as follows, 87.52% for restaurants' data and 84.16% for laptops data.

Muhamad et al. (20) adapted BERT for ATE for tourist reviews in the Indonesian language. They collect data from TripAdvisor to be used in retraining and fine-tuning the multilingual uncased BERT model. They experimented with different scenarios, with and without using the preprocessing phase. The best model was without preprocessing, and it achieved the best accuracy and F1-measure score with 79.9% and 73.8%, respectively.

Youwei et al. (21) presented a new method to utilize the intermediate layers of BERT in order to improve the performance of fine-tuning BERT to solve ABSA tasks. Their method starts by designing two pooling strategies to connect the intermediate layers' representation and using this pooling module to fine-tune BERT. The authors evaluated the model through three datasets from Semeval-2014, and the results are as follows: BERT-PT-LSTM achieves 85.29% accuracy score on restaurants' data, BERT-PT-Attention achieves 77.68% accuracy score on laptops data, and BERT-Attention achieves 73.35% accuracy score on Twitter data.

Hu et al. (22) investigate the inner working of masked language modeling (MLM) such as BERT for ABSA tasks. They analyze MLM through self-attention of aspects and hidden representations on aspects and found that the BERT model focuses on the semantic knowledge of the review domain, not the opinion. So they concluded that MLM models are suitable for ATE but still need more work for ABSA that depends on the opinion.

Akbar et al. (23) proposed a new methodology using a deep learning model for ATE and ATP. They applied parallel and hierarchical aggregation modules to utilize BERT layers to capture more profound knowledge of the input sequences. It works on computing the average loss value for the prediction of selected hidden layers for both modules.

Chapter Three: Background

This chapter talks about the concepts and models used in this research. It starts with self-training first, and then the embeddings are explained, followed by a brief introduction to deep learning models and an overview of each model applied in this study. It also describes transformer-based models like BERT and then goes over the evaluation metrics and evaluation methods used for this task.

3.1 Self-training

Instead of manually labeling a vast dataset and training a machine or deep learning model, a self-training approach can make things more manageable. Self-training suggests using labeled data to train a model, labeling unlabeled data, and then using both the labeled data and the new data with its new predicted labels to re-train the model. Figure 1 describes a high overview of how the self-training process works. It starts by using a labeled dataset to train a model, then uses the trained model to generate pseudo labels for unlabeled data. After that, it combines both labeled and pseudo labeled datasets and re-trains the model. Self-training increases the flexibility of a project, and it works when there is a low data regime, high data regime, weak data augmentation, or even strong data augmentation.

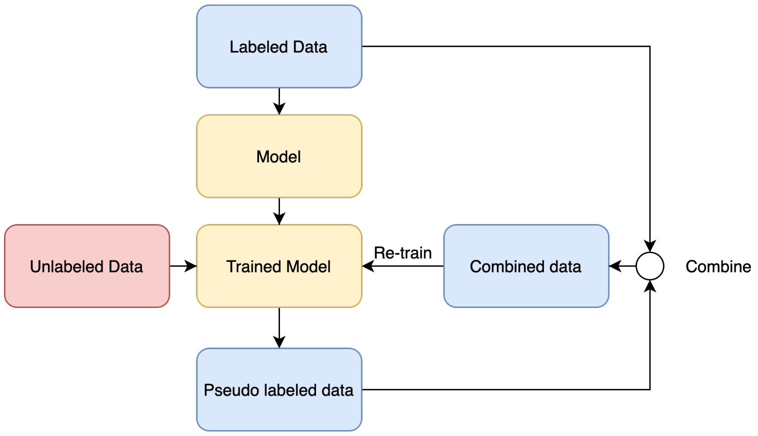
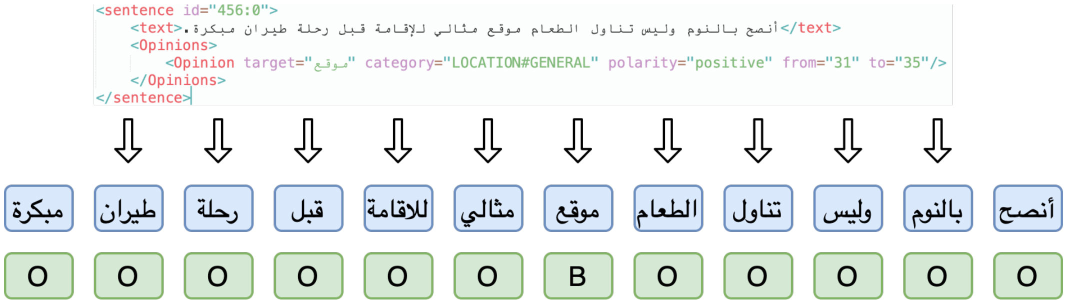


Figure 1: A high overview of how the self-training process works

3.2 Inside–outside–beginning

The Inside–outside–beginning (IOB) format is a common tagging format for tagging tokens in a text. It is commonly used with many NLP tasks such as named entity recognition, part of speech tagging and more. An example of how IOB tagging works with the Arabic hotel reviews shown in Figure 2



```
<sentence id="456:0">
    <text>أنصح بالنوم وليس تناول الطعام موقع مثالي لإقامة قبل رحلة طيران مبكرة.</text>
    <Opinions>
        <Opinion target="موقع" category="LOCATION#GENERAL" polarity="positive" from="31" to="35"/>
    </Opinions>
</sentence>
```

Figure 2: An example of how IOB tags are generated

3.3 Conditional Random Field

CRF (60) is a class of discriminative models. Usually, researchers use CRF in prediction tasks; it enables the model to make an isolated prediction without the effect of the neighboring" samples. Moreover, CRF uses contextual information to make better predictions.

CRF deals with many applications that have structured data (61), often used in NLP, such as Parts-of-Speech tagging, and other domains such as parts-recognition in Images and gene prediction.

3.4 Pre-trained embeddings

Researchers used features like TF-IDF, n-grams, and a bag of words to represent the text input in early NLP use cases. However, nowadays, researchers use more sophisticated ways to represent the text used for NLP tasks. One of the methods researchers use is word embeddings. Researchers first used word embeddings in 2010 after it showed an excellent performance and training speed. In 2013, Google introduced the first pre-trained word embeddings, word2vec (62). After that, word embeddings became more popular and widely investigated by researchers. Word embeddings convert the text to a vector of numbers that has meaningful information such as semantic and syntactic information 4. These vectors can capture information between related words. For example, subtract the vector representation for the word "Man" from the word representation for the word "King," the output will be a vector that represents the subtraction of vector representation for word "Woman" from the word representation for the word "Queen" as shown in 3.

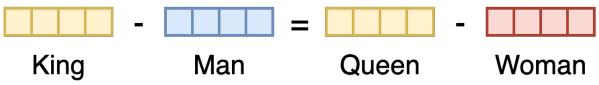


Figure 3: Mathematics operations on word representations generated by word embeddings provide meaningful information

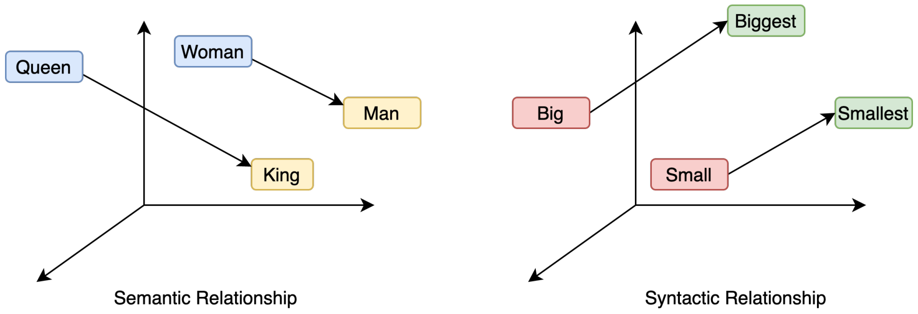


Figure 4: Semantic and Syntactic relationship in word representations

3.4.1 FastText

Word embeddings capture the hidden information from the text; for example, it can extract the semantic and analogies features from the given text (63, 64). These features can be helpful for text classification tasks. fastText is an open-source library created by Facebook's AI Research lab in 2015 (65, 66). It learns word embedding and representation, and also it can work with classification tasks. On the fastText website, there are publicly available pre-trained word embeddings for 157 languages, and these pre-trained word embeddings have to ability to extract the features for a given text to feed the features to the classification model. Fasttext is an extension of the well-known word2vec model. However, it differs in the way it extracts the vector representation for each word than word2vec. Fasttext uses n-grams to split the word into smaller chunks and then trains a skip-grams model to learn the embedding. This helps the model to understand prefixes and suffixes as well as understand smaller words. Fasttext outperforms other pre-trained word embeddings like Glove (67) and word2vec because it breaks down the word into smaller pieces and then finds the representation for that word, whereas word2vec and Glove cannot generate the vector representation for words that the model did not see before.

3.5 Deep Learning

Deep learning is a part of the big machine learning group, and it is divided into supervised, semi-supervised, and unsupervised. The first working algorithm for supervised, deep, feedforward, multilayer perceptrons was in 1967. Deep learning did not gain much attention back then due to a lack of resources. However, nowadays, with resources not being much of a problem, many researchers widely use deep learning to solve different tasks such as NLP tasks, computer vision tasks, speech recognition, and more. Deep learning architectures can be Recurrent Neural Network (68), Convolutional Neural Network (CNN) (69, 70), Deep Neural Network.

3.5.1 Convolutional Neural Network

Generally, CNN performs better than a feedforward neural network for many reasons regarding image-related tasks. Feedforward neural networks require the image/input to be flat before feeding it into the network, which means much of the critical information such as spatial information will be lost when flattening an image to a one-dimensional array. Another problem that might arise when using a feedforward neural network is the number of dense layers in the model, leading to overfitting problems. So instead of using a feedforward neural network, a CNN model can be used to deal with such tasks. CNN takes the image as input without flattening it, which means it can leverage the spatial information from an image and take much less computation time than feedforward neural networks because it applies parameters and weights sharing.

CNN can also be used to process text. Many researchers used CNN's with text-based tasks in a wide range of research papers (71, 72, 73, 74). The primary process of CNN is convolution; instead of taking the whole picture, the image is divided into parts and analyzed. It is sparsely connected because it is not related to all pixels but the respective pixel. The convolution process extracts the feature mapping, and then the filter is trained to recognize specific features within the input data; it is possible to apply more than one filter on all layers. Applying more filters helps to extract more features from the image. After each convolution process, a pooling layer works to reduce the size of the image, with a focus on specific things, as required. The features are extracted from the image; then, it is fed to a flatting layer, converting it from 2d to 1d. Finally, it enters the fully connected layer and uses one of the activation functions with the desired prediction. CNN problems need a considerable amount of RAM during the training process (75).

Figure 5 presents the complete picture of a CNN architecture. It contains a convolutional layer, a pooling layer, and a fully connected layer, and these layers work together as described below.

- Convolution Layer: This is the core of the CNN architecture. This layer performs a dot product between kernels (learnable parameters) with parts of the input leading to a more miniature representation. During training, the kernel moves in a convolutional motion and generates a representation for each part of the input. The final output of this kernel is a two-dimensional array called an activation map. The number of steps a kernel moves while going over the input is called strides, and you can modify it as required.

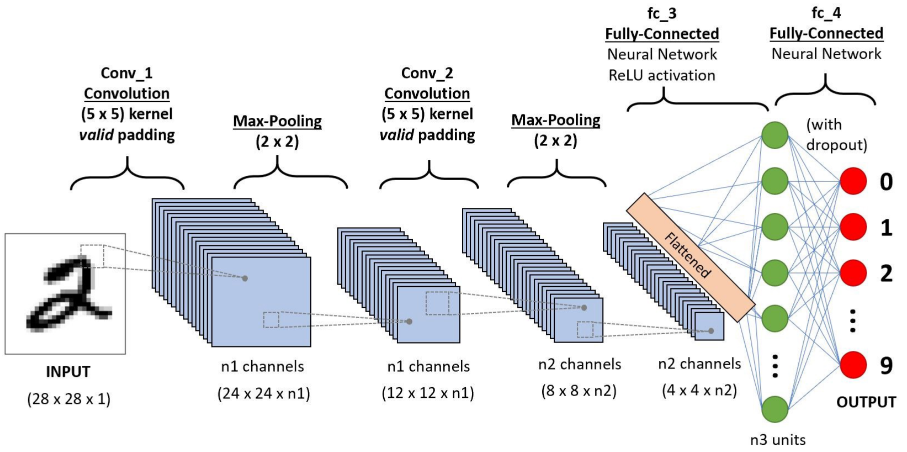


Figure 5: The CNN architecture for handwriting digits classification

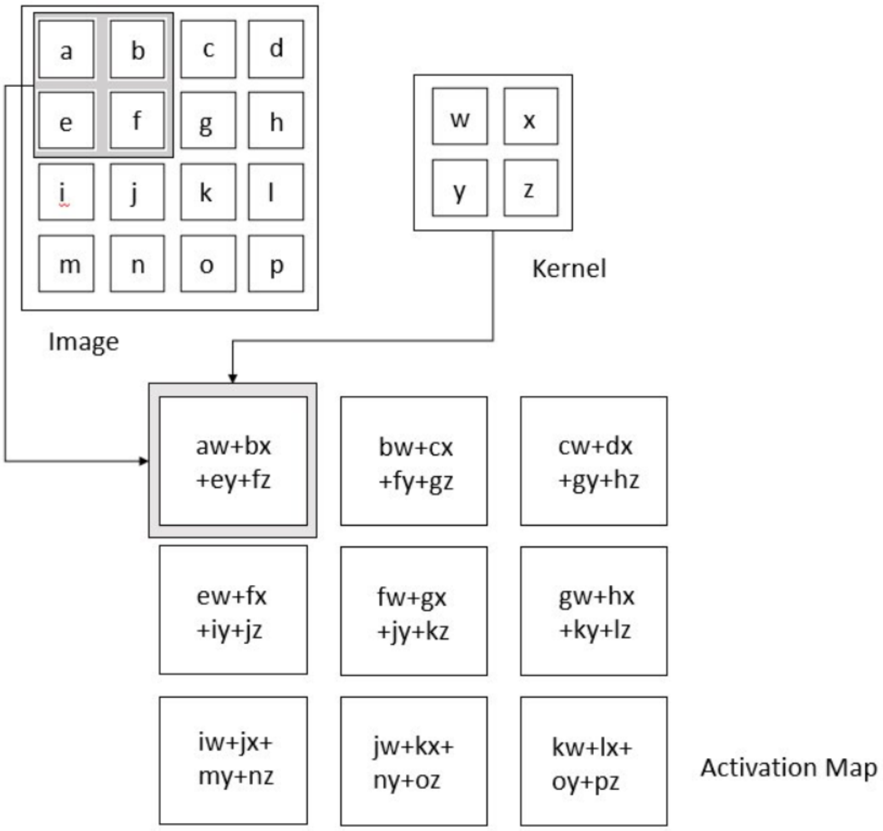


Figure 6: Convolutional operation (76)

- Pooling Layer: The primary function of a pooling layer is to reduce the size of each feature map, which results in fewer parameters and fewer weights to train. There are different ways to apply pooling on the feature map, such as max pooling, average pooling. Max pooling searches and takes the maximum value of each filter size area in the feature map, while average pooling calculates the average value of each filter size area in the feature map.

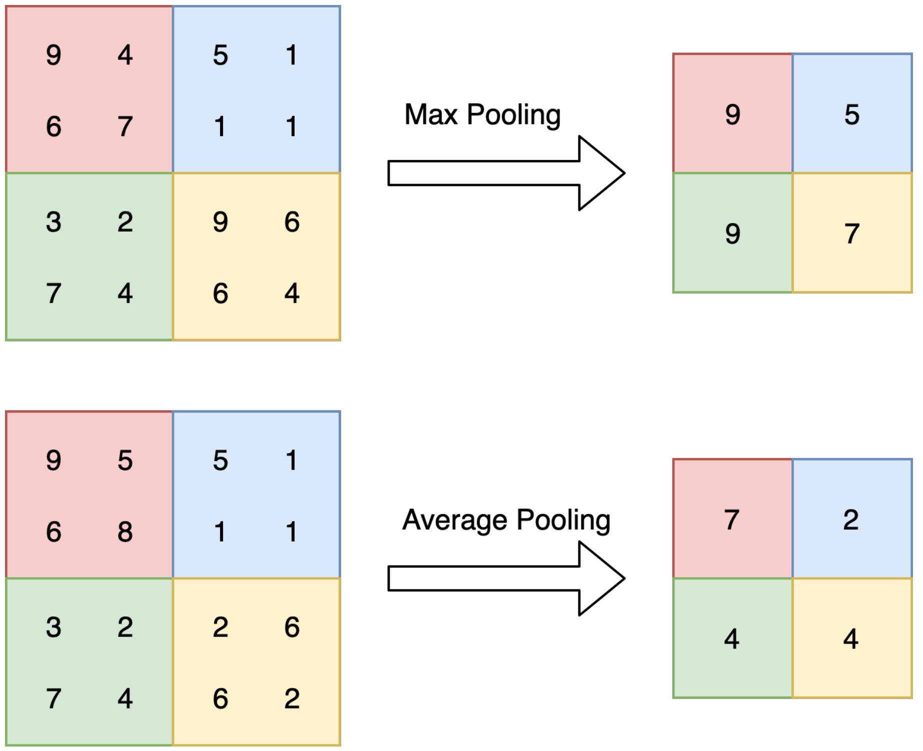


Figure 7: Example of max and average pooling

- Fully Connected Layer: This is the final layer of CNN architecture. The model first flattened the input before entering this layer using a flatten layer, converting the data into a one-dimensional array.

3.5.2 Long Short Term Memory

Long Short-Term Memory (LSTM) networks are a form of RNN architecture that overcomes the vanishing gradient problem and allows the model to learn long-term dependencies. For example, the network can learn that it should delete this data and keep other data. The LSTM may delete or add information to the cell state using structures known as gates. Gates are a method of optionally allowing information to pass through. They contain a sigmoid neural network layer and a pointwise multiplication operation.

The sigmoid layer produces values ranging from zero to one, indicating how much each component should pass. The first step is to select what information will be discarded from the cell state. This choice is determined by a sigmoid layer known as the forget gate layer F_t; it takes a look at H_{t-1}. and X_t. Then it gives an output that ranges between 0 and 1 for each number in the cell state C_{t-1}. The number 1 indicates that something should be kept, whereas the number 0 indicates that it should be discarded.

$$f_t = \sigma(W_f[H_{t-1}, X_t] + B_f) \tag{3.1}$$

Then decide what new information will be stored in the cell state. This has two parts:

- Input gate layer: a sigmoid layer decides which values will be updated.
- tanh layer: this layer creates a vector of new candidate values, C_t that could be added to the state.

The input gate i_t and tanh layer will then be merged to generate a state update. The LSTM then changes the old cell state C_{t-1} into the new cell state C_t by multiplying the old state by the forget gate layer, allowing it to forget the items that were previously determined to forget.

$$i_t = \sigma(W_i[H_{t-1}, X_t] + B_i) \tag{3.2}$$

A sigmoid layer merged with the tanh layer decides what parts of the cell state are going to pass via the output gate i_o.

$$i_o = \sigma(W_o[H_{t-1}, X_t] + B_o) \tag{3.3}$$

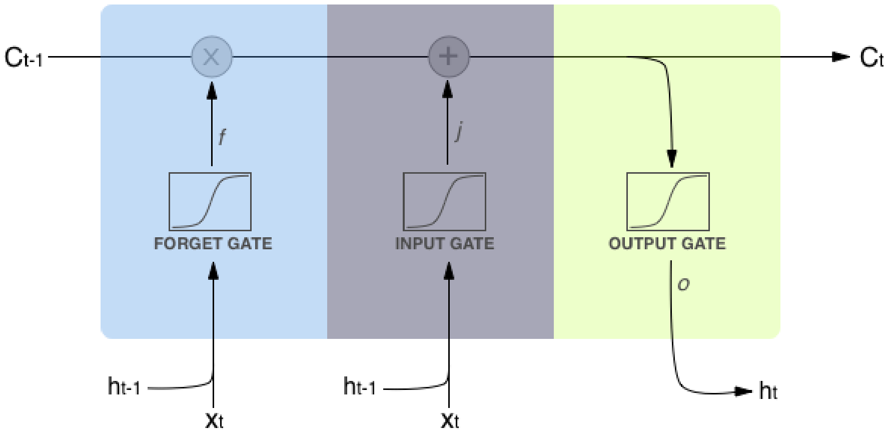


Figure 8: The LSTM cell from the inside.

3.5.3 Gated Recurrent Unit

Gated Recurrent Unit (GRU) was first introduced in 2014 by (77). Like the LSTMs, GRU is a type of RNN and solves the vanishing gradient problem with RNN models. GRU is a lighter version of LSTM. It has two gates only, a rest gate and an update gate. It is faster than LSTM since it has fewer parameters than LSTM. GRU uses the update gate and the reset gate to decide the important information from the previous input that should be remembered and what information should be removed. These gates allow the GRU to remember related information during training and forget irrelevant information.

The formula for update gate is the following:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \tag{3.4}$$

It takes X_t and multiplies it by its weight W_z. It does this also to the information for the previous $t-1$ units, h_{t-1} is multiplied by its weight U_z. Next, the results are added together and fed to a sigmoid function. The update gate decides how much previous information will pass through the gate and be considered in the following processing.

The second gate (Reset gate) is responsible for determining which information should be forgotten from the previous input. The formula for this gate is as following:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \tag{3.5}$$

This gate acts similar to the update gate. It takes the input and previous input and multiplies them with their corresponding weights. Then the results are summed and fed to a sigmoid function.

The reset gate is used to calculate the new memory content, it picks the irrelevant part to be forgotten and keeps the important parts to be stored in the memory. The new memory content is calculated as follows:

$$h\prime_t = tanh(Wx_t + r_t \cdot Uh_{t-1}) \tag{3.6}$$

Finally, the last memory at the current time step is computed. The update gate does this. The update gate is used to calculate the h_t vector that includes the information for the current unit and transfer it to the next unit. It decides what to collect from the present memory content $h\prime_t$ and what to collect from the previous steps h_{t-1}. This is done as follows:

$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot h\prime_t \tag{3.7}$$

3.6 Transformers

Transformers is a novel neural network architecture that was produced in 2017 by (78). Transformers proved their effectiveness in several NLP tasks. It consists of an encoder and a decoder. Both the encoder and the decoder are composed of modules that can be stacked on top of each other; the Nx in the figure 13 describes that. The encoder consists of a set of

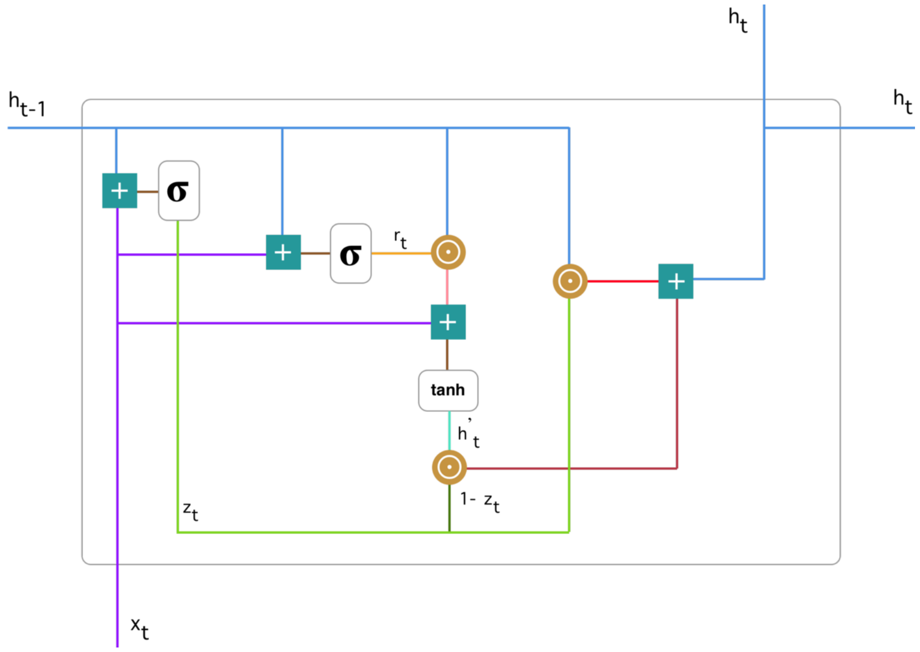


Figure 9: The GRU cell from the inside.

encoding layers that processes the input iteratively, one layer after another, and the decoder consists of a set of decoding layers that does the same thing to the output of the encoder. Both the encoder and decoder use the attention mechanism, which gives weight to the input while being processed inside the encoder and decoder.

The attention mechanism has proven its effectiveness and success in various models in recent years. Attention is a powerful mechanism that was developed and proposed in 2014 by (79); it was developed to improve the performance of the encoder decoder-based neural machine translation system. While the proposed encoder-decoder model (77) improved the neural machine translation and sequence-to-sequence prediction in general, it still has drawbacks. For example, (80) stated that when increasing the input sentence length, the performance of the encoder-decoder model degrades rapidly. On the other hand, the attention mechanism solves the long input sentence problem by listing the most important words from the input each time the model processes an output word. In other words, adding the attention layer between the encoder and decoder helps the model focus only on essential words each time the decoder processes a new word; this method helps the model handle long input sentences effectively.

There are two types of attention, General Attention which is between the input and output elements, and Self-Attention, which is used Within the input elements.

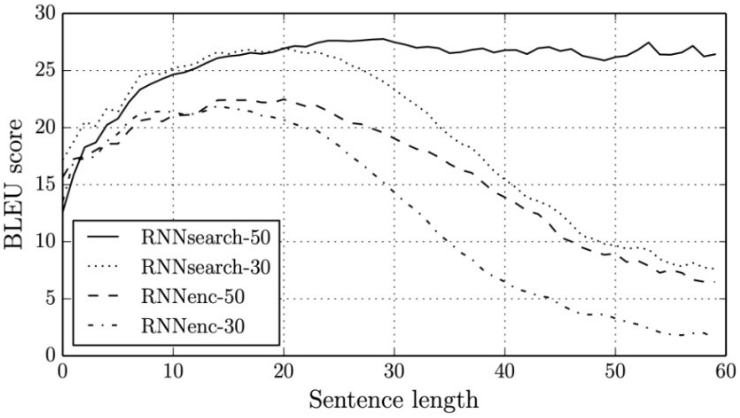


Figure 10: A comparison by (79) between encoder-decoder architectures with and without attention mechanism. The scores used are the BLEU scores of the generated translations. RNNsearch-50 represents a model trained on sentences of length up to 50 words using the attention mechanism.

3.6.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) was proposed by (81) in 2018, and it started a new era in the NLP world; BERT presented state-of-the-art results in a wide variety of NLP tasks. BERT can be used with a wide range of tasks such as question answering (82, 83), summarization (84, 85), natural language inference (86), sentiment classification (87), word similarity (88), and text classification (89).

BERT is based on transformers architecture, which consists of stacked encoders. BERT framework provides two stages: pre-training and fine-tuning. In the pre-training stage, the model is training on a vast unlabeled dataset, it uses plain text corpus such as Wikipedia dumps, and it learns from it by masking words in a sentence and predicting the masked words. This method serves the purpose of teaching the BERT model to understand the language. However, to utilize BERT to its full potential, the model can be fine-tuned by adding an extra layer to train it on a specific task.

BERT takes a pair of sentences as input and adds two special tokens to them; a [CLS] token at the beginning of the sentence and a [SEP] token between sentences, and then it uses a wordpiece tokenizer to split the sentences into tokens. For example, the input "my dog is cute" and "he likes playing" will be converted to "[CLS] my dog is cute [SEP] he likes play ##ing [SEP]" as shown in figure 11. The wordpiece tokenizer generates the # symbol, this is done when splitting the input sentence into tokens, and it helps with handling out-of-vocabulary words and limiting the number of vocabulary.

BERT further transforms the input tokens into three embeddings: position embedding, segment embedding, and token embedding. The position embedding is used to tell the position of each word in the sentence, which helps BERT understand and utilize the words order information. The second representation is segment embeddings when BERT takes two sentences as input; it learns a unique embedding for each sentence in the input to help the model differentiate between them, as shown in Figure 11 all tokens from the first sentence will have the representation EA. In contrast, all the tokens from the second sentence will have the representation EB. The final representation is token embedding; this is similar to word embeddings generated by pre-trained word embeddings like word2vec and fasttext. However, BERT generates embeddings affected by the context, which gives richer information to the representation generated for each word. Finally, all three representations are summed to form the input embeddings.

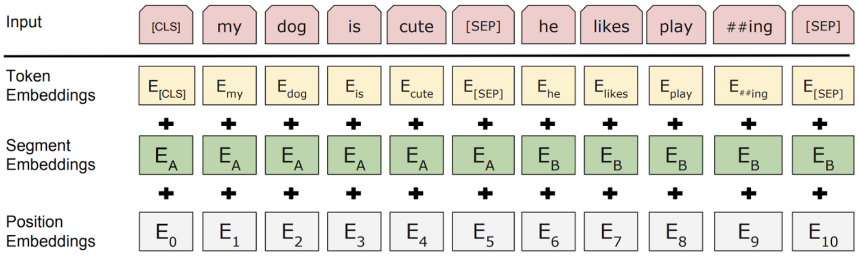

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 11: BERT's inputs (81)

BERT learns from unlabeled text using two methods: MLM and Next Sentence Prediction (NSP). The principle underlying MLM is simple: given a large text, BERT substitutes 15% of the tokens with [MASK] tokens and predicts it using the context around the masked words, as illustrated in 12. While for NSP, BERT uses NSP to learn the connection between the two input sentences by predicting whether the second sentence comes after the first sentence or not.

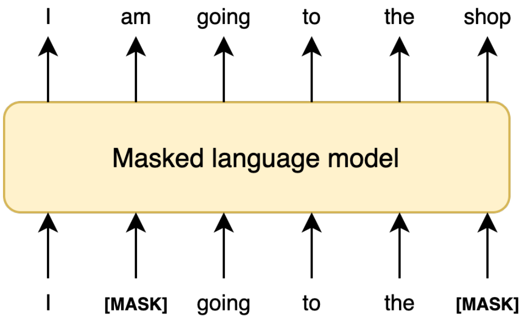


Figure 12: BERT using masked words to train and understand the language

3.6.2 CAMeLBERT

CAMeLBERT is a collection of pre-trained models for Arabic NLP tasks (90). The authors trained four BERT models on MSA, dialects, classical, and a mix of all. The authors evaluated the models on five NLP tasks: NER, POS tagging, SA, dialect identification, and poetry classification.

3.6.3 AraBERT

AraBERT is an Arabic pre-trained language model based on Google's BERT architecture (91). AraBERT uses the same BERT-Base config. The pre-training data used for the new AraBERT model is also used for Arabic GPT2 and ELECTRA. The dataset that used to train AraBERT consists of 77GB of text, which are 200,095,961 lines or 8,655,948,860 words. The model was evaluated on SA using 6 different datasets (HARD (92), ASTD-Balanced (93), ArsenTD-Lev (94), LABR(95)), Named Entity Recognition with the ANERcorp (96), and Arabic Question Answering on Arabic-SQuAD and ARCD (97).

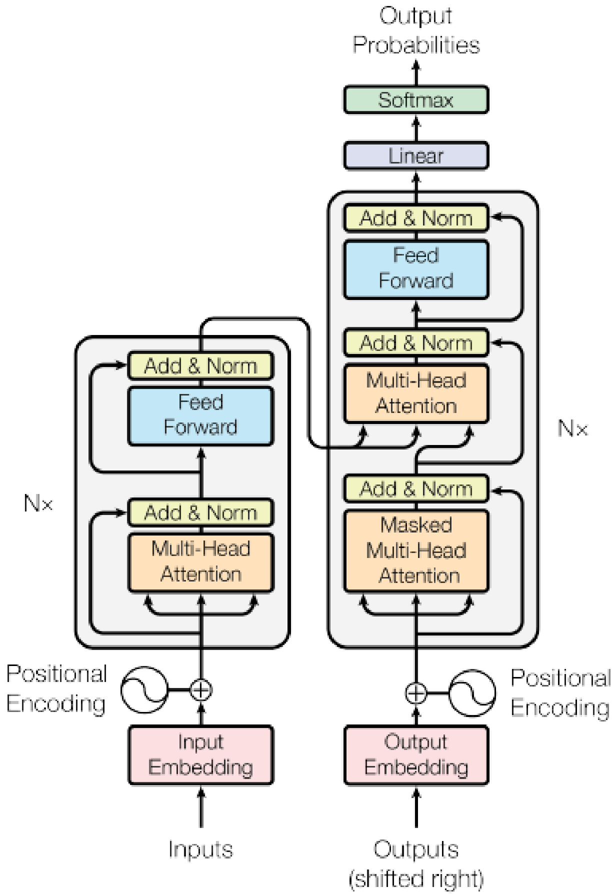


Figure 13: The Transformer architecture by "Attention is all you need" (78)

3.7 Evaluation Metrics

Evaluation of the model is an essential part of any project; it shows if the model is good or not. One of the most used metrics is the accuracy_score metric. It is an excellent metric to evaluate the model, but sometimes using the accuracy_score metric alone is not enough, so this research uses other metrics such as F1_score, Precision, and Recall.

3.7.1 F1-Score

F-Measure combines Precision and Recall into one measurement tool; it uses the harmonic mean to combine them. The Equation 3.8 to calculate this measure.

$$F1 = 2 * \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{3.8}$$

Precision is one of the known measures, and it measures the classifier's ability to return relevant instances only. The equation used to calculate the precision is Equation 3.9, the number of correct positive results is divided by the number of the positive results predicted by the algorithm.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{3.9}$$

Recall (also known as sensitivity) measures the classifier's ability to identify all relevant instances. The equation used to calculate the recall is Equation 3.10, the number of correct positive results is divided by the number of all relevant samples.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{3.10}$$

3.7.2 Accuracy

Accuracy is the most popular performance measure used; it is the ratio of correctly predicted observations to the total observations. Accuracy would evaluate the model well only if the dataset was balanced. Since our data is not balanced, other measurement tools were used to evaluate the model. Equation 3.11 to calculate this measure, which is the same as Equation 3.12.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{3.11}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.12}$$

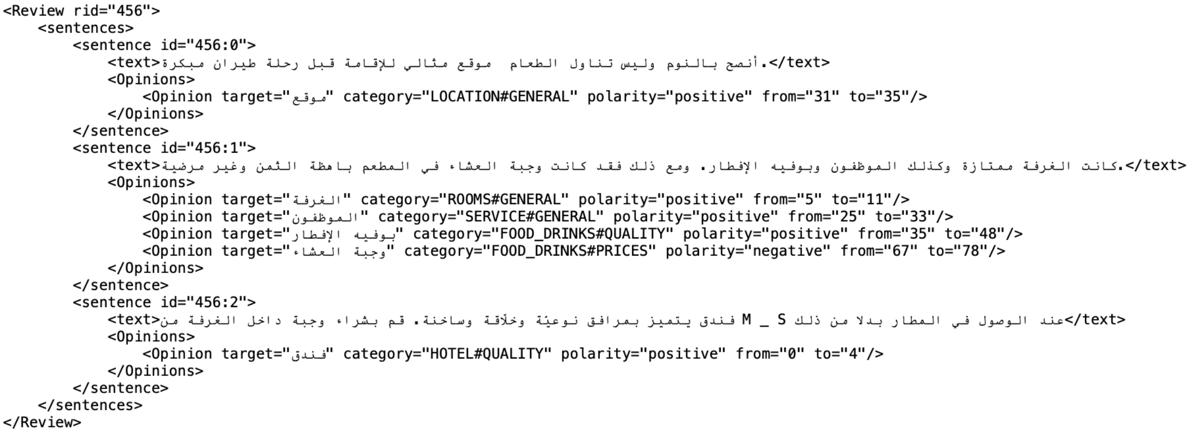
Chapter Four: Methodology

This chapter will explain the dataset used, and the methodology followed to create the proposed AHR-BERT model. It also presents different approaches using deep learning.

4.1 Data Description

The dataset used in this research is from ABSA within the Semantic Evaluation Workshop 2016. That has been prepared by (98) to support the Arabic track of Task5. The dataset is a subset of the Hotels' reviews collected in (99).

Our methodology is evaluated using ABSA-Hotels dataset (98), from SemEval-2016 task 5: ABSA. ABSA-Hotels contains 24,028 annotated tuples on both text-level (2,291 reviews' texts) and sentence-level (6,029 annotated sentences). The data is from famous hotel booking websites like Booking.com and TripAdvisor.com. The reviews are from hotels in different Arabian cities such as Dubai, Mecca, Amman, Beirut, and others. ABSA-Hotels is based on (99) dataset. An example of the data is shown in figure 14.



```
<Review rid="456">
    <sentences>
        <sentence id="456:0">
            <text>أنصح بالنوم وليس تناول الطعام  موقع مثالي للإقامة قبل رحلة طيران مبكرة.</text>
            <Opinions>
                <Opinion target="موقع" category="LOCATION#GENERAL" polarity="positive" from="31" to="35"/>
            </Opinions>
        </sentence>
        <sentence id="456:1">
            <text>كانت الغرفة ممتازة وكذلك الموظفون وبوفيه الإفطار. ومع ذلك فقد كانت وجبة العشاء في المطعم باهظة الثمن وغير مرضية</text>
            <Opinions>
                <Opinion target="الغرفة" category="ROOMS#GENERAL" polarity="positive" from="5" to="11"/>
                <Opinion target="الموظفون" category="SERVICE#GENERAL" polarity="positive" from="25" to="33"/>
                <Opinion target="بوفيه الإفطار" category="FOOD_DRINKS#QUALITY" polarity="positive" from="35" to="48"/>
                <Opinion target="وجبة العشاء" category="FOOD_DRINKS#PRICES" polarity="negative" from="67" to="78"/>
            </Opinions>
        </sentence>
        <sentence id="456:2">
            <text>عند الوصول في المطار بدلا من ذلك S _ M فندق يتميز بمرافق نوعيّة وخلّاقة وساخنة. قم بشراء وجبة داخل الغرفة من</text>
            <Opinions>
                <Opinion target="فندق" category="HOTEL#QUALITY" polarity="positive" from="0" to="4"/>
            </Opinions>
        </sentence>
    </sentences>
</Review>
```

Figure 14: Example from the dataset in xml format

This research considered sentence-level reviews only; the dataset has a total of 6028 Arabic hotel reviews, 4802 reviews for training, and 1226 for testing. On average, each review contains a couple of aspects, some reviews have up to nine aspects, and some have zero aspects. There are 6197 positive aspects, 683 neutral aspects, and 3629 negative aspects for the training dataset. There are 1508 positive aspects, 169 neutral aspects, and 924 negative aspects for the testing dataset. Figure 15 summarizes the data numbers.

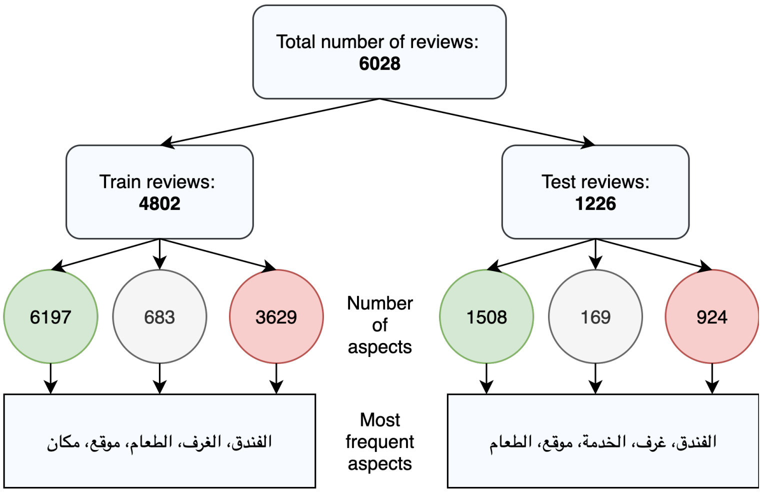


Figure 15: Data analysis for Arabic hotel reviews, the color representations are the following, Green: Positive, Light gray: Neutral , Red: Negative

4.2 Deep Learning models

This section will describe the deep learning systems used for aspect term extraction and aspect term polarity extraction. Compared to other research papers that used deep learning models, our aspect term polarity extraction model outperforms the existing models proposed by the research papers using the same dataset; however, our aspect term extraction model does worse than the proposed models from research papers on the same dataset.

4.2.1 Bi-GRU Model

A Bi-GRU model was used for the aspect extraction task; it extracts both word and character embeddings from the input and uses a CRF layer to find the correct tag for each input word. Usually, researchers use word embeddings only in such tasks, but this produces unknown words if they are not available in the pre-trained word embeddings like fasttext. To solve this problem, character embeddings that eliminate the unknown words from the inputs were used. Then padded the length to a fixed size for each character sequence using a padding character (token). Since there are no pre-trained embeddings for characters, we used the Keras embedding layer to map each character into an embedding space. After that, the character embeddings are fed to a 1D-convolution using a kernel of size 5 and 30 convolution filters. Finally, we fed the output of the 1D-convolution layer to a max-pooling layer and flattened the output to make it mergeable with the word embeddings that were extracted before. After concatenating both word and character embeddings, a Bi-GRU is used to process the features in both directions and then feed the output to a CRF layer that produces a tag for each input word, B-A, I-A, or O tag. The parameters used for this model are shown in table 1

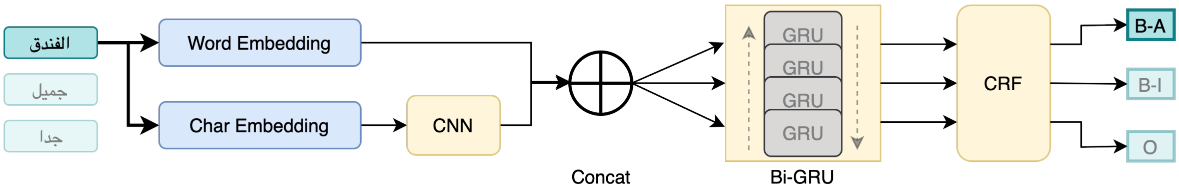


Figure 16: The system used for aspect extraction subtask using a deep learning model

Parameter	Value
GRU_Layer	100
Learning_Rate	0.001
Loss_Function	crf loss_function
Word_Embedding_output_dim	300
Char_Embedding_output_dim	25
Kernal	5
Filters	30
Optimizer Parameters	
Optimizer	RMSprop
rho	0.9
epsilon	1e-07
clipnorm	5.0

Table 1: Parameters used for aspect term extraction model

4.2.2 LSTM-Attention Model

For the aspect term polarity extraction task, the goal is to find the polarity of each aspect in the sentence. Each aspect could be positive, neutral, or negative. In this subtask, an LSTM with an attention model is used. Following the work of (48, 100), this model takes the embeddings for the input sentence and the aspect word at the same time to highlight the part of the input sentence that the model should focus on. For example, there could be more than one aspect in an input sentence, like “The hotel was nice but the food we ate yesterday was not good at all.” This input sentence is turned into two training sentences as shown in figure 17; the first one takes the aspect “hotel” and gives it a positive label, while the second input sentence takes “food” and labels it with negative. The rest of the model is illustrated in the figure 18

The parameters used for this model are shown in table 2

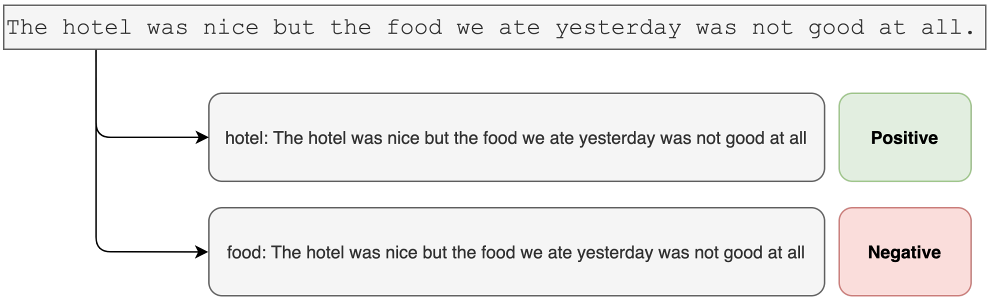

The hotel was nice but the food we ate yesterday was not good at all.
hotel: The hotel was nice but the food we ate yesterday was not good at all
Positive
food: The hotel was nice but the food we ate yesterday was not good at all
Negative

Figure 17: Training set for the aspect term polarity extraction subtask

Parameter	Value
LSTM_Layer	300
Word_Embedding_output_dim	300
Aspect_Embedding_output_dim	300
Dense_Layers	300
n_epochs	25
batch_size	32
Loss_Function	categorical_crossentropy
Optimizer	Adam

Table 2: Parameters used for aspect term polarity extraction model

4.3 The Proposed AHR-BERT Model

The Arabic Hotel Reviews BERT (AHR-BERT) model uses the public BERT model (101), which is a trained BERT model on the Arabic language. The main idea behind this proposed model is that it is further trained on a domain-specific text to increase its understanding of the task domain, the Arabic hotels’ reviews. The explanation of the post-training

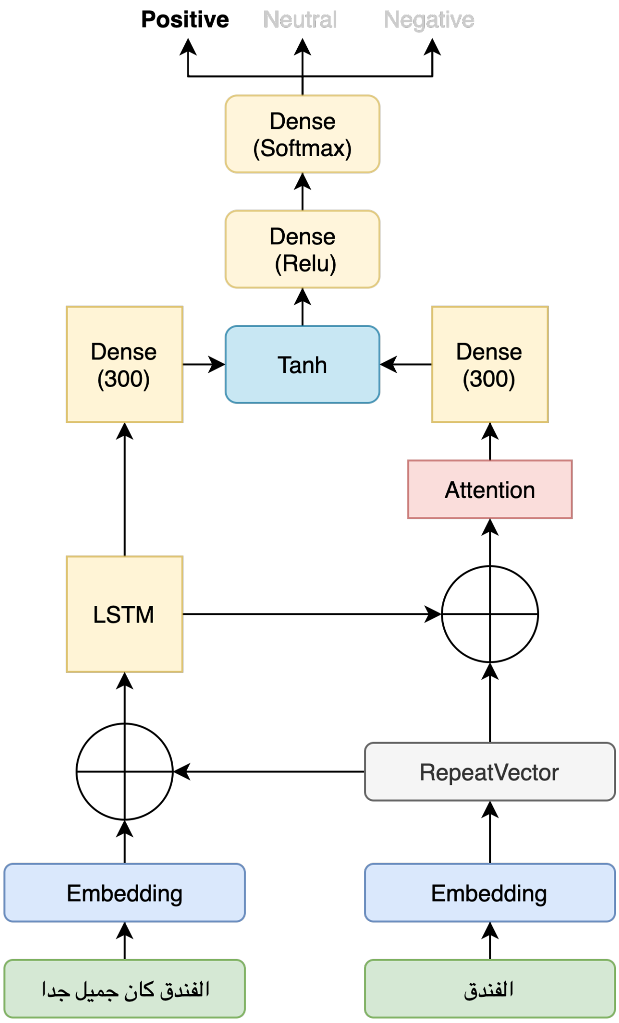


Figure 18: The system used for aspect polarity extraction subtask using a deep learning model

phase can be found in 4.3.2. Then the model is fine-tuned on the Arabic hotel reviews dataset; more details are illustrated in 4.3.3. This method shows improvement in results for both aspect term extraction and aspect term polarity tasks.

4.3.1 Bert Base CAMeLBERT MSA Eighth

This BERT model is used as a starting point to train our model. The bert base camelbert msa eight model was introduced by (101). This model is a part of the CAMeLBERT collection, a collection of pre-trained models for Arabic NLP tasks. The CAMeLBERT msa eight model was trained on 14GB text (which is 1.6 billion words). It has eight in its name because this model was pre-trained on eight of the full MSA dataset they collected.

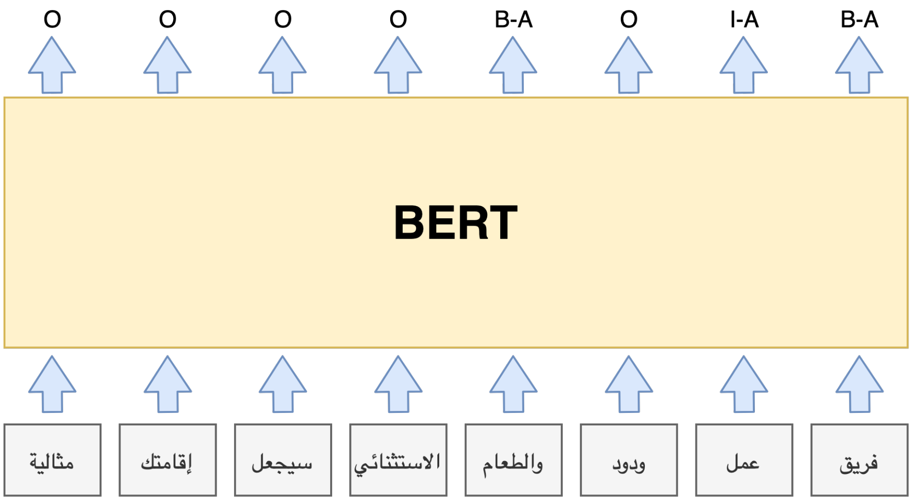


Figure 19: Using BERT model for Aspect Extraction

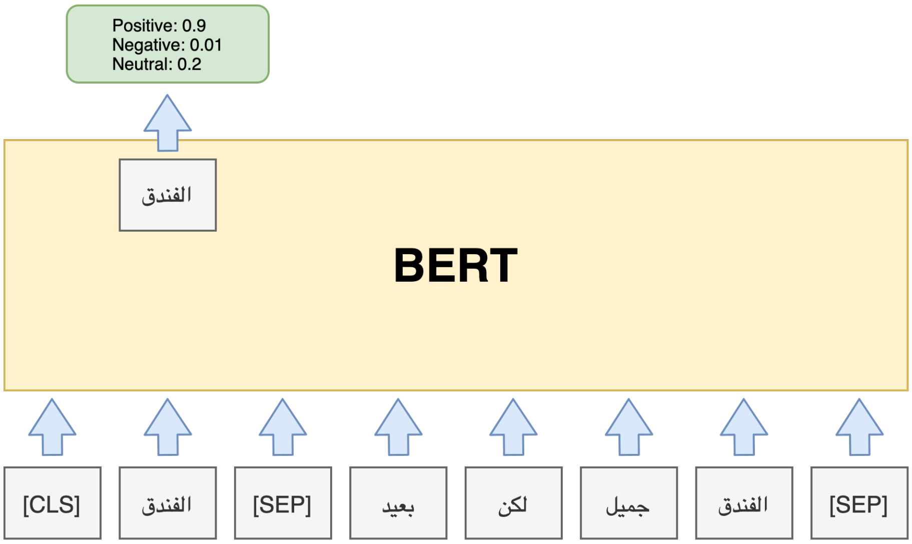


Figure 20: Using BERT model for Sentiment Prediction

4.3.2 Post-Training BERT

(102) proved that using a post-trained BERT model helps to get better results. (54) demonstrate the improved performance of BERT on ABSA for online reviews by post-training it on data from the same domain. Post-training involves initializing the model with the BERT pre-trained weights and continuing on a more specific level with the same self-supervised training regime. In our case, our downstream task covers hotel reviews, so a large chunk of online hotel reviews (92) is used to post-train the Arabic pre-trained BERT model (90). The parameters used while training the model are shown in Table 3.

Parameter	Value
attention_probs_dropout_prob	0.1
hidden_act	gelu
hidden_dropout_prob	0.1
hidden_size	768
layer_norm_eps	1.00E-12
num_attention_heads	12
num_hidden_layers	12
vocab_size	30000

Table 3: Parameters used to post-train the BERT model

The large chunk of online hotel reviews (92) is called Hotel Arabic-Reviews Dataset (HARD). It contains 93700 Arabic hotel reviews. The hotel reviews are from Booking.com during 2016, and they are in MSA and dialectal Arabic.

4.3.3 Fine-Tuning

Instead of training a model from scratch, a pre-trained model that was trained on a large corpus was used as a starting point. Then train this model on a much smaller dataset. This is what fine-tuning is. This research uses a pre-trained BERT model trained on a large Arabic corpus, then post-trains it on a domain-specific dataset for Arabic hotel reviews, and finally, fine-tunes it on the Arabic ABSA dataset that was used for this research. Figure 19 and figure 20 shows the BERT models for ATE and ATP. The parameters used to fine-tune the model are shown in table 4.

Parameter	Value
train_batch_size	5
num_train_epochs	2
gradient_accumulation_steps	8
learning_rate	0.0001

Table 4: Parameters used to fine-train the BERT model

Model	New prediction time	Post-train time
AHR model (ATE)	236ms	30280 seconds
AHR model (ATP)	41ms	30280 seconds

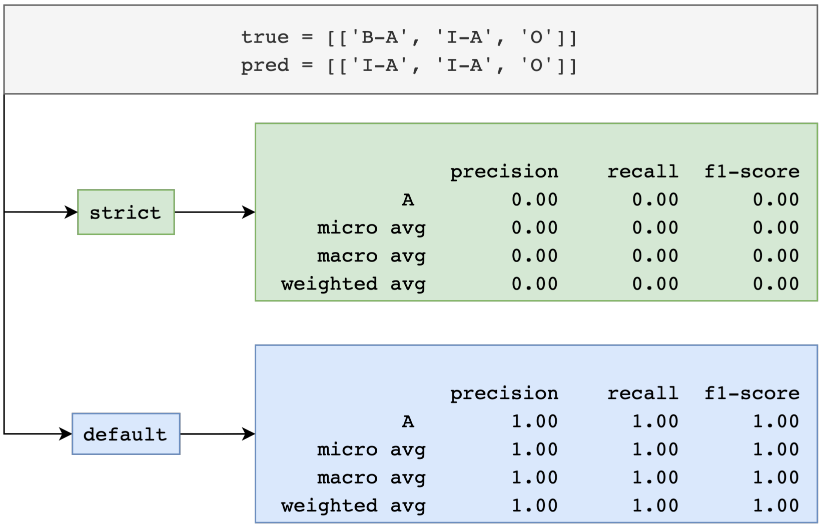
Table 5: Time taken to post-train the AHR model and make new predictions using the (measurements are done using google colaboratory pro)

4.3.4 Time Taken by AHR Model

Since this research is working with a model based on BERT, which is considered a large model, it reports the time taken for each model to make a new prediction in table 5. It also reports the duration it took to post-train the AHR model on the Arabic hotel reviews. Time is essential when implementing these models with projects that require many customers interactions.

4.3.5 Evaluation method

The evaluation metric used for this task is the F1-score. A python library called seqeval (103) is used to calculate the F1-score for the aspect extraction task. This library provides different ways to calculate the score for IOB tags. For this task, the "strict" mode is used, and the "IOB2" scheme, and the "strict" mode means that both the B tag followed by the I tag must match both true labels and the predictions; otherwise, it would be wrong. This is illustrated in figure 21. Keep in mind that the O tag is excluded from the evaluation. Otherwise, the accuracy will be biased towards the O tags.



```
true = [['B-A', 'I-A', 'O']]
pred = [['I-A', 'I-A', 'O']]
```

```
              precision    recall  f1-score
           A       0.00      0.00      0.00
   micro avg       0.00      0.00      0.00
   macro avg       0.00      0.00      0.00
weighted avg       0.00      0.00      0.00
```

```
              precision    recall  f1-score
           A       1.00      1.00      1.00
   micro avg       1.00      1.00      1.00
   macro avg       1.00      1.00      1.00
weighted avg       1.00      1.00      1.00
```

Figure 21: Seqeval library, the difference between strict mode and default mode

Chapter Five: Results and Discussions

The baseline approach used for ATE and ATP was SVM with n-grams features. The baseline leverages the LIBSVM library (104) to train an SVM model and make predictions. The work in this research outperformed the baseline by 43.5% for aspect term extraction and 13.6% for aspect term polarity. The AHR-BERT model results significantly improved the research papers that used deep learning with the same Arabic hotel reviews dataset. The proposed approach achieved 74.4% F1-score in the aspect extraction task outperforming Arabic-trained BERT models and implementations of LSTM and GRU. While in aspect term polarity task, the AHR-BERT model achieves a 90% accuracy score, outperforming the deep learning models applied to the same dataset.

The evaluation metric used for this task is the F1-score. A library called seqeval (103) is used to calculate the F1-score for the aspect extraction task. This library provides different ways to calculate the score for IOB tags. This task uses the “strict” mode, and “IOB2” scheme; the “strict” mode means that both the B tag followed by the I tag must match both true labels and the predictions; otherwise, it would be wrong.

Then a comparison is conducted between our proposed model AHR-BERT and other research papers that used different models such as BERT, deep learning, and machine learning models as shown in Table 6.

Table 6 shows a comparison between the best deep learning models and our proposed AHR-BERT model for both aspect extraction and aspect term polarity subtasks. The models mentioned in the table use the same Arabic hotel reviews used by this research. Al-Smadi et al. (48) used a Bidirectional LSTM approach for both subtasks; they used character-level bidirectional LSTM along with a CRF (Bi-LSTM-CRF) for the first subtask. Using this method, they overcame the baseline’s result by 39% for the OTE extraction subtask. For the second subtask, they used aspect-based LSTM in which the aspect-OTEs are considered attention expressions to support the sentiment polarity identification, and this method achieved a 6% increase to the baseline result for the polarity subtask. Their overall results are 69.98% F1-score for the first subtask and an 82.6% accuracy for the second subtask.

Task	Approach		F1-Score	Accuracy
Aspect extraction	BERT	Proposed AHR-BERT model	**74.4**	-
		bert-base-camelbert-msa-eighth	73.6	-
		bert-base-arabertv02	73.4	-
Aspect extraction	Deep learning	BLSTM-CRF (48)	69.98	-
		BGRU-CRF (105)	69.44	-
		Char-BGRU-CRF	64.0	-
	Machine Learning	Baseline	30.9	-
Aspect term polarity	BERT	Proposed AHR-BERT model	-	**90.0**
	Deep learning	LSTM-Attention model	-	84.38
		AB-LSTM-PC (48)	-	82.6
		IAN (BGRU) (105)	-	83.98
		INSIGHT-1 (CNN) (52)	-	82.7
	Machine Learning	Baseline	-	76.4

Table 6: The AHR-BERT results on Aspect extraction and Aspect term polarity in comparison to the baseline results based on SVM along with N-gram features as well as approaches form related work

Chapter Six: Conclusions and Future Work

6.1 Conclusions of the Outcomes

ABSA provides more details than the traditional SA tasks; leveraging this valuable information can gain helpful insights and usages to the business world. ABSA consists of three main subtasks; the first subtask is about extracting the aspects mentioned in the review. The second subtask is about finding the polarity for the extracted aspects; the polarity can be negative, neutral, or positive. The third subtask is about detecting the category of the extracted aspects. This research focused on the first and second subtasks. For both subtasks, the final work used several deep learning models and a state-of-the-art BERT model. The best model was the BERT model; the BERT model was post-trained on a large chunk of domain-specific data and then fine-tuned it on the Arabic hotel reviews from SemEval-2016 task 5. To the best of our knowledge, our proposed model AHR-BERT outperforms all the deep learning models used for the same dataset. The AHR-BERT achieves a 74.4% f1 score for ATE task and 90% for the ATP task.

6.2 Future Work

For future work, we plan to increase the accuracy of our model by collecting a domain-specific dataset and using it to post-train the AHR-BERT model. We also plan to create an automated system that scraps hotel reviews every once in a while and then updates the analysis results.

References:

1. Ali Farghaly and Khaled Shaalan. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8 (4):1–22, 2009.

2. Nizar Habash, Owen Rambow, and Ryan Roth. Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, volume 41, page 62, 2009.

3. Imad A. Al-Sughaiyer and Ibrahim A. Al-Kharashi. Arabic morphological analysis techniques: A comprehensive survey. *J. Assoc. Inf. Sci. Technol.*, pages 189–213, 2004. doi: 10.1002/asi.10368. URL `https://doi.org/10.1002/asi.10368`.

4. Adrian Micu, Angela Eliza Micu, Marius Geru, and Radu Constantin Lixandroiu. Analyzing user sentiment in social media: Implications for online marketing strategy. *Psychology & Marketing*, 34(12):1094–1100, 2017.

5. Brian Dean. Social network usage & growth statistics: How many people use social media in 2021. *Published August*, 12, 2020.

6. Christy M. K. Cheung and Dimple R. Thadani. The effectiveness of electronic word-of-mouth communication: A literature analysis. page 18, 2010. URL `http://aisel.aisnet.org/bled2010/18`.

7. Inès Blal and Michael C Sturman. The differential effects of the quality and quantity of online reviews on hotel room sales. *Cornell Hospitality Quarterly*, 55(4):365–375, 2014.

8. M Nick Hajli. A study of the impact of social media on consumers. *International Journal of Market Research*, 56(3):387–404, 2014.

9. Daniel J. Power and Gloria E. Phillips-Wren. Impact of social media and web 2.0 on decision-making. *J. Decis. Syst.*, pages 249–261, 2011. doi: 10.3166/jds.20.249-261. URL `https://doi.org/10.3166/jds.20.249-261`.

10. B Narendra, K Uday Sai, G Rajesh, K Hemanth, MV Chaitanya Teja, and K Deva Kumar. Sentiment analysis on movie reviews: a comparative study of machine learning algorithms and open source technologies. *International Journal of Intelligent Systems and Applications*, 8(8):66, 2016.

11. Rushlene Kaur Bakshi, Navneet Kaur, Ravneet Kaur, and Gurpreet Kaur. Opinion mining and sentiment analysis. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 452–455. IEEE, 2016.

12. Lei Zhang and Bing Liu. Sentiment analysis and opinion mining. In *Encyclopedia of Machine Learning and Data Mining*, pages 1152–1161. Springer, 2017. doi: 10.1007/978-1-4899-7687-1\\_907. URL `https://doi.org/10.1007/978-1-4899-7687-1\_907`.

13. Federico Neri, Carlo Aliprandi, Federico Capeci, Montserrat Cuadros, and Tomas By. Sentiment analysis on social media. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 919–926. IEEE, 2012.

14. Francesco Benedetto and Antonio Tedeschi. Big data sentiment analysis for brand monitoring in social media streams by cloud computing. In *Sentiment Analysis and Ontology Engineering*, pages 341–377. Springer, 2016.

15. Bryan Li, Dimitrios Dimitriadis, and Andreas Stolcke. Acoustic and lexical sentiment analysis for customer service calls. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5876–5880. IEEE, 2019.

16. Meena Rambocas, João Gama, et al. Marketing research: The role of sentiment analysis. Technical report, Universidade do Porto, Faculdade de Economia do Porto, 2013.

17. Cäcilia Zirn, Mathias Niepert, Heiner Stuckenschmidt, and Michael Strube. Fine-grained sentiment analysis with structural features. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 336–344, 2011.

18. Kerstin Denecke. Using sentiwordnet for multilingual sentiment analysis. In *Proceedings of the 24th International Conference on Data Engineering Workshops, ICDE 2008, April 7-12, 2008, Cancún, Mexico*, pages 507–512. IEEE Computer Society, 2008. doi: 10.1109/ICDEW.2008.4498370. URL `https://doi.org/10.1109/ICDEW.2008.4498370`.

19. Ioannis Pavlopoulos. Aspect based sentiment analysis. *Athens University of Economics and Business*, 2014.

20. Muhamad Rizky Yanuar and Shun Shiramatsu. Aspect extraction for tourist spot review in indonesian language using BERT. In *2020 International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2020, Fukuoka, Japan, February 19-21, 2020*, pages 298–302. IEEE, 2020. doi: 10.1109/ICAIIC48513.2020.9065263. URL `https://doi.org/10.1109/ICAIIC48513.2020.9065263`.

21. Youwei Song, Jiahai Wang, Zhiwei Liang, Zhiyue Liu, and Tao Jiang. Utilizing BERT intermediate layers for aspect based sentiment analysis and natural language inference. *arXiv preprint arXiv:2002.04815*, 2020. URL `https://arxiv.org/abs/2002.04815`.

22. Hu Xu, Lei Shu, Philip S. Yu, and Bing Liu. Understanding pre-trained BERT for aspect-based sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 244–250. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.21. URL `https://doi.org/10.18653/v1/2020.coling-main.21`.

23. Akbar Karimi, Leonardo Rossi, and Andrea Prati. Improving BERT performance for aspect-based sentiment analysis. *arXiv preprint arXiv:2010.11731*, 2020. URL `https://arxiv.org/abs/2010.11731`.

24. Zhengxuan Wu and Desmond C. Ong. Context-guided BERT for targeted aspect-based sentiment analysis. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14094–14102. AAAI Press, 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/17659`.

25. Oanh Thi Tran and Viet The Bui. A bert-based hierarchical model for vietnamese aspect based sentiment analysis. In *12th International Conference on Knowledge and Systems Engineering, KSE 2020, Can Tho City, Vietnam, November 12-14, 2020*, pages 269–274. IEEE, 2020. doi: 10.1109/KSE50997.2020.9287650. URL `https://doi.org/10.1109/KSE50997.2020.9287650`.

26. Hamoon Jafarian, Amirhosein Taghavi, Alireza Javaheri, and Reza Rawassizadeh. Exploiting BERT to improve aspect-based sentiment analysis performance on persian language. *arXiv preprint arXiv:2012.07510*, 2020. URL `https://arxiv.org/abs/2012.07510`.

27. Akbar Karimi, Leonardo Rossi, and Andrea Prati. Adversarial training for aspect-based sentiment analysis with BERT. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pages 8797–8803. IEEE, 2020. doi: 10.1109/ICPR48806.2021.9412167. URL `https://doi.org/10.1109/ICPR48806.2021.9412167`.

28. Tomás Brychcín, Michal Konkol, and Josef Steinberger. UWB: machine learning approach to aspect-based sentiment analysis. In Preslav Nakov and Torsten Zesch, editors, *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 817–822. The Association for Computer Linguistics, 2014. doi: 10.3115/v1/s14-2145. URL `https://doi.org/10.3115/v1/s14-2145`.

29. Tomás Hercig, Tomás Brychcín, Lukás Svoboda, and Michal Konkol. UWB at semeval-2016 task 5: Aspect based sentiment analysis. In Steven Bethard, Daniel M. Cer, Marine Carpuat, David Jurgens, Preslav Nakov, and Torsten Zesch, editors, *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 342–349. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/s16-1055. URL `https://doi.org/10.18653/v1/s16-1055`.

30. Tamara Álvarez-López, Jonathan Juncal-Martínez, Milagros Fernández Gavilanes, Enrique Costa-Montenegro, and Francisco Javier González-Castaño. GTI at semeval-2016 task 5: SVM and CRF for aspect detection and unsupervised aspect-based sentiment analysis. In Steven Bethard, Daniel M. Cer, Marine Carpuat, David Jurgens, Preslav Nakov, and Torsten Zesch, editors, *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 306–311. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/s16-1049. URL `https://doi.org/10.18653/v1/s16-1049`.

31. Mengxiao Jiang, Zhihua Zhang, and Man Lan. ECNU at semeval-2016 task 5: Extracting effective features from relevant fragments in sentence for aspect-based sentiment analysis in reviews. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 361–366. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/s16-1058. URL `https://doi.org/10.18653/v1/s16-1058`.

32. Ayush Kumar, Sarah Kohail, Amit Kumar, Asif Ekbal, and Chris Biemann. IIT-TUDA at semeval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 1129–1135. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/s16-1174. URL `https://doi.org/10.18653/v1/s16-1174`.

33. Asha S. Manek, P. Deepa Shenoy, M. Chandra Mohan, and K. R. Venugopal. Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and SVM classifier. *World Wide Web*, pages 135–154, 2017. doi: 10.1007/s11280-015-0381-x. URL `https://doi.org/10.1007/s11280-015-0381-x`.

34. Mohammed Matuq Ashi, Muazzam Ahmed Siddiqui, and Farrukh Nadeem. Pre-trained word embeddings for arabic aspect-based sentiment analysis of airline tweets. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, AISI 2018, Cairo, Egypt, September 3-5, 2018*, pages 241–251. Springer, 2018. doi: 10.1007/978-3-319-99010-1\\_22. URL `https://doi.org/10.1007/978-3-319-99010-1\_22`.

35. Mohammad Al-Smadi, Omar Qawasmeh, Mahmoud Al-Ayyoub, Yaser Jararweh, and Brij B. Gupta. Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of arabic hotels' reviews. *J. Comput. Sci.*, 27:386–393, 2018. doi: 10.1016/j.jocs.2017.11.006. URL https://doi.org/10.1016/j.jocs.2017.11.006.

36. Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yaser Jararweh, and Omar Qawasmeh. Enhancing aspect-based sentiment analysis of arabic hotels' reviews using morphological, syntactic and semantic features. *Inf. Process. Manag.*, 56(2):308–319, 2019. doi: 10.1016/j.ipm.2018.01.006. URL https://doi.org/10.1016/j.ipm.2018.01.006.

37. Mark A. Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explor.*, pages 10–18, 2009. doi: 10.1145/1656274.1656278. URL https://doi.org/10.1145/1656274.1656278.

38. Trigui Sana, Ines Boujelben, Salma Jamoussi, and Yassine Ben Ayed. A hybrid method for arabic aspect-based sentiment analysis. *Int. J. Hybrid Intell. Syst.*, pages 99–110, 2020. doi: 10.3233/HIS-200285. URL https://doi.org/10.3233/HIS-200285.

39. Caroline Brun, Diana Nicoleta Popa, and Claude Roux. XRCE: hybrid classification for aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 838–842. The Association for Computer Linguistics, 2014. doi: 10.3115/v1/s14-2149. URL https://doi.org/10.3115/v1/s14-2149.

40. Aitor García Pablos, Montse Cuadros, and German Rigau. W2VLDA: almost unsupervised system for aspect based sentiment analysis. *Expert Syst. Appl.*, 91:127–137, 2018. doi: 10.1016/j.eswa.2017.08.049. URL https://doi.org/10.1016/j.eswa.2017.08.049.

41. Zhiqiang Toh and Jian Su. NLANGP at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 282–288. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/s16-1045. URL https://doi.org/10.18653/v1/s16-1045.

42. Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 616–626. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1059. URL https://doi.org/10.18653/v1/d16-1059.

43. Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. Unsupervised word and dependency path embeddings for aspect term extraction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2979–2985. IJCAI/AAAI Press, 2016. URL `http://www.ijcai.org/Abstract/16/423`.

44. Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 999–1005. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1103. URL `https://doi.org/10.18653/v1/d16-1103`.

45. Yukun Ma, Haiyun Peng, and Erik Cambria. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5876–5883. AAAI Press, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16541`.

46. Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1546–1556. ACL, 2016. URL `https://www.aclweb.org/anthology/C16-1146/`.

47. Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 486–495. The Association for Computer Linguistics, 2015. doi: 10.18653/v1/s15-2082. URL `https://doi.org/10.18653/v1/s15-2082`.

48. Mohammad Al-Smadi, Bashar Talafha, Mahmoud Al-Ayyoub, and Yaser Jararweh. Using long short-term memory deep neural networks for aspect-based sentiment analysis of arabic reviews. *Int. J. Mach. Learn. Cybern.*, pages 2163–2175, 2019. doi: 10.1007/s13042-018-0799-4. URL `https://doi.org/10.1007/s13042-018-0799-4`.

49. Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia V. Loukachevitch, Evgeniy V. Kotelnikov, Núria Bel, Salud María Jiménez Zafra, and Gülsen Eryigit. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 19–30. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/s16-1002. URL `https://doi.org/10.18653/v1/s16-1002`.

50. Wei Xue and Tao Li. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2514–2523. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1234. URL `https://www.aclweb.org/anthology/P18-1234/`.

51. Adnan Ishaq, Sohail Asghar, and Saira Andleeb Gillani. Aspect-based sentiment analysis using a hybridized approach based on CNN and GA. *IEEE Access*, pages 135499–135512, 2020. doi: 10.1109/ACCESS.2020.3011802. URL `https://doi.org/10.1109/ACCESS.2020.3011802`.

52. Hai Ha Do, P. W. C. Prasad, Angelika Maag, and Abeer Alsadoon. Deep learning for aspect-based sentiment analysis: A comparative review. *Expert Syst. Appl.*, pages 272–299, 2019. doi: 10.1016/j.eswa.2018.10.003. URL `https://doi.org/10.1016/j.eswa.2018.10.003`.

53. Bo Wang and Min Liu. Deep learning for aspect-based sentiment analysis. *Stanford University report*, 2015.

54. Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2324–2335. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1242. URL `https://doi.org/10.18653/v1/n19-1242`.

55. Ruining He and Julian J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517. ACM, 2016. doi: 10.1145/2872427.2883037. URL `https://doi.org/10.1145/2872427.2883037`.

56. Nabiha Asghar. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*, 2016. URL `http://arxiv.org/abs/1605.05362`.

57. Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. Aspect-based sentiment analysis using BERT. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics, NoDaLiDa 2019, Turku, Finland, September 30 - October 2, 2019*, pages 187–196. Linköping University Electronic Press, 2019. URL `https://aclweb.org/anthology/W19-6120/`.

58. Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. Exploiting BERT for end-to-end aspect-based sentiment analysis. In *Proceedings of the 5th Workshop on Noisy User-generated Text, W-NUT@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 34–41. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-5505. URL `https://doi.org/10.18653/v1/D19-5505`.

59. Junqi Dai, Hang Yan, Tianxiang Sun, Pengfei Liu, and Xipeng Qiu. Does syntax matter? A strong baseline for aspect-based sentiment analysis with roberta. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1816–1829. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.146. URL `https://doi.org/10.18653/v1/2021.naacl-main.146`.

60. Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, 2:93–128, 2006.

61. Roman Klinger and Katrin Tomanek. *Classical probabilistic models and conditional random fields*. Citeseer, 2007.

62. Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL `http://arxiv.org/abs/1301.3781`.

63. Tomás Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 746–751. The Association for Computational Linguistics, 2013. URL `https://www.aclweb.org/anthology/N13-1090/`.

64. Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul R. Kingsbury, and Hongfang Liu. A comparison of word embeddings for the biomedical natural language processing. *J. Biomed. Informatics*, pages 12–20, 2018. doi: 10.1016/j.jbi.2018.09.008. URL `https://doi.org/10.1016/j.jbi.2018.09.008`.

65. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, pages 135–146, 2017. URL `https://transacl.org/ojs/index.php/tacl/article/view/999`.

66. Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431. Association for Computational Linguistics, 2017. doi: 10.18653/v1/e17-2068. URL `https://doi.org/10.18653/v1/e17-2068`.

67. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014. doi: 10.3115/v1/d14-1162. URL `https://doi.org/10.3115/v1/d14-1162`.

68. Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

69. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

70. Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Handwritten digit recognition with a back-propagation network. pages 396–404, 1989. URL `http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network`.

71. Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 103–112. The Association for Computational Linguistics, 2015. doi: 10.3115/v1/n15-1011. URL `https://doi.org/10.3115/v1/n15-1011`.

72. Linjie Xing and Yu Qiao. Deepwriter: A multi-stream deep CNN for text-independent writer identification. In *15th International Conference on Frontiers in Handwriting Recognition, ICFHR 2016, Shenzhen, China, October 23-26, 2016*, pages 584–589. IEEE Computer Society, 2016. doi: 10.1109/ICFHR.2016.0112. URL `https://doi.org/10.1109/ICFHR.2016.0112`.

73. Daojian Zeng, Yuan Dai, Feng Li, Jin Wang, and Arun Kumar Sangaiah. Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism. *J. Intell. Fuzzy Syst.*, pages 3971–3980, 2019. doi: 10.3233/JIFS-169958. URL `https://doi.org/10.3233/JIFS-169958`.

74. Budi M Mulyo and Dwi H Widyantoro. Aspect-based sentiment analysis approach with cnn. In *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pages 142–147. IEEE, 2018.

75. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, pages 84–90, 2017. doi: 10.1145/3065386. URL `http://doi.acm.org/10.1145/3065386`.

76. Grégoire Mesnil, Yann N. Dauphin, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian J. Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, David Warde-Farley, Pascal Vincent, Aaron C. Courville, and James Bergstra. Unsupervised and transfer learning challenge: a deep learning approach. In *Unsupervised and Transfer Learning - Workshop held at ICML 2011, Bellevue, Washington, USA, July 2, 2011*, pages 97–110. JMLR.org, 2012. URL `http://proceedings.mlr.press/v27/mesnil12a.html`.

77. Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL, 2014. doi: 10.3115/v1/d14-1179. URL `https://doi.org/10.3115/v1/d14-1179`.

78. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

79. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.0473`.

80. Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-4012. URL `https://www.aclweb.org/anthology/W14-4012/`.

81. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pretraining of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL `https://doi.org/10.18653/v1/n19-1423`.

82. Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. BERT with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1133–1136. ACM, 2019. doi: 10.1145/3331184.3331341. URL `https://doi.org/10.1145/3331184.3331341`.

83. Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 72–77. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-4013. URL `https://doi.org/10.18653/v1/n19-4013`.

84. Derek Miller. Leveraging BERT for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*, 2019. URL `http://arxiv.org/abs/1906.04165`.

85. Yang Liu. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019. URL `http://arxiv.org/abs/1903.10318`.

86. Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. Are natural language inference models imppressive? learning implicature and presupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8690–8705. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.768. URL `https://doi.org/10.18653/v1/2020.acl-main.768`.

87. Xinlong Li, Xingyu Fu, Guangluan Xu, Yang Yang, Jiuniu Wang, Li Jin, Qing Liu, and Tianyuan Xiang. Enhancing BERT representation with context-aware embedding for aspect-based sentiment analysis. *IEEE Access*, pages 46868–46876, 2020. doi: 10.1109/ACCESS.2020.2978511. URL `https://doi.org/10.1109/ACCESS.2020.2978511`.

88. Nour Al-Khdour, Mutaz Bni Younes, Malak Abdullah, and Mohammad Al-Smadi. Justmasters at semeval-2020 task 3: Multilingual deep learning model to predict the effect of context in word similarity. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation, SemEval@COLING 2020, Barcelona (online), December 12-13, 2020*, pages 292–300. International Committee for Computational Linguistics, 2020. URL `https://www.aclweb.org/anthology/2020.semeval-1.37/`.

89. Mutaz Bni Younes and Nour Al-Khdour. Team alexa at authorship identification of source code (AI-SOCO). In *Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020*, pages 699–704. CEUR-WS.org, 2020. URL `http://ceur-ws.org/Vol-2826/T5-4.pdf`.

90. Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. The interplay of variant, size, and task type in arabic pre-trained language models. *arXiv preprint arXiv:2103.06678*, 2021. URL `https://arxiv.org/abs/2103.06678`.

91. Wissam Antoun, Fady Baly, and Hazem M. Hajj. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*, 2020. URL `https://arxiv.org/abs/2003.00104`.

92. Ashraf Elnagar, Yasmin S Khalifa, and Anas Einea. Hotel arabic-reviews dataset construction for sentiment analysis applications. In *Intelligent natural language processing: Trends and applications*, pages 35–52. Springer, 2018.

93. Mahmoud Nabil, Mohamed A. Aly, and Amir F. Atiya. ASTD: arabic sentiment tweets dataset. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2515–2519. The Association for Computational Linguistics, 2015. doi: 10.18653/v1/d15-1299. URL `https://doi.org/10.18653/v1/d15-1299`.

94. Ramy Baly, Alaa Khaddaj, Hazem M. Hajj, Wassim El-Hajj, and Khaled Bashir Shaban. Arsentd-lev: A multi-topic corpus for target-based sentiment analysis in arabic levantine tweets. *arXiv preprint arXiv:1906.01830*, 2019. URL `http://arxiv.org/abs/1906.01830`.

95. Mohamed A. Aly and Amir F. Atiya. LABR: A large scale arabic book reviews dataset. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 494–498. The Association for Computer Linguistics, 2013. URL `https://www.aclweb.org/anthology/P13-2088/`.

96. Yassine Benajiba, Mona Diab, Paolo Rosso, et al. Arabic named entity recognition: An svm-based approach. In *Proceedings of 2008 Arab International Conference on Information Technology (ACIT)*, pages 16–18. Citeseer, 2008.

97. Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem M. Hajj. Neural arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop, WANLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 108–118. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-4612. URL `https://doi.org/10.18653/v1/w19-4612`.

98. Mohammad Al-Smadi, Omar Qawasmeh, Bashar Talafha, Mahmoud Al-Ayyoub, Yaser Jararweh, and Elhadj Benkhelifa. An enhanced framework for aspect-based sentiment analysis of hotels' reviews: Arabic reviews case study. In *11th International Conference for Internet Technology and Secured Transactions, ICITST 2016, Barcelona, Spain, December 5-7, 2016*, pages 98–103. IEEE, 2016. doi: 10.1109/ICITST.2016.7856675. URL `https://doi.org/10.1109/ICITST.2016.7856675`.

99. Hady ElSahar and Samhaa R. El-Beltagy. Building large arabic multi-domain resources for sentiment analysis. In *Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part II*, pages 23–34. Springer, 2015. doi: 10.1007/978-3-319-18117-2\\_2. URL `https://doi.org/10.1007/978-3-319-18117-2\_2`.

100. Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 606–615. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1058. URL `https://doi.org/10.18653/v1/d16-1058`.

101. Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online), April 2021. Association for Computational Linguistics.

102. Bashar Talafha, Mohammad Ali, Muhy Eddin Za'ter, Haitham Seelawi, Ibraheem Tuffaha, Mostafa Samir, Wael Farhan, and Hussein T. Al-Natsheh. Multi-dialect arabic BERT for country-level dialect identification. *arXiv preprint arXiv:2007.05612*, 2020. URL `https://arxiv.org/abs/2007.05612`.

103. Hiroki Nakayama. seqeval: A python framework for sequence labeling evaluation, 2018. URL `https://github.com/chakki-works/seqeval`. Software available from https://github.com/chakki-works/seqeval.

104. Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

105. Mohammed Mustafa, Taysir Hassan A. Soliman, Ahmed Ibrahim Taloba, and Mohammed Fawzi Seedik. Arabic aspect based sentiment analysis using bidirectional GRU based models. *arXiv preprint arXiv:2101.10539*, 2021. URL `https://arxiv.org/abs/2101.10539`.