applied sciences

MDPI

Article

MenuNER: Domain-Adapted BERT Based NER Approach for a Domain with Limited Dataset and Its Application to Food Menu Domain

Muzamil Hussain Syed [1,*] and Sun-Tae Chung [2]

[1] Department of Information and Telecommunication, Graduate School, Soongsil University, Seoul 06978, Korea
[2] School of Artificial Intelligence Convergence, Soongsil University, Seoul 06978, Korea; cst@ssu.ac.kr
* Correspondence: engr.muzamilshah@gmail.com

Abstract: Entity-based information extraction is one of the main applications of Natural Language Processing (NLP). Recently, deep transfer-learning utilizing contextualized word embedding from pre-trained language models has shown remarkable results for many NLP tasks, including Named-entity recognition (NER). BERT (Bidirectional Encoder Representations from Transformers) is gaining prominent attention among various contextualized word embedding models as a state-of-the-art pre-trained language model. It is quite expensive to train a BERT model from scratch for a new application domain since it needs a huge dataset and enormous computing time. In this paper, we focus on menu entity extraction from online user reviews for the restaurant and propose a simple but effective approach for NER task on a new domain where a large dataset is rarely available or difficult to prepare, such as food menu domain, based on domain adaptation technique for word embedding and fine-tuning the popular NER task network model 'Bi-LSTM+CRF' with extended feature vectors. The proposed NER approach (named as 'MenuNER') consists of two step-processes: (1) Domain adaptation for target domain; further pre-training of the off-the-shelf BERT language model (BERT-base) in semi-supervised fashion on a domain-specific dataset, and (2) Supervised fine-tuning the popular Bi-LSTM+CRF network for downstream task with extended feature vectors obtained by concatenating word embedding from the domain-adapted pre-trained BERT model from the first step, character embedding and POS tag feature information. Experimental results on handcrafted food menu corpus from customers' review dataset show that our proposed approach for domain-specific NER task, that is: food menu named-entity recognition, performs significantly better than the one based on the baseline off-the-shelf BERT-base model. The proposed approach achieves 92.5% F1 score on the YELP dataset for the MenuNER task.

Keywords: named-entity recognition; domain adaptation; BERT pre-training; food menu

Citation: Syed, M.H.; Chung, S.-T. MenuNER: Domain-Adapted BERT Based NER Approach for a Domain with Limited Dataset and Its Application to Food Menu Domain. *Appl. Sci.* **2021**, *11*, 6007. https://doi.org/10.3390/app11136007

Academic Editors: Maxim Mozgovoy and Calkin Suero Montero

Received: 26 May 2021
Accepted: 24 June 2021
Published: 28 June 2021

1. Introduction

The aim of named-entity recognition (NER) is to detect chunks in a sentence representing an entity and assign a class to that entity. Named-entity recognition (NER) is one of the first and most important steps in Information Extraction pipelines that is to identify mentions of entities (persons, locations, organizations, menus, etc.) within unstructured text [1]. Although NER tasks have been effectively applied to various application domains and the general domain, several application domains remain not well discovered. The growing numbers of emerging food dishes with their diverse forms make recognizing food entities on social media platforms more challenging. Another challenge posed by user-generated content is its unique characteristics and use of informal language, which is typically short context, noisy, sparse and ambiguous contents [1].

In deep learning, NER is considered a sequence labeling task where the neural models usually contain three components: word embedding layer, context encoder layer, and

decoder layer [2]. Recurrent Neural Networks (RNNs) models, specifically, Bidirectional long short-term memory networks (Bi-LSTM) [3], due to their sequential characteristics and tremendous capability to learn the contextual representation of words, and Conditional Random Fields (CRF) [4], due to better sentence-level prediction are widely adapted for context encoder and decoder layers for sequence labeling tasks [5–8].

For word embedding, pre-trained model-based word embeddings, such as word2vec [9] and GloVe [10], and other contextualized word embeddings, such as ELMo [11] and BERT [12], have been utilized.

As a state-of-the-art pre-trained language model, BERT (Bidirectional Encoder Representations from Transformers) has achieved state-of-the-art accuracy in various NLP tasks, including text classification [13], machine translation [14] and named-entity recognition [15], etc. BERT is based on a multi-layer bidirectional transformer [16] architecture. It is pre-trained on a large amount of unlabeled data for masked word prediction and next sentence prediction tasks [12], which can then be fine-tuned on a labeled dataset for a specific downstream task. Developing BERT pre-trained model for a new domain from scratch is quite expensive since even the BERT-base model has 12-layer with roughly 110 million parameters. To train such a dense network requires a huge amount of dataset. It is extremely difficult for the majority of application domains, including the food domain, to make a dataset with that many documents.

Furthermore, training such a large dataset requires a huge amount of computing resources. A logical notion is to further pre-train the original off-the-shelf BERT with a limited target-domain dataset. When further pre-trained on domain-specific corpora, the BERT pre-trained model can improve performance in domain-specific tasks while maintaining good performance in general domain tasks [17,18]. The idea is termed as domain adaptation or domain transfer technique and has been applied in various NLP applications [13,19,20].

Post-training approaches for domain-adaptation are highly effective for the tasks where the training dataset is limited [17]. The main challenge in domain-specific NER is the lack of an annotated corpus.

In the case of food menu domain which has a complex set of multilingual global and regional cuisines names, finding food menu entities is a challenging task requiring domain-specific pre-trained models to learn the actual context of the text.

To address the limited number of hand-crafted training examples in the application domain, we first apply domain adaption to incorporate domain-specific weights in the vector space by further pre-training off-the-shelf BERT-base model (as shown in Figure 1) on target-domain dataset (YELP reviews dataset) to learn the better representation of target-domain entities (food menu entities) on user-generated text (as shown in Figure 2). Next, for fine-tuning the downstream NER network model, we use concatenation of character-level embedding and POS tag features to fine-tune MenuNER, in addition to domain-adapted BERT embeddings (as shown in Figure 3), which replaces the traditional word embeddings used in [21] and adopt a popular NER task network model 'Bi-LSTM+CRF' (as shown in Figure 4).

First, we perform ablation test to investigate the impacts of each input feature vector on the adopted NER task network model. It reveals that the POS tag and character-level embedding features and word embedding from the domain-adapted BERT (BERT-domain) have a synergistic effect. Later, the performance of the NER task network model with BERT-domain is compared with the BERT base model (BERT-base-cased) for the food menu NER task. Our findings based on experiments show that the proposed NER task approach based on model adaptation and extended feature vector with Bi-LSTM+CRf network model produces reasonably good performance on an application domain where a large dataset is hardly available and performs significantly better than the baseline BERT-base model.

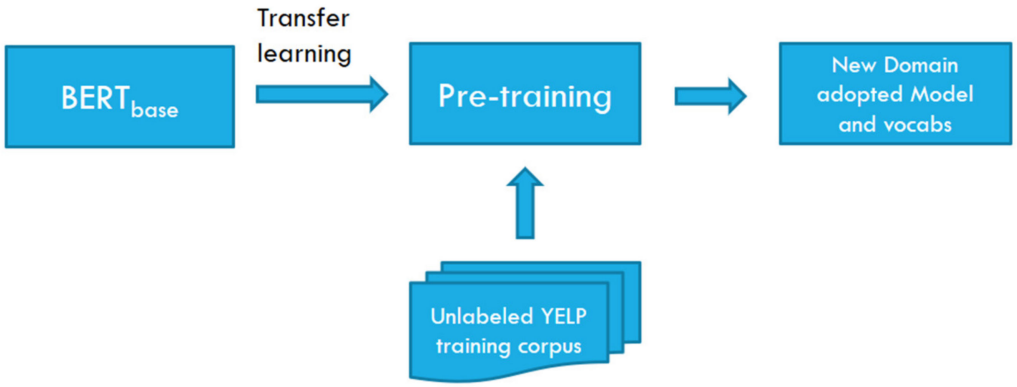

Figure 1. Further pre-training Pipeline of BERT language model.

Our main contributions of this paper are summarized as follows:

- Proposing a simple yet effective NER task approach based on domain adaptation technique with extended input features for an application domain where a large dataset is scarce.
- Evaluating the advantage and effectiveness of using domain adaptation and transfer learning techniques for NER task to recognize rich dishes' names in the food domain.
- Evaluating the impact of using domain-adapted BERT embedding along with character-level and POS tag feature representation to train the NER task network model with limited datasets.
- Achieving better results on the proposed NER task approach compared with baseline models for food menu NER.
- Preparing carefully annotated dataset for menu dishes.

In the remainder of the paper, we first present an overview of the related work. Then, we present the implementation of transfer learning approach followed by extended feature vectors representation with downstream Bi-LSTM+CRF model for the food menu named-entity recognition. Next, the data used for evaluation is described, followed by the experimental results. Finally, the conclusions of the paper are presented.

2. Related Work

This section first summarizes previous methods and studies on NER in general with BERT-based domain-adaptation techniques and deep learning models. Then we discuss related work for the entity recognition in the food domain.

2.1. Neural Network Model Architecture for NER

In 2015, Huang et al. [5] proposed the utilization of a Bi-LSTM network for past and future input features for sequence tagging task and CRF for sentence-level tag information to produce higher tagging accuracy. Compared with other state-of-the-art NER network architectures for the CoNLL-2003 (English) corpus, their NER model with both Senna embedding and gazetteer features achieved superior performance of 90.10% F1. Since then, the combination of Bi-LSTM and CRF has been popularly accepted as an effective network architecture for sequence tagging tasks, including NER.

Later, Lample et al. [7] proposed a neural network for NER task, consisting of Bi-LSTM and CRF with character-based word representations learned from the supervised corpus and unsupervised word representations learned from unannotated corpora. They utilized word embedding, which was pre-trained using skip-n-gram and character embedding based on a lookup table.

Ma and Hovy [6] introduce a novel neural network architecture that benefits from both GloVe-based word embedding, and character-level features, by using a combination of Bi-LSTM, CNN and CRF. Similarly, Saad et al. [22] presented a deep neural network architecture based on Bi-LSTM and CRF. They utilized GloVe word embedding models

and Bi-LSTM-based character level embeddings for biomedical named entities recognition. They showed that the proposed deep neural network model using word and character-level embeddings outperforms significantly well compared to only word-level embeddings, which implies that character-level embeddings can help with out-of-vocabulary words and misspelled words, different forms of the same entity, etc. As a result, the character-level representation can infer a representation of unknown words in the training data and increase the Bi-LSTM model performance [22].

By reflecting such previous research results, we construct a neural network model based on Bi-LSTM and CRF architecture, and imposing input feature vectors obtained from concatenating contextual word embedding, POS feature and character-level feature representations.

2.2. NER Based on Contextualized Word Representations

Seok et al. [21] compared word embedding features by applying GloVe, Word2Vec and CCA on general NER tasks. It was argued that when contextualized word embedding is used as a feature for the NER task, better results can be obtained than the baseline that does not use word embedding.

Zhai et al. [23] explored the performance of Chemical NER in patents by using BiLSTM-CRF model utilizing pre-trained word embeddings, character-level word representations and contextualized ELMo word representations. Their results depict that domain-specific embeddings trained on chemical patents and chemical-specific tokenizers improve NER performance.

Emelyanov and Artemova [24] proposed a solution for multilingual named-entity recognition task without requiring any additional labelled data. The BERT language model was used as embeddings without any fine-tuning with Bi-LSTM, Multi-Head attention, and NCRF on the top.

2.3. Transfer Learning and Domain Adaptation in NLP Tasks

Transfer learning is a machine learning technique in which a model that has been trained for one task is reused for another task with less labelled training data. Francis et al. [25] applied transfer learning for NER in financial and biomedical documents. They studied how a neural model for NER task trained for one entity type can be utilized to another entity type that is structurally or contextually similar to avoid the need for training the target model from scratch.

Recently, NER methods utilizing embeddings from pre-trained language model based on transformers such as BERT have been proposed in the literature. Rietzler et al. [20] approach ATSC (Aspect-Target Sentiment Classification) using a two-step procedure: self-supervised domain-specific BERT language model fine-tuning, followed by supervised task-specific fine-tuning. Their findings reveal that a cross-domain adapted BERT language model outperforms the strong baseline models like vanilla BERT-base and XLNet-base substantially.

In order to maximize the utilization of BERT for the text classification task, Sun et al. [13] proposed further pre-training BERT on task domain data and fine-tuning BERT for the target task. The off-the-shelf BERT model is pre-trained in the general domain, which has a different data distribution from the target domain. In addition to the usual fine-tuning of the back-end downstream network architecture with front-end BERT pre-trained model for the target task, pre-training BERT with target domain data is a natural notion. Their work confirms that further target-domain specific pre-training of the BERT pre-trained model can significantly boost its performance for the target task. The fact based on their outcomes is the motivation for using domain-adapted BERT model in our work.

2.4. NER in Food Domain

There are a few works on the NER task in food domain such as drNER [26] and FoodIE [27], which utilized the rule-based computational linguistics and semantic information to characterize each food and dietary notion.

To the best of our knowledge, no NER work has been reported in the field of food domain using current state-of-the-art pre-trained language models and deep learning- based approaches. Unlike FoodIE [27], where rules consider word chunking while extracting and annotating the food concepts, our MenuNER work focuses on extracting well-chunked food menus mentioned in the text rather than food concepts.

3. Methodology

Our implementation of the menu named-entity recognition task (MenuNER) is a two-step process. First, the pre-trained weights of the off-the-shelf BERT language model are fine-tuned on a menu domain-specific corpus to learn the contextual representation of menu entities in the domain language model, as shown in Figure 1. In the second step, we fine-tune the downstream MenuNER task on Bi-LSTM-CRF-based model by utilizing hidden feature vectors from the BERT-domain language model (obtained from the first step) with character-level features and POS tag representations on labeled data for the menu named-entity recognition task.

3.1. BERT Training

BERT is highly bi-directional, learns the context of a word from its surroundings (left and right). BERT proposes a new training objective: the "Masked Language Model (MLM)" and "Next Sentence Prediction (NSP)". The MLM randomly masks some of the tokens from the input, and the task is to predict the original vocabulary IDs of the masked word from its context [12]. Given a review example from the food domain corpus: "Their yellow [MASK] is my favorite". The pre-trained BERT model on Books and Wikipedia corpus could predict the masked output as "color", but in the food domain, it could guess as a "chicken" and identify "yellow chicken" as a menu dish entity which is the expected output in our case. The objective of the next sentence prediction is to predict if the second sentence in the pair is the subsequent sentence in the text. Based on the above two purposes, a domain-based pre-trained BERT can capture more accurate context and long-term dependencies from the text. In this work, we adapted the BERT-base model from HuggingFace [28].

BERT-base (cased): 12 layers of transformer encoders, 768 hidden layers and 12 attention heads, with 110M trainable parameters in total.

3.2. BERT-Domain; Further Training of BERT Language Model

The general-domain corpora (Wikipedia and Book corpus) were used to train the pre-trained BERT language model. In domain-specific tasks, the data distribution of the target domain dataset may be different from the pre-trained BERT model. However, the BERT model can be further pre-trained on the target domain dataset with masked language model and next sentence prediction task to learn the better representation of domain data.

We further pre-train the BERT-base language model on food domain corpus to capture the contextual meaning of the menu entities and to enhance the performance on the target domain MenuNER task. The preparation of domain corpus for BERT pre-training is described in Section 4.1.1.

The BERT pre-training for language model on target domain requires the input samples in the same format as in MLM and NSP training in the original BERT-base model, where two sequences S_A and S_B are represented as "*[CLS]* S_A *[SEP]* S_B *[SEP]*", where *[CLS]* and *[SEP]* are special tokens.

As a result, a new target domain-adapted BERT model (we call it BERT-domain, hereafter) is obtained with updated weights for effective recognition of target domain entities (food menu entities).

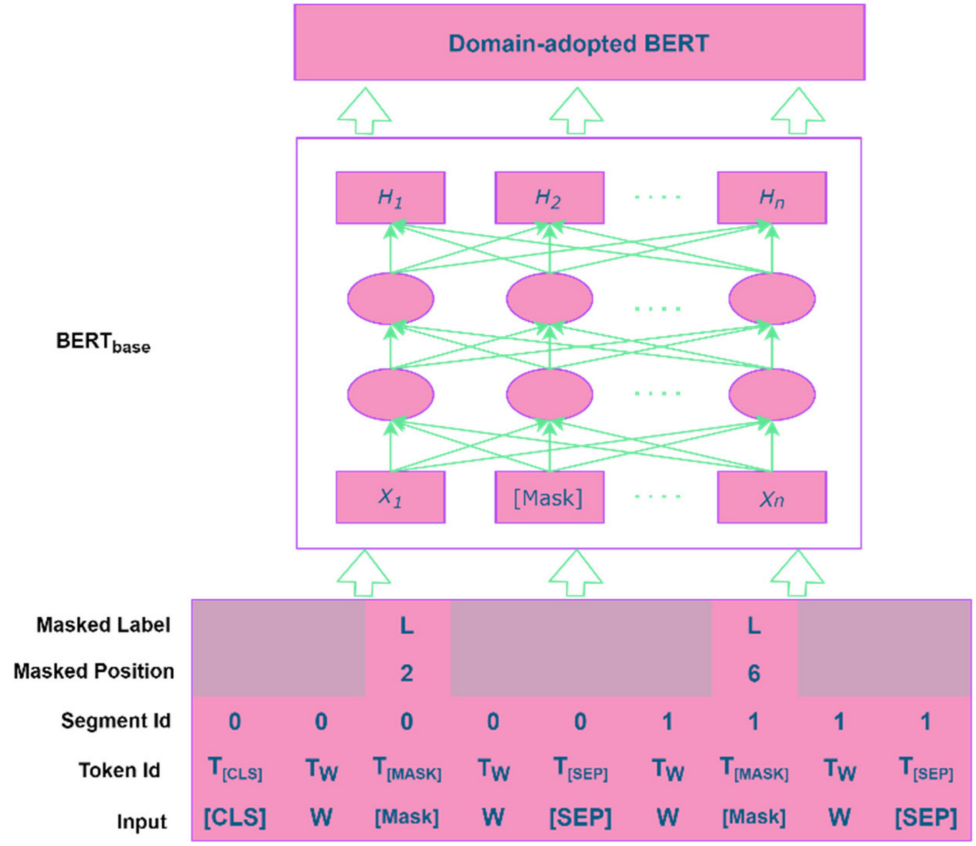

Figure 2. BERT post-training on domain corpus.

3.3. Neural Network Architecture for NER Task Fine-Tuning

The food menu entity recognition is a relatively challenging task as it requires recognizing unseen, polysemy tokens. To handle the challenges imposed by data limitation and user-generated text, the stacked Bi-LSTM-based model architecture with character-level and POS tag feature representations in addition to word embedding from BERT-domain, as shown in Figure 3, is proposed.

The following section discusses the overall network architecture by first describing the embedding layer and next, the neural network model.

3.3.1. Embedding Layer

The embedding layer transforms words, characters and POS representation from an input sentence into dense feature vectors where a vector represents the projection of the word into a continuous vector space. The position of a word in the learned vector space is referred to as its embedding [29]. In our work, we utilize POS tag, word- and character-level embedding features.

For a given sentence of word sequence $S = \{w_1, w_2, \ldots, w_k\}$, the dense embedding feature vector is created by concatenating character-level feature $e_i^c \in \mathbb{R}^{d_c}$, word-level feature $e_i^w \in \mathbb{R}^{d_w}$ and POS tag feature $e_i^p \in \mathbb{R}^{d_p}$. Note that the word embedding feature e_i^w is obtained from the pre-trained BERT-domain model. The final embedding vector can be formulated as $X_i = e_i^c \oplus e_i^w \oplus e_i^p$, where $i \in \{1, 2, \ldots, k\}$; e_i^c, e_i^w and e_i^p are the character, word and POS feature embeddings, respectively, and d_c, d_w and d_p are the dimensions of character, word and POS feature vector space, respectively; $\oplus$ denotes the concatenation operation.

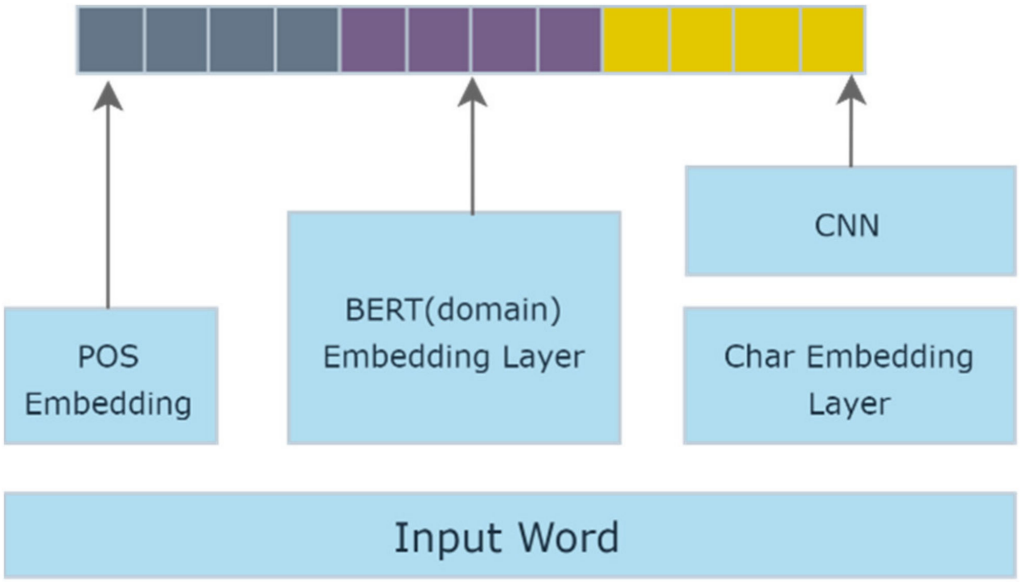

Figure 3. Embedding vector from different level of representations.

Word Embedding (BERT Embedder)

Word embedding is a powerful technique for capturing a word's semantic and syntactic meanings. It has been a common technique and is widely utilized for a variety of natural language processing applications, such as named-entity recognition. Furthermore, the embeddings from domain-adapted model (BERT-domain) can capture higher semantic and syntactic meanings of a word for the target domain.

To obtain word embedding, we adapt the BERT-domain model. The embedding size is 12 × 768 for each word token with a padded sequence length of 512. We utilize the feature-based BERT-domain model to obtain only the embeddings of the input text by freezing all the layers.

For feature selection from the BERT-domain model, the BERT authors examined that concatenating the last four layers produces the best results for feature-based approach, which is also observed through our experiments. We adapt this setting by applying mean pooling on the last four hidden states. We denote a word represented by a word embedding as $e_i^w \in \mathbb{R}^{d_w}$; for each word w_k in a sequence S, where d_w is the dimension of embedding vector space.

Character-Level Embedding (CharCNN: Character-level Convolutional Neural Network)

Character-level features have shown to be effective for extracting morphological information [8] and have been adopted to improve the learning of misspelling and out-of-vocabulary word representations from the user-generated text (i.e., user reviews). Therefore, we employ a convolutional neural network (ConvNets) [30] to adopt character-level representation of words.

Let C be the character vocabulary of a word token with a vector of dimension d, we compose the character embedding matrix $X_{emb} \in \mathbb{R}^{d \times C}$. Then, we perform a convolution operation between X_{emb} and a filter (kernel) $H \in \mathbb{R}^{d \times w}$ to extract features map X_f from the word. Each filter has a shape (w_i, d) where w_i is the filter size of the ith filter:

$$X_f = Conv1D(X_{emb}) \in \mathbb{R}^{C-w_i+1} \quad (1)$$

$$X_p = MaxPool\Big(ReLU\Big(X_f\Big)\Big) \quad (2)$$

$$e_i^c = concat(Xp) \quad (3)$$

We use nonlinear activation function ReLU following the max-over-time-pooling to each feature map X_f. Finally, we concatenate the pooled output to obtain the character-level representation of a word.

Part-Of-Speech (POS) Tag Feature Embedding

We use Part-Of-Speech (POS) tag features to extract syntactic information in our network model for the NER task. Syntactic features help the model to disambiguate the word based on the grammatical composition and pattern in the sentence. Named-entity recognition (NER) task aims to locate entities and classify them into some predefined categories. The POS tagging (like noun, pronoun, verb, etc.) can enable NER task to improve their decision. For each word w_k in a sequence S, a lookup layer maps each word w_k from one-hot vector to a low-dimensional vector $e_i^p \in \mathbb{R}^{d_p}$ where d_p is the dimension of the POS embedding.

3.3.2. NER Task Network Architecture

We construct a Bi-LSTM-CRF network architecture shown in Figure 4, for fine-tuning the downstream MenuNER task, which consists of two layers: Bi-LSTM layer—captures the long-term dependencies and semantic information from input sequence; CRF layer—labels the tokens based on output probabilities. We also construct a variant of Bi-LSTM-CRF model by applying multi-head self-attention layer on the top of stacked Bi-LSTM.

In the input embeddings, we extract the feature vector of each word w_k from the sequence by combining word- and character-level features and auxiliary POS tag features as described previously and fed into the stacked Bi-LSTM layer.

Bidirectional Long short-term memory (Bi-LSTM) Layer

LSTM-based networks are capable of learning long-range dependencies and prevent gradient vanishing problem. Bi-LSTMs enable the hidden states to make full use of the context information of the input sequence. A single LSTM memory unit consists of an input gate (i_t), output gate (o_t), forget gate (f_t) and a memory cell (c_t), and produces a hidden vector of LSTM cell unit using the following mathematical formulation:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma\Big(W_{xf}x_t + W_{hf}h_{t-1} + b_f\Big) \tag{5}$$

$$g_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{7}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{8}$$

$$h_t = o_t \odot \tan\mathrm{h}(c_t) \tag{9}$$

where σ and tan h denotes the logistic function and a hyperbolic tangent function, respectively, and $\odot$ is a Hadamard product. W_i, W_f, $W_o \in \mathbb{R}^{\mathrm{d}\times 2\mathrm{d}}$ are the weighted matrices.

Our network uses the stacked Bi-LSTM by concatenating the forward $\overrightarrow{h_t}$ and backward $\overleftarrow{h_t}$ states. The forward LSTM layer extracts the past information while the backward layer captures the future information of the sequence and produces the final output $y_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right]$. In the stacked Bi-LSTM network, the output y_t from the lower layer feeds into the upper-level Bi-LSTM layer. We process Bi-LSTM with 512 hidden units so that the output representation dim is 1024 for each token in each stacked layer of the network with dropout 0.3. This recurrent layer learns the dependencies between tokens in an input sequence.

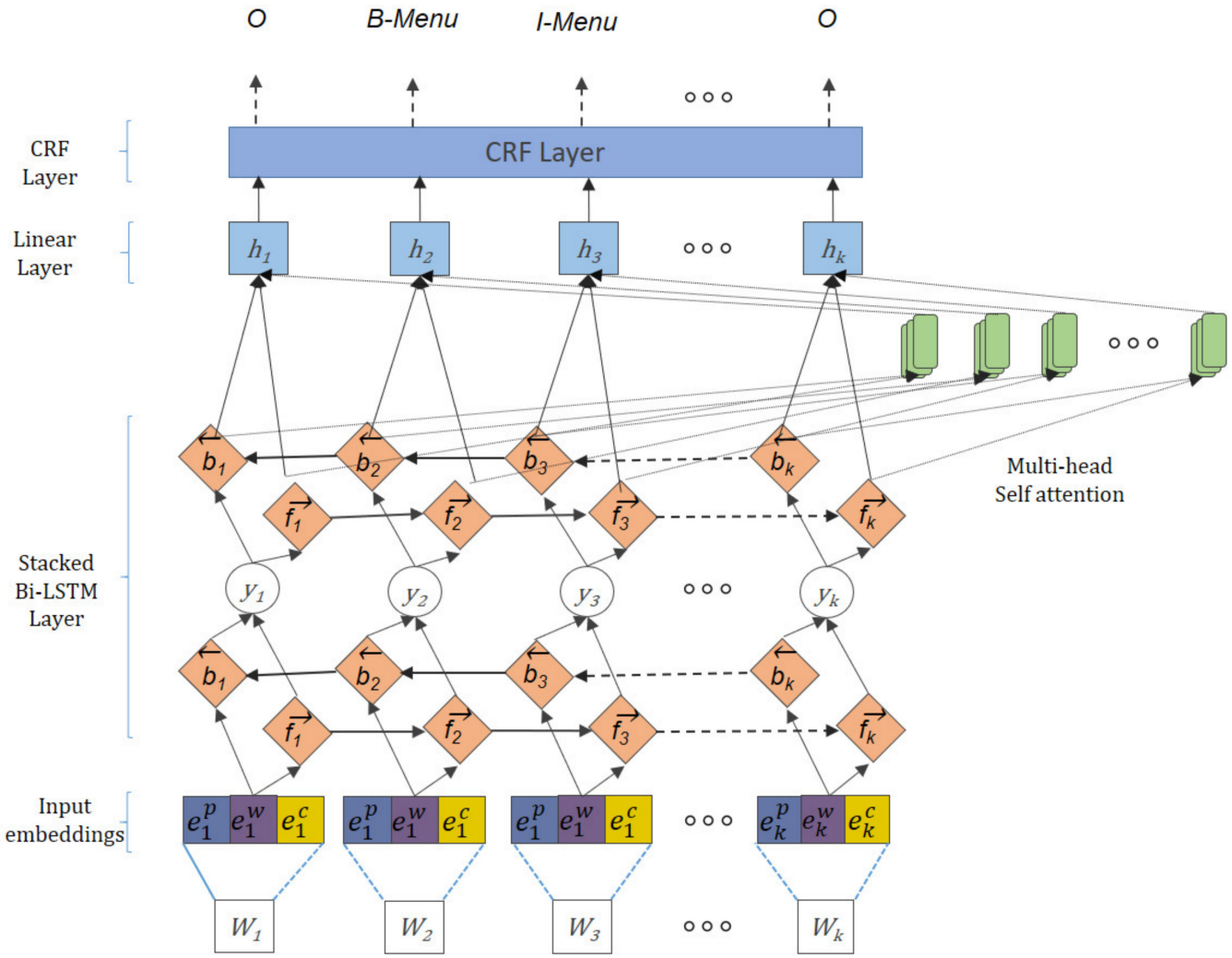

Figure 4. The proposed Bi-LSTM-CRF neural network with a variation of multi-head self-attention for MenuNER task, fine-tuned on domain specific BERT.

Conditional Random Field (CRF) Layer

We adopt the Conditional Random Field (CRF) model for our prediction task. CRF is based on conditional probability model that is used for labeling sequential data.

For an input sequence $X = \{x_1, x_2, \ldots, x_n\}$, where x_n is the input vector of the nth word and $y = \{y_1, y_2, \ldots, y_n\}$ is the corresponding label sequences for X. We have $n \times l$ matrix P, obtained from the previous stacked Bi-LSTM layer, where n is the number of input word and l is the number of distinct tags, $P_{i,j}$ is the probability of label i of the word j in the sentence. The score is computed as:

$$s(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=0}^{n} P_{i, y_i} \tag{10}$$

where $A_{i,j}$ is the likelihood of transition from the tag y_i to tag y_j. The y_0 and y_n are the first and last tags of a sentence, respectively, that we add to a list of possible tags. We then obtain the normalized probability of the label sequence y and output the label sequence $\overline{y}$ by using softmax function:

$$p(y|X) = \frac{e^{s(X, y)}}{\Sigma_{\overline{y} \in Y_X} e^{s(X, \overline{y})}} \tag{11}$$

where for a given sentence $X, \ldots, Y_X$ represents all conceivable tag sequences. We use the loss function defined below to optimize the log-likelihood of the correct label sequence for CRF training:

$$\log(p(y|X)) = s(X, y) - \log\left(\Sigma_{\overline{y} \in Y_X} e^{s(X, \overline{y})}\right) \quad (12)$$

$$y^* = argmax_{\overline{y} \in Y_X} s(X, \overline{y}) \quad (13)$$

Finally, the model uses the Viterbi algorithm using Equation (13) to predict the optimal score.

4. Evaluation on Food Domain

In this section, we discuss the experiments and results of our models.

4.1. Experimental Environments

4.1.1. Dataset for MenuNER and Fine-Tuning

Our experiments used the YELP dataset [31] and processed reviews only from the restaurant category to make food domain dataset. A corpus with 15,000,000 sentences was prepared from the YELP dataset for further pre-training of the BERT-base language model to adopt the domain language model. Although a larger corpus would yield better results, we limit the number of sentences due to computational constraints.

To prepare the training corpus from the raw data, which consist of user-generated reviews, we first performed sentence segmentation and normalization using the Spacy library [32]. Then, we pre-generated five arbitrarily processed corpora for further pre-training of the BERT model. Each generated document contains tokens, segment IDs' masked labels, masked label's position and a flag indicating whether the next segment in the masked language model is random.

To fine-tune the downstream MenuNER task, a carefully handcrafted menu dataset is created in IOB (Inside, Outside, Beginning) format (Table 1) using the WebAnno2 tool [33]. We use NLTK library [34] for POS tagging and annotation. We divide dataset into train, test and valid file for training and evaluation. We also validate our dataset using the proposed deep learning model and manually identify the TP, TN, FP, and FN labels to assess human error in the annotation process.

Table 1. Description of annotated menu dataset.

Dataset Features	Train	Test	Valid
No. of sentences	5517	3037	2211
Unique menu entities	1027	637	516
Total menu entities	2626	1424	1106

4.1.2. Experiment Metrics

The evaluation and ablation study measures of the Bi-LSTM-CRF model are carried out on the basis of three metrics scores: precision, recall, and the F1:

$$Precision(P) = \frac{TP}{TP + FP} \quad (14)$$

$$Recall(R) = \frac{TP}{TP + FN} \quad (15)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (16)$$

Here, *TP* (True Positive) and *FP* (False Positive) represent the number of a menu label, which is predicted as a *Menu* label, and the number of a non-entity label *O*, which is predicted as a *Menu* label. *TN* (True Negative) and *FN* (False Negative) represent the number of a non-entity label *O* which is predicted as a non-entity label *O*, and the number

of the *Menu* label, which is predicted as a non-entity label *O*, respectively. The F1-score is obtained by calculating the harmonic average of precision (P) and recall (R).

4.1.3. Experiment Setting

The BERT language model is further pre-trained on five arbitrarily generated restaurant corpora to obtain the BERT-domain model. We train the BERT MLM on food corpus for 80,000 steps, which took (34,708.82 s) ~9.5 h for 3 epochs on Nvidia TITAN X GPU. We use Adam optimizer with 1×10^{-8} and a learning rate of 3×10^{-5}. Figure 5 shows the training loss at every 50 steps.

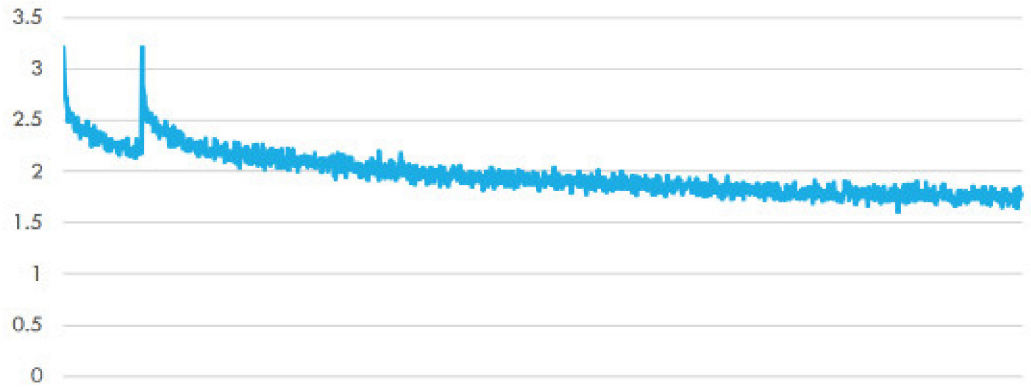

Figure 5. Training loss at every 50 steps of BERT-domain specific fine-tuning.

For fine-tuning the MenuNER task using BiLSTM-CRF network, we use Adam optimizer with a learning rate of 1×10^{-3}, and Adam epsilon (ε) 1×10^{-8}, weight decay of 0.01. The model training was performed on Nvidia TITAN X GPU with a batch size of 64. We set the number of epochs equal to 30 and the patience value of 7 for early stopping criteria if the model ceased to decrease.

In our model, we adopt setting with mean pooling the hidden outputs from the last four layers of the pre-train BERT model and apply this setting for all the downstream fine-tuning tasks and model evaluation experiments.

4.2. Experimental Results

4.2.1. Ablation Test

We performed ablation experiments to investigate the model behavior with different features and components. For ablation experiments, we apply 'Bi-LSTM-CRF' network model by utilizing the BERT-domain model and use default parameter settings as used in the downstream fine-tuning NER task. Table 2 summarizes the ablation experimental results, which show the impact of additional embedding features (Character-level CharCNN feature and POS tag feature) in the network model.

Table 2. Ablation test to investigate the impact of hidden features in our model utilizing BERT (feature-based).

Embeddings	F1 Score (Test)	Training Time (s)	Test Execution Time (s)
BERT-domain	91.70	2851.52	61.92
BERT-domain + POS	91.74	4156.91	62.40
BERT-domain + CharCNN	91.85	2484.65	62.01
All (BERT-domain + CharCNN + POS)	92.46	4827.43	64.14

The results of the ablation tests in Table 2 indicate that when POS tag embedding is used in conjunction with character-level embedding, a synergistic effect occurs. The training time, on the other hand, varies depending on the combination of features added.

Due to the small size of the annotated dataset, all embedding results converge with a minor distinction in performance. The results, however, indicate that character-level features significantly improve the model's performance on user-generated text and enable it to recognize misspelled and unique word contexts. Additionally, the character-level embedding exhibits the quickest convergence. Moreover, the POS tag features demonstrate an increase in the ability to capture multi-word menu entities; when all feature vectors are combined (BERT-domain + POS + CharCNN), the optimizer takes somewhat longer to optimize and capture multiple features during the training phase, but the execution time on the test dataset has a negligible influence yet delivers synergistically better outcomes. Figure 6 shows the model training results and convergence for various embedding combinations.

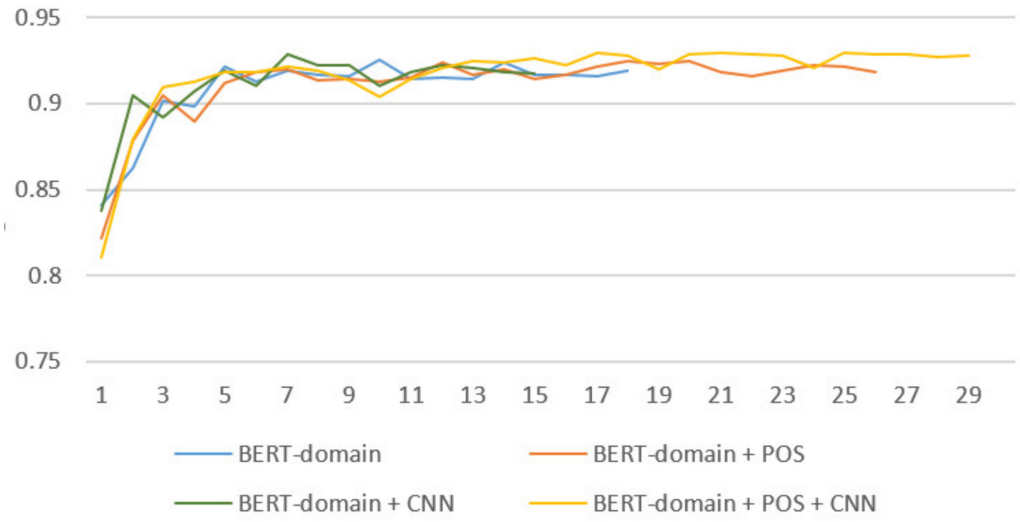

Figure 6. Model training with different embedding combinations.

4.2.2. Evaluation for MenuNER

In our experiments, we tried to examine how the domain-adapted BERT language model improves performance in MenuNER task by considering the following factors:

- What is the effect of domain adaptation (further pre-training of BERT-base model) on a MenuNER task in terms of its performance?
- What impact does a feature-based BERT model put on the NER task while combining with character-level and POS tag features? (see Table 2)
- Comparison of neural network architectures for the MenuNER task: "BERT with fully-connected FC layer and a SoftMax layer" vs. "Bi-LSTM-CRF" and a variant with multi-head self-attention.

For the MenuNER task, various network model variants are compared to BERT-base and BERT-domain features. The results of Table 3 demonstrate that the 'Bi-LSTM-CRF'-based neural network model when used with BERT-domain embedding as a feature produces effectively better results. In the first NER network model, we fine-tune BERT with a SoftMax layer. It is intriguing to see that fine-tuning the BERT model using a simple neural network results in improved performance. However, it lacks in identifying informal user-generated text and multi-word menu entities as shown in Table 4. To tackle this issue, we employ feature-based BERT with character-level and POS tag features in the Bi-LSTM-CRF and its multi-head attention network variant. However, because the BERT low-level representation and the Bi-LSTM upper-level layers already incorporate contextual information such as multi-head attention, the effect of the multi-head attention network does not appear compelling in our experiment.

Table 3. Summary of experimental results for MenuNER with (BERT-base-cased) and target domain-adapted version (BERT-domain) on NER task model. We utilize feature-based BERT model with extended features for NER network 2 and 3 while fine-tuning BERT model on downstream dataset for network 1.

#	Word Embedding	BERT-Base Embedding			BERT-Domain Embedding		
	NER Network	**P**	**R**	**F1**	**P**	**R**	**F1**
1	SoftMax	88.7	91.7	90.2	91.7	92.1	91.9
2	Bi-LSTM-CRF	90.3	91.3	91.8	**92.7**	**92.3**	**92.5**
3	Bi-LSTM-Attn-CRF	92.2	90.5	91.4	91.0	92.1	91.6

Table 4. Menu entities' prediction from applying BERT-base features vs. BERT-domain features to the deep learning model.

#	Review Text	BERT-Base	BERT-Domain
1	I tried their *wazeri platter* which is *grilled chicken*, *beef kafta kabab* and *rice*.	wazeri platter, chicken, beef kafta kabab, rice	wazeri platter, grilled chicken, beef kafta kabab, rice
2	I never finish the *rice* and always ask for a takeaway container for the *naan* + leftover *rice*.	rice, naan, leftover rice	rice, naan, rice
3	We both ordered *green tea* and it had a very unique taste.	tea	green tea
4	The *red spicy tomato-based sauce* on the side was delicious.	sauce	red spicy tomato-based sauce
5	I would say it's like the size of 2 medium sized *chicken breasts* for the order.	chicken	chicken breasts
6	Also check out *Firni* (a *sweet custard*) for desert	sweet custard	Firni, sweet custard
7	*Chicken* is well cooked tooOmg, super delicious *kebabs*.	chicken, tooOmg, kebabs	Chicken, kebabs
8	It's called *Double Ka Meeta* and we loved it!		Double Ka Meeta
9	Dessert options were *kesari*, *gajar halwa*, *strawberry shortcake* and *chocolate mousse*.	kesari, gajar halwa, strawberry shortcake	kesari, gajar halwa, strawberry shortcake, chocolate mousse

Table 4 compares NER task model outputs when utilizing word features from BERT-base and BERT-domain model. From examples 1, 2, 3, and 4, it is shown that the pre-trained domain model learns better dependencies between menu token pairs. As in examples 3 and 5, in general, the context of the words "*breasts*" and "*green*" are related to body part and color, respectively; however, the domain-adapted model could recognize it as a menu name from the context. In example 6, "*Firni*" (sweet pudding), is a unique word which does not exist in the training or valid dataset. However, the pre-training domain model recognizes and learns from its context. From example 7, the misspelled out-of-vocabulary (OOV) word "*tooOmg*" is recognized as the menu entity by the base model, which is not valid.

5. Conclusions and Future Work

This paper proposed a simple but effective approach for the domain task where training annotated data is limited or not available. We apply domain adaptation technique to further pre-train the BERT language model on domain-specific raw text (unlabeled review dataset from YELP) for contextual learning. As proof of the effectiveness of our NER approach, we performed experiments on the food menu named-entity recognition (MenuNER) task on annotated dataset. We manually labeled menu entities from the user review dataset provided by YELP. We first further pre-trained the off-the-shelf BERT-base model on a food domain corpus to obtain a domain-adapted BERT language model, BERT-domain. Next, we fine-tuned the downstream MenuNER task using the Bi-LSTM+CRF model with a concatenated feature vector consisting of word embedding from BERT-domain, character embedding, and POS tag features.

We investigated the effect of applying fine-tuned domain-specific BERT language model on down-stream task performance. We found that the off-the-shelf BERT model, when further pre-trained on domain-specific corpus, can learn new context and perform even better and can be used for "feature extraction" or "fine-tuning" downstream tasks.

Entity extraction is the first and essential step in the information extraction process. Our work extensively contributes to the field of food domain and can be extended to solve many problems in the food domain. As a future direction, our work could lead to extending many IE and entity-linking tasks including creation of food knowledge graph, menu entity-level sentiment analysis from text and menu-based text summarization from raw data, etc.

Author Contributions: Conceptualization, M.H.S.; methodology, M.H.S.; software, M.H.S.; validation, M.H.S.; formal analysis, M.H.S.; data curation, M.H.S.; writing—original draft preparation, M.H.S.; writing—review and editing, S.-T.C.; supervision, S.-T.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Technology development Program (No. S2953591 and No. S3044682) funded by the Ministry of SMEs and Startups (MSS, Korea).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset used in this work can be found here: dataset-Google Drive; accessed on 18 June 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lin, B.Y.; Xu, F.; Luo, Z.; Zhu, K. Multi-channel BiLSTM-CRF Model for Emerging Named Entity Recognition in Social Media. In Proceedings of the 3rd Workshop on Noisy User-generated Text, Copenhagen, Denmark, 7 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 160–165.
2. Yan, H.; Deng, B.; Li, X.; Qiu, X. TENER: Adapting Transformer Encoder for Named Entity Recognition. *arXiv* **2019**, arXiv:1911.04474.
3. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
4. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the 18th International Conference on Machine Learning, Williams College, Williamstown, MA, USA, 28 June–1 July 2001; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001; pp. 282–289.
5. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
6. Ma, X.; Hovy, E. End-to-end sequence labeling via bidirectional LSTM-CNNS-CRF. *arXiv* **2016**, arXiv:1603.01354.
7. Lample, G.; Ballesteros, B.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural Architectures for Named Entity Recognition. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 260–270.
8. Chiu, J.P.C.; Nichols, E. Named Entity Recognition with Bidirectional LSTM-CNNs. *TACL* **2016**, *4*, 357–370. [CrossRef]
9. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.
10. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
11. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
12. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
13. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. How to Fine-Tune BERT for Text Classification? In *CCL, Lecture Notes in Computer Science*; Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y., Eds.; Springer: Cham, Switzerland; Kunming, China, 2019; Volume 11856, pp. 194–206.
14. Zhu, J.; Xia, Y.; Wu, L.; He, D.; Qin, T.; Zhou, W.; Li, H.; Liu, T. Incorporating BERT into neural machine translation. In Proceedings of the 18th International Conference on Learning Representations (ICLR), Virtual Conference, Formerly Addis Ababa Ethiopia. 26 April–1 May 2020.

15. Symeonidou, A.; Sazonau, V.; Gorth, P. Transfer Learning for Biomedical Named Entity Recognition with BioBERT. *SEMANTICS Posters Demos* **2019**, *2451*, 100–104.
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 2017, pp. 5999–6009.
17. Rongali, S.; Jagannatha, A.; RAWAT, B.P.S.; Yu, H. Continual Domain-Tuning for Pretrained Language Models. *arXiv* **2021**, arXiv:2004.02288.
18. Ma, X.; Xu, P.; Wang, Z.; Nallapati, R.; Xiang, B. Domain Adaptation with BERT-based Domain Classification and Data Selection. In Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019), Hong Kong, China, 3 November 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 76–83.
19. Xu, H.; Liu, B.; Shu, L.; Yu, P.S. 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: NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 2324–2335.
20. Rietzler, A.; Stabinger, S.; Opitz, R.; Engl, S. Adapt or get left behind: Domain adaptation through BERT language model fine-tuning for aspect-target sentiment classification. *arXiv* **2019**, arXiv:1908.11860.
21. Seok, M.; Song, H.J.; Park, C.; Kim, J.D.; Kim, Y.S. Named Entity Recognition using Word Embedding as a Feature. *Int. J. Softw. Eng. Appl.* **2016**, *10*, 93–104. [CrossRef]
22. Saad, F.; Aras, H.; Hackl-Sommer, R. Improving Named Entity Recognition for Biomedical and Patent Data Using Bi-LSTM Deep Neural Network Models. In *Natural Language Processing and Information Systems (NLDB), Lecture Notes in Computer, Science*; Métais, E., Meziane, F., Horacek, H., Cimiano, P., Eds.; Springer: Cham, Switzerland, 2020; Volume 12089.
23. Zhai, A.; Nguyen, D.Q.; Akhondi, S.; Thorne, C.; Druckenbrodt, C.; Cohn, T.; Gregory, M.; Verspoor, K. Improving Chemical Named Entity Recognition in Patents with Contextualized Word Embeddings. In Proceedings of the 18th BioNLP Workshop and Shared Task, Florence, Italy, 1 August 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 328–338.
24. Emelyanov, A.; Artemova, E. Multilingual Named Entity Recognition Using Pretrained Embeddings, Attention Mechanism and NCRF. In Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing, Florence, Italy, 2 August 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 94–99.
25. Francis, S.; Landeghem, V.J.; Moens, M. Transfer Learning for Named Entity Recognition in Financial and Biomedical Documents. *Information* **2019**, *10*, 248. [CrossRef]
26. Eftimov, T.; Seljak, B.K.; Korošec, P. DrNER: A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PLoS ONE* **2017**, *12*, e0179488. [CrossRef] [PubMed]
27. Popovski, G.; Kochev, S.; Seljak, B.K.; Eftimov, T. Foodie: A rule-based named-entity recognition method for food information extraction. In Proceedings of the 8th International Conference on Pattern Recognition Application and Methods (ICPRAM), Prague, Czech Republic, 19–21 February 2019; pp. 915–922.
28. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv* **2019**, arXiv:1910.03771.
29. Zhang, F. A hybrid structured deep neural network with Word2Vec for construction accident causes classification. *Int. J. Constr. Manag.* **2019**, 1–21. [CrossRef]
30. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. In Proceedings of the 28th International Conference on Neural Information Processing System, Montreal, QC, Canada, 7–12 December 2015; Volume 1, pp. 649–657.
31. YELP Dataset. Available online: https://www.yelp.com/dataset/ (accessed on 14 May 2021).
32. Spacy. Available online: https://spacy.io/ (accessed on 14 May 2021).
33. WebAnno Tool. Available online: https://webanno.github.io/webanno/ (accessed on 14 May 2021).
34. Loper, E.; Bird, S. NLTK: The natural language toolkit. *arXiv* **2002**, arXiv:cs/0205028.