classification, CRF was employed to label opinion holders. Seki et al. [3] considered opinion holder detection as a binary classification between author and authority for Japanese and English texts. Opinion sentences were classified by analyzing different features associated with the author and authority. In this case, an authority refers to any third party who expresses rules, advice, or orders regarding on opinion text. Various key phrases ("may be", "can say") and terms ("should", "must") tend to be used in author opinion, rather than authority, opinion sentences. After the classification, author and authority holders were detected according to specific rules and patterns. However, the construction of these rules and patterns is costly. To address this problem, we applied deep learning approaches based on pre-trained language models.

One role of the INSIDE/OUTSIDE classification is to evaluate the text's informativeness. The following examples illustrate this concept.

1. Consider the following sentences:

   e3) This laptop is not good for me.

   e4) They said this laptop is poor.

   e5) I heard this laptop is good.

   All three sentences belong to the INSIDE class, as the opinion holder words (me, They, I) appear within them. In the case of e3, the opinion target is a laptop, the opinion phrase is "not good," and the sentiment is negative. The opinion holder of e3 is "me," who is also the writer. We regard e3 as informative because this information is true for the writer.

   The opinion target, opinion word, and sentiment in e4 are laptop, poor, and negative, respectively. However, the opinion holder of e4 is "they," which refers to an external source. Because this statement belongs to someone other than the writer, the informativeness of e4 is lower than that of e3 (i.e., in the middle).

   The opinion holder candidate in e5 that expresses a positive sentiment for the laptop is "I." However, the source of information is unknown. Therefore, e5 represents hearsay information with a low informativeness.

2. Consider the following statement:

   e6) This laptop is good.

   The class of e6 is OUTSIDE, as the opinion holder is unknown. Any statement labeled OUTSIDE has a low informativeness.

Figure 1 illustrates the merit of our task. By detecting opinion holders through INSIDE/OUTSIDE classification, we can obtain reliable information. This information can then be applied to multiple tasks such as summarization and fact-based opinion analysis. By discarding any unreliable information, the overall consensus becomes more appropriate. For example, the consensus between e3, e4, and e5 is "The laptop is not good."

Key contributions of our study are summarized as follows:

1. We define a new classification task for opinion holder detection using the INSIDE and OUTSIDE class labels.

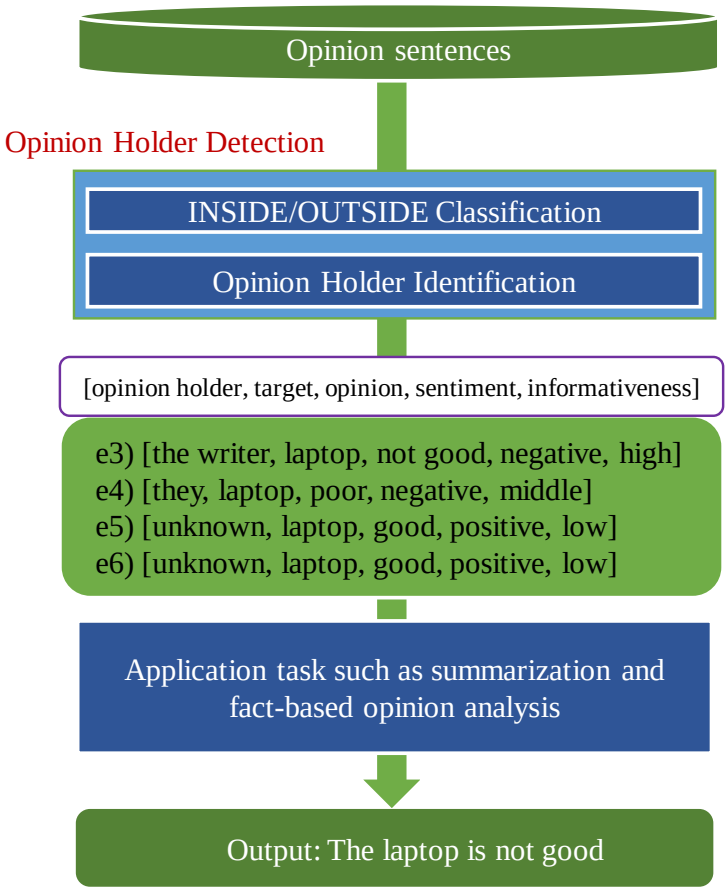

Figure 1: Flowchart of the opinion holder detection process.

2. We prepared a new English-language dataset to detect opinion holders. For this purpose, we employed ASTE-DATA-V2 [4] for the laptop domain, and annotated the data with the two class labels. Subsequently, we annotated IOB (Inside-Outside-Beginning) tags corresponding to opinion holders.

3. For the INSIDE/OUTSIDE classification task, we compared the performance of transformer-based deep learning pre-trained language models on our prepared dataset.

4. For the opinion holder identification task, we compared the performance of feature-based and fine-tuning-based architectures by applying deep learning-based pre-trained language models to our prepared dataset.

2 Related Work

The following section summarizes relevant prior studies and their drawbacks.

Seki et al. [3] examined noun phrases and linguistic features within Japanese and English texts, and used support vector machine (SVM) to classify opinion holders as authors

or authorities. Wu et al. [5] solved the co-reference resolution, and extracted opinion holders via rules involving punctuation marks, conjunctions, prefixes, identification of opinion holder 385 suffixes, and opinion operators. They achieved an F1 score of 0.825 in the NTCIR7 MOAT task on traditional Chinese data. Kim et al. [6] employed a maximum entropy model to extract opinion holders and targets from news articles. They first extracted key terms and labeled semantic roles, and then identified semantic roles corresponding to opinion holders and targets. Kim et al. [7, 8] classified opinion holders as authors, simple holders, or co-referenced holders from textual data. Subsequently, they extracted lexical and syntactic features for SVM to determine appropriate opinion holders for English texts. Xu et al. [9] examined words and parts of speech as features in an L2-norm linear SVM to solve an opinion holder detection problem as a similar method for named entity recognition. Choi et al. [10] and Breck et al. [11] utilized lexical, syntactic, dictionary-based, and dependency features by conditional random field (CRF). Meng et al. [12] used words, parts of speech, and opinion operators, whereas Liu et al. [13] extracted parts of speech, semantic features, contextual features, dependency features, and position features by CRF. Lu et al. [14] applied dependency parsing to Chinese texts. This approach yielded encouraging performance on opinion holder/target identification with NTCIR-7's traditional Chinese dataset and outperformed existing methods, including the CRF-based model. Elarnaoty et al. [15] employed CRF and semi-supervised pattern recognition techniques to examine Arabic texts. They investigated a comprehensive feature set to compensate for the lack of parsing structural outcomes. The study in question represents leading research for opinion holder extraction in Arabic news sources, independent from any lexical parsers. Wiegand et al. [16] utilized a convolutional kernel classifier, and introduced a new corpus focusing on the roles of opinion verbs from the subjectivity lexicon. Subsequently, they demonstrated the potential benefits of this corpus. Wiegand et al. [17] used k-nearest neighbor graphs to categorize opinion verbs among three different types. Each type had a characteristic mapping between semantic roles, opinion holders, and targets.

The aforementioned studies were mainly based on machine learning with feature extraction. In contrast, recent research trends have considered automatic feature engineering via deep neural networks. Traditional machine-learning-based models (e.g., SVM) require large amounts of training data, which are difficult and costly to prepare. To avoid this issue, we employed pre-trained models, including BERT by Devlin et al. [18], DistilBERT by Sanh et al. [19], and CSE by Akbik et al. [20].

Unlike the above mentioned existing studies, we defined the presence of opinion holders within texts as a binary classification task: the INSIDE class, and the OUTSIDE class. As discussed in Section 1, this definition yields several useful features, and represents a novel point of our study. Furthermore, we constructed a new English-language dataset for further use across various real-world applications.

3 Dataset Construction

Pontiki et al. [21, 22, 23] prepared datasets for SemEval 2014-2016. These datasets are generally used for sentiment analyses in the laptop, hotel, and restaurant domains. For our task, the laptop domain dataset was obtained from ASTE-DATA-V2 [4], which was prepared based on SemEval 2014-2016 [24]. However, because our objective entails the detection of the opinion holders, we applied INSIDE/OUTSIDE and opinion holder tags throughout the dataset.

The laptop domain of ASTE-DATA-V2 comprises 1,453 instances (i.e., opinion sentences). The original data encompasses three types of tag information for each sentence: the target position, opinion position, and sentiment. Consider the following sentence:

e7) It's a great product for a great price.

This sentence comprises eight tokens (subunits, such as individual words or terms). The target word "price" appears in the 8th index. The corresponding opinion word "great," which appears in the 7th index, implies a positive sentiment.[1] The class of e7 is OUTSIDE, as no opinion holder appears in the sentence. Because we denote the value of OUTSIDE by 0, the sentence is labeled with "[([7],[6],'POS'), 0]." We then annotated IOB tags for opinion holder identification. As the most widespread encoding scheme in sequence labeling tasks, IOB tagging represents spans by combining the positional and category tags. We used three tags for opinion holder identification: B-OH (beginning of opinion holder), I-OH (inside of opinion holder), and O (outside). The following labels were added to e7: It's/O a/O great/O product/O for/O a/O great/O price/O. Because no opinion holders appear in the sentence, every word was tagged with O. Consider another sentence:

e8) My laptop now has no battery.

This sentence consists of six tokens. Of the six tokens that appear in this sentence, the target is the word "battery," the opinion is the word "no," and the sentiment is negative.[2] The class of e8 is INSIDE because the opinion holder is referenced by the word "My." Because the INSIDE class is denoted by 1, the sentence is labeled by "[([5],[4],'NEG'), 1]." The IOB tagging is: My/B-OH laptop/O now/O has/O no/O battery/O. Consider another example of the INSIDE class that contains both B-OH and I-OH tags;

e9) My friend reports the notebook is astonishing in performance , picture quality , and ease of use.[3]

It is apparent that the opinion holder is writer's friend. Therefore, the corresponding IOB tagging: My/B-OH friend/I-OH reports/O the/O notebook/O is/O astonishing/O in/O performance/O ,/O picture/O quality/O ,/O and/O ease/O of/O use/O.

Our prepared dataset contains a total of 1,453 opinion sentences, wherein 37% belongs to the INSIDE class and 63% belongs to the OUTSIDE class. To verify the accuracy of annotations, we calculated the Cohen κ score and obtained a value of 0.667, which indicates a substantial strength of agreement. There is a total of 971 "B-OH" tags, 9 "I-OH" tags, and 24,036 "O" tags in our dataset. In the training set, there are 798 "B-OH" tags, 5 "I-OH" tags, and 19,940 "O" tags. In the validation set, there are 74 "B-OH" tags, 0 "I-OH" tags, and 1,809 "O" tags. In the testing set, there are 99 "B-OH" tags, 4 "I-OH" tags, and 2,287 "O" tags.

4 Proposed Method

4.1 INSIDE/OUTSIDE Classification

We employed two models to perform the INSIDE/OUTSIDE classification task: BERT with logistic regression (BERT+LR), and DistilBERT with logistic regression (DistilBERT+LR).

[1] In the dataset, it is expressed as "It's a great product for a great price####[([7],[6],'POS')]".

[2] In the dataset, it is expressed as "My laptop now has no battery####[([5], [4], 'NEG')]".

[3] In the dataset, it is expressed as "My friend reports the notebook is astonishing in performance , picture quality , and ease of use####[([8], [6], 'POS'), ([10, 11], [6], 'POS'), ([16], [14], 'POS')]".

BERT is a deep-learning-based pre-trained language model. DistilBERT is an extension of BERT that has 40% less parameters and runs 60% faster while preserving over 95% of the original BERT's performance.

Because DistilBERT retains the high performance of BERT on smaller datasets, we compared both models through a performance analysis.

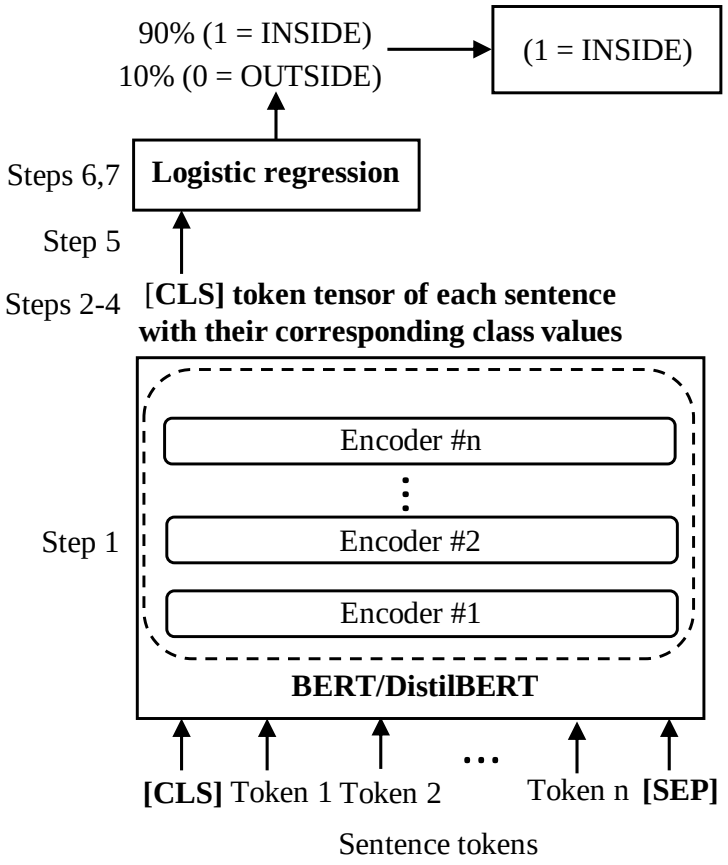

Figure 2: Flowchart of the INSIDE/OUTSIDE classification.

As shown in Figure 2, our INSIDE/OUTSIDE classification task comprises the following seven steps.

- Seven Steps:
  - Step 1: We use the language model to embed all opinion sentences of the dataset.
  - Step 2: For INSIDE/OUTSIDE classification, we are only interested in model's output for the [CLS] token tensor. Hence, we select that slice of the [CLS] tensor cube and discard other token tensors.
  - Step 3: We convert the [CLS] tensor to obtain the two dimensional feature array containing all sentences within our dataset. Each row corresponds to a sentence

in our dataset. Each column corresponds to the output of a hidden unit from the feed-forward neural network at the top transformer block of the model.

- Step 4: We map binary class values of each sentence to their respective two dimensional array features.
- Step 5: We allocate the data among training/(validation)/testing sets for logistic regression.
- Step 6: We train the logistic regression classifier using the training set.
- Step 7: We test the logistic regression classifier using the testing set.

4.2 Opinion Holder Identification

For opinion holder identification, we employed both feature-based and fine-tuning-based architectures, two standard NER (i.e., sequence labeling) architectures commonly used in the literature [25]. In these architectures, we employed contextual string embedding (CSE) and conditional random field (CRF) with BERT and DistilBERT. Here BERT, DistilBERT, and CSE were used for contextual word embedding, and CRF was employed for the sequence labeling task; i.e., opinion holder identification. For feature-based architecture, we utilized the following five models: CSE+CRF, BERT+CRF, (BERT&CSE)+CRF, DistilBERT+CRF, and (DistilBERT&CSE)+CRF. For fine-tuning architecture, we utilized the following six models: BERT, BERT+CRF, (BERT&CSE)+CRF, DistilBERT, DistilBERT+CRF, and (DistilBERT&CSE)+CRF.

4.2.1 Contextual String Embedding (CSE)

CSE is a contextualized embedding for any string of characters in a sentential context. In this approach, each sentence is passed as sequences of characters into a character-level language model (pre-trained bidirectional character language model on large unlabeled corpora) to form word-level embedding, as shown in Figure 3.

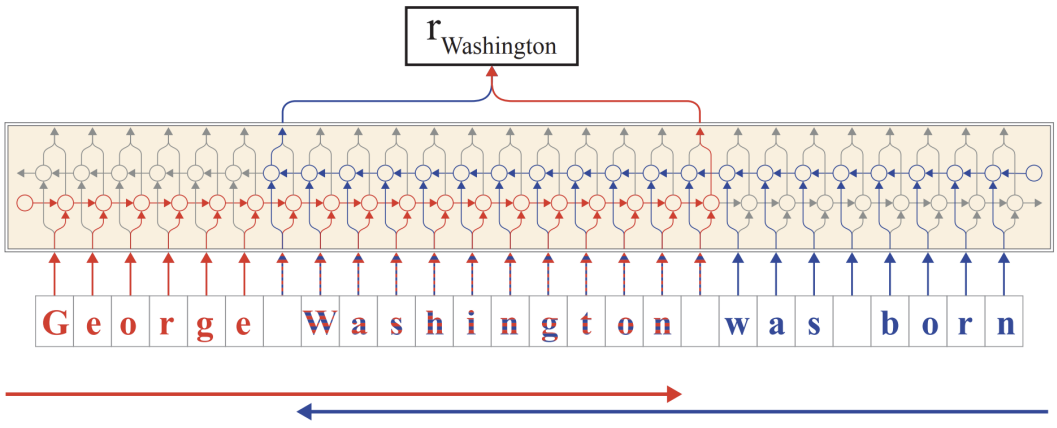

Figure 3: Extraction of a CSE for the word (“Washington”). The image is taken from Akbik et al. [20].

From the forward language model (denoted by red in Figure 3), the hidden output state is extracted after the last character in the word. This hidden state contains information

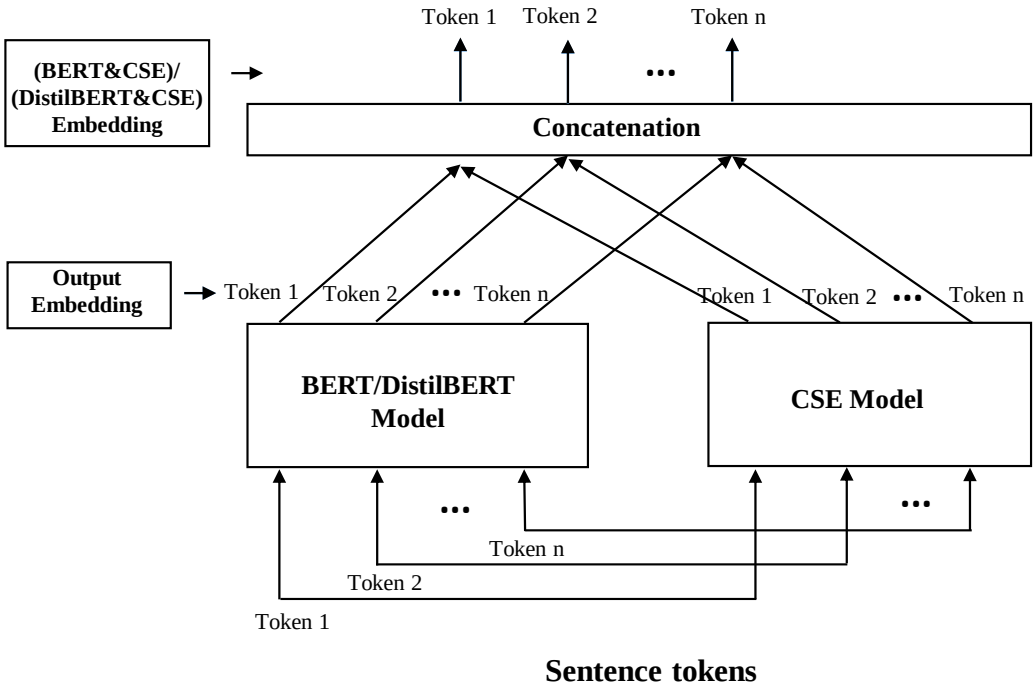

Figure 4: Concatenation of BERT/DistilBERT embedding with CSE for each token in a sentence.

propagated from the beginning of the sentence up to this point. From the backward language model (denoted by blue in Figure 3), the hidden output state is extracted before the first character in the word. This state contains information propagated from the end of the sentence to this point. Both hidden output states are concatenated to form the final embedding [20]. Thus, a character-level embedding is obtained for each word in a sentence, and consequently each sentence in the dataset. Finally, the word embedding of each sentence (or concatenation of word embeddings of each sentence with their corresponding BERT/DistilBERT embeddings) is passed into the CRF model for the sequence labeling task.

4.2.2 BERT/DistilBERT Embedding With CSE

We performed BERT/DistilBERT embedding with CSE for both types of architectures to obtain (BERT&CSE)/(DistilBERT&CSE) embeddings. In these embeddings, individual tokens of each sentence are passed to the model and CSE separately. Then, the output embedding of a particular token by the model is concatenated with that by CSE for the same token. Figure 4 illustrates the embedding process. This information is subsequently utilized in CRF for the opinion holder identification task.

4.2.3 Conditional Random Field (CRF)

CRFs are frequently applied in sequence labeling tasks, such as the named entity recognition (NER) tasks reported in [26]. A discrete classifier predicts a label for a single tag

without considering neighboring tags. However, a CRF accounts for context. For example, the linear chain CRF (commonly employed in NER) predicts sequences of labels for sequences of input samples [27, 28]. In the process, it jointly models the label decision by capturing dependencies across adjacent labels. Panchendrarajan et al. [29] and Watanabe et al. [30] have shown that the combination of CRF with different language models has recently been highly successful in the field of sequence labeling tasks. Accordingly, we combined the CRF model with BERT and DistilBERT for opinion holder identification. In our model architectures the CRF model receives inputs (i.e., emission scores/logits) that are generated by the language model's top linear layer. The architecture then predicts the likelihood of sequences of opinion holder tags. Figure 5 represents a diagram of the CRF layer.

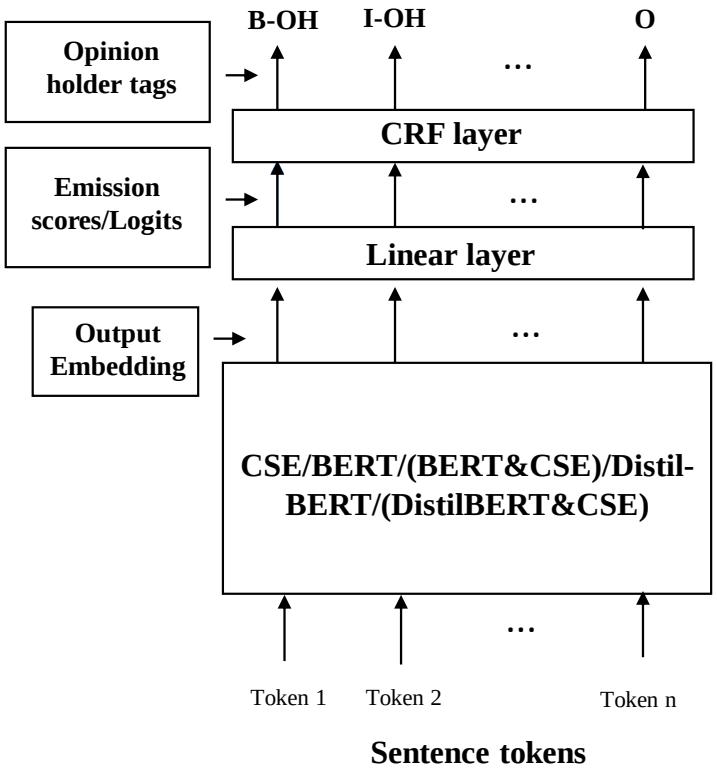

Figure 5: CRF layer on top of the language model.

5 Experiment and Analysis

5.1 INSIDE/OUTSIDE Classification

We employed six transformer-based pre-trained language models for the experiment: BERT-base-uncased, BERT-base-cased, BERT-large-uncased, BERT-large-cased, DistilBERT-base-uncased, and DistilBERT-base-cased. The "large" models contain more encoders, attention heads, and model parameters compared to the "base" models. The "uncased" models do not distinguish between lowercase and uppercase characters, whereas the

Table 1: Experimental Results Without Shuffling for the INSIDE/OUTSIDE Classification Task

Models	**F1 score**				**accuracy**			
	75:25	***80:20***	***80:10:10***	***10-fold***	***75:25***	***80:20***	***80:10:10***	***10-fold***
BERT-base-uncased	0.812	0.824	0.825	0.840	0.863	0.869	0.863	0.880
BERT-base-cased	0.847	0.844	0.796	0.869	0.885	0.883	0.842	0.902
BERT-large-uncased	0.792	0.810	0.778	0.820	0.849	0.863	0.836	0.867
BERT-large-cased	0.835	0.863	0.764	0.856	0.879	0.900	0.822	0.892
DistilBERT-base-uncased	0.852	0.869	0.860	**0.871**	0.890	**0.904**	0.890	**0.904**
DistilBERT-base-cased	0.814	0.829	0.743	0.835	0.868	0.880	0.815	0.878

Table 2: Experimental Results With Shuffling for the INSIDE/OUTSIDE Classification Task

Models	**F1 score**				**accuracy**			
	75:25	***80:20***	***80:10:10***	***10-fold***	***75:25***	***80:20***	***80:10:10***	***10-fold***
BERT-base-uncased	0.859	0.845	0.793	0.849	0.901	0.876	0.842	0.888
BERT-base-cased	0.845	0.865	0.864	0.863	0.890	0.893	0.884	0.896
BERT-large-uncased	0.838	0.821	0.846	0.826	0.885	0.859	0.890	0.871
BERT-large-cased	0.885	0.871	0.846	0.856	0.912	0.893	0.890	0.891
DistilBERT-base-uncased	0.824	**0.901**	0.837	0.871	0.887	**0.924**	0.890	0.901
DistilBERT-base-cased	0.818	0.870	0.796	0.837	0.868	0.904	0.870	0.881

"cased" models are case-sensitive. Therefore, all text in the "uncased" version is converted to lowercase prior to WordPiece tokenization.

5.1.1 Experimental Settings and Data Allocation

All experiments were conducted on a Linux server (CPU: Xeon E5-2620@2.10GHz 32proc, Mem: 256GB, GPU: Quadro RTX8000 (48GB)) and implemented in Python 3.10.

To avoid overfitting, it is important to allocate the data among training, validation, and testing sets. The ratio is empirically determined: 70-80% of the data is allocated for the training and the remaining 20-30% is used for the testing. Specifically, 75:25 is the default splitting ratio of the train_test_split() method in Python, whereas 80:20 and 80:10:10 are the most common ratios used for machine learning models for training/(validation)/testing. Accordingly, we employed all three ratios. We also evaluated our task with 10-fold cross-validation to prevents overfitting during training.

We conducted the experiments with and without shuffling prior to allocation. Data shuffling is performed prior to model training in order to create more representative training, validation, and testing sets. It is generally useful when data are ordered and sorted, and it reduces the bias of the training process. We also tested results without shuffling, as the [CLS] tensor feature data are unordered and unsorted.

We set maximum epochs to 100 and tolerance limit to 0.0001 as stopping criteria for training.

5.1.2 Experimental Result Analysis

Because the distribution of class labels in our prepared dataset is imbalanced, we employed both F1 score and accuracy as evaluation metrics.

Tables 1 and 2 list the experimental results without and with shuffling the data for the INSIDE/OUTSIDE classification task, respectively.

Our result analysis is considers four perspectives: (a) comparison between "base" and "large" models, (b) effect of data shuffling prior to allocation, (c) effect of cross-validation with and without data shuffling, and (d) comparison between the BERT and DistilBERT models.

As seen in Tables 1 and 2, most smaller models (such as DistilBERT-base-uncased and DistilBERT-base-cased) obtained a better F1 score and accuracy in most of the cases for the 80:20 splitting ratio. In this case, the best performance was obtained by DistilBERT-base-uncased. Conversely, in case of with shuffling, larger models such as BERT-base-uncased, BERT-base-cased, BERT-large-uncased, and BERT-large-cased performed well with other allocation ratios, as they were able to extract more useful features. For 75:25 splitting, BERT-large-cased obtained the best result, as it has more stack encoders, attention heads, and hidden layers than any other models used in our experiment. Consequently, it can extract more useful features for the classification task with relatively less training data.

In most cases, data shuffling before data allocation improved the F1 score and accuracy, as it maintains the model's generality and avoids the overfitting. Thus, it prevents the model from training bias.

Cross-validation is a method wherein different portions of data are used in different iterations. The purpose is to evaluate model performance in each iteration, thus avoiding overfitting and selection bias. We employed 10-fold cross-validation for both cases, as shown in Tables 1 and Table 2. Because 10-fold cross-validation employs a 90:10 allocation ratio, this approach yielded a maximal proportion of training data in the context of our study. Although F1 score and accuracy were improved in the case without shuffling, this was not always true in the case with shuffling.

In our experiment, DistilBERT-base-uncased produced superior performance compared to all other models, denoted by bold in Tables 1 and 2. The BERT-base and BERT-large models perform well with relatively larger training sets, as in the GLUE benchmark reported by Wang et al. [31]. However, as our dataset is relatively small, DistilBERT yielded the optimal performance in our study.

5.2 Opinion Holder Identification

5.2.1 Experimental Settings and Data Splittings

All experiments were conducted on the same Linux server, and implemented in the same programming language, as mentioned in subsection 5.1.1. We utilized the Flair framework [32] in this experiment.

We split the dataset into 80% training, 10% validation, and 10% testing. We set no. of epochs = 5, mini batch size = 32, learning rate = 0.1 (for the feature-based architecture) and 0.00005 (for the fine-tuning-based architecture), Optimizer = Stochastic Gradient Descent (for feature-based architectures) and AdamW (for fine-tuning-based architectures).

Because the class distribution of opinion holder tags is imbalanced, we employed the F1 score as an evaluation criterion. We also considered the total processing time of each model for comparison.

Table 3: Experimental Results for Opinion Holder Identification

Architectures	Models	Precision	Recall	F1 score	Processing time
Feature-based	CSE+CRF	0.9400	0.9495	0.9447	**49 sec.**
	BERT+CRF	0.9216	0.9495	0.9353	1 min. 45 sec.
	(BERT&CSE)+CRF	0.9307	0.9495	0.9400	2 min. 25 sec.
	DistilBERT+CRF	0.9400	0.9495	0.9447	1 min. 36 sec.
	(DistilBERT&CSE)+CRF	0.9314	0.9596	**0.9453**	2 min. 2 sec.
Fine-tuning-based	BERT	0.9400	0.9495	0.9447	3 min. 39 sec.
	BERT+CRF	0.9394	0.9394	0.9394	3 min. 37 sec.
	(BERT&CSE)+CRF	0.9400	0.9495	0.9447	4 min. 20 sec.
	DistilBERT	0.9400	0.9495	0.9447	2 min. 58 sec.
	DistilBERT+CRF	0.9307	0.9495	0.9400	3 min. 8 sec.
	(DistilBERT&CSE)+CRF	0.9314	0.9596	**0.9453**	3 min. 33 sec.

5.2.2 Experimental Result Analysis

Table 3 lists the experimental results for opinion holder identification. In our feature-based architecture, the network layers of CSE, BERT, and DistilBERT were kept static and re-training was not performed. Hence, the pre-trained parameters of these models were not changed during training. In contrast, in our fine-tuning-based architecture, gradients were calculated for BERT and DistilBERT, and the network layers were re-trained. Hence, the pre-trained parameters were updated during training. Note that in the concatenated model for the fine-tuned-based architecture, we fine-tuned only BERT for the (BERT&CSE)+CRF model, and DistilBERT for the (DistilBERT&CSE)+CRF model but not the CSE. Among the feature-based models, (DistilBERT&CSE)+CRF performed best in terms of F1 score (i.e., 0.9453). However, CSE+CRF performed best in terms of processing time (i.e., 49 s). Among the fine-tuning-based models, (DistilBERT&CSE)+CRF performed best in terms of F1 score (i.e., 0.9453), whereas DistilBERT performed best in terms of processing time (i.e., 2 min 58 s).

Within the same architecture, DistilBERT-based models demand less processing time than BERT-based models considering their respective non-concatenated and concatenated forms with CSE. The reason is DistilBERT is a lighter model, as it has less parameters than BERT.

In most cases, the concatenation of CSE slightly improved the F1 scores obtained by the original language models. However, more processing time was required.

It is apparent that both the feature- and fine-tuning-based (DistilBERT&CSE)+CRF models jointly performed best in terms of F1 score, which is 0.9453. However, the feature-based model required significantly less processing time than the fine-tuning-based model. This is generally the case when comparing the two architectures.

The feature-based CSE+CRF model performed best in terms of processing time (i.e., 49 s) while yielding a highly competitive F1 score of 0.9447.

6 Conclusion

We proposed deep learning-based pre-trained models for opinion holder detection in text. To detect the textual presence of opinion holder, we defined a binary classification task

between the INSIDE and OUTSIDE classes. To identify opinion holders, we performed a sequence labeling task. We constructed a new English dataset in the laptop domain from the ASTE-DATA-V2 dataset to perform the opinion holder detection task. For detecting the presence of opinion holder in the text, the DistilBERT-base-uncased model achieved superior performance among all tested models. For opinion holder identification, the feature- and fine-tuning-based (DistilBERT&CSE)+CRF models jointly obtained the optimal performance. However, the feature-based CSE+CRF model outperformed all other models in terms of processing time while yielding a comparable F1 score to those obtained by the best-performing models.

Because our dataset is relatively small, processing time was not a significant concern throughout the experiments. However, this is not the case with larger datasets. When processing time is a significant concern, feature-based models are more suitable than fine-tuning-based models, especially in the presence of CSE. However, fine-tuning-based models generally yield superior performance in terms of F1 score.

Aspect-based sentiment analysis with opinion holder information is the most crucial potential future direction of research, as it has not been considered by any existing studies. The informativeness of sentences in the INSIDE class may encompass a wide range, as mentioned in Section 1. Therefore, predicting the informativeness of each INSIDE sentence via regression models is another important direction of research. The introduction of a fine-grained classification task to the current classification may also be interesting. In addition, we only considered laptop domain data throughout this study. In future studies, we can use datasets from a wide variety of domains to extend the applicability of our methods.

Acknowledgments

This work was supported by JST SPRING, Grant Number JPMJSP2154.

References

[1] Al-Mahmud and Kazutaka Shimada. Dataset construction and classification based on pre-trained models for opinion holder detection. In *12th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 65–70, 2022.

[2] Lun-Wei Ku, Chia-Ying Lee, and Hsin-Hsi Chen. Identification of opinion holders. *Computational Linguistics and Chinese Language Processing*, 14(4):383–402, 2009.

[3] Yohei Seki, Noriko Kando, and Masaki Aono. Multilingual opinion holder identification using author and authority viewpoints. *Information Processing Management*, 45(2):189–199, 2009.

[4] https://github.com/xuuuluuu/SemEval-Triplet-data/tree/master/ASTE-Data-V2-EMNLP2020.

[5] Yu-Chieh Wu, Li-Wei Yang, Jeng-Yan Shen, Liang-Yu Chen, and Shih-Tung Wu. Tornado in multilingual opinion analysis: a transductive learning approach for chinese sentimental polarity recognition. In *Proceedings of the NTCIR-7 workshop*, pages 301–306, 2008.

[6] Soo-Min Kim and Eduard Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 1–8, 2006.

[7] Youngho Kim, Seongchan Kim, and Sung-Hyon Myaeng. Extracting topic-related opinions and their targets in NTCIR-7. In *Proceeedings of NTCIR-7 workshop*, pages 247–254, 2008.

[8] Youngho Kim, Yuchul Jung, and Sung-Hyon Myaeng. Identifying opinion holders in opinion text from online newspapers. In *2007 IEEE International Conference on Granular Computing (GRC 2007)*, pages 699–702, 2007.

[9] Ruifeng Xu, Kam-Fai Wong, and Yunqing Xia. Coarse-fine opinion mining-WIA in NTCIR-7 MOAT task. In *Proceedings of NTCIR-7 workshop*, pages 307–313, 2008.

[10] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 355–362, 2005.

[11] Eric Breck, Yejin Choi, and Claire Cardie. Identifying expressions of opinion in context. In *Proceedings of IJCAI*, volume 7, pages 2683–2688, 2007.

[12] Meng Xinfan and Wang Houfeng. Detecting opinionated sentences by extracting context information. In *Proceedings of the NTCIR-7 workshop*, pages 268–271, 2008.

[13] Kang Liu and Jun Zhao. NLPR at multilingual opinion analysis task in NTCIR-7. In *Proceedings of NTCIR-7 workshop*, pages 226–231, 2008.

[14] Bin Lu. Identifying opinion holders and targets with dependency parser in chinese news texts. In *Proceedings of the NAACL HLT 2010 student research workshop*, pages 46–51, 2010.

[15] Mohamed Elarnaoty, Samir AbdelRahman, and Aly Fahmy. A machine learning approach for opinion holder extraction in arabic language. *arXiv preprint arXiv:1206.1011*, 2012.

[16] Michael Wiegand, Marc Schulder, and Josef Ruppenhofer. Opinion holder and target extraction for verb-based opinion predicates–the problem is not solved. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 148–155, 2015.

[17] Michael Wiegand and Josef Ruppenhofer. Opinion holder and target extraction based on the induction of verbal categories. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 215–225, 2015.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[20] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

[21] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, 2014.

[22] 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 2015)*, pages 486–495, 2015.

[23] 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 Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud Maria Jiménez-Zafra, and Gülsen Eryigit. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, pages 19–30, 2016.

[24] Lu Xu, Hao Li, Wei Lu, and Lidong Bing. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349, 2020.

[25] Stefan Schweter and Alan Akbik. FLERT: document-level features for named entity recognition. *CoRR*, abs/2011.06993, 2020.

[26] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.

[27] https://medium.com/data-science-in-your-pocket/named-entity-recognition-ner-using-conditional-random-fields-in-nlp-3660df22e95c.

[28] https://hyperscience.com/tech_blog/exploring-conditional-random-fields-for-nlp-applications/.

[29] Rrubaa Panchendrarajan and Aravindh Amaresan. Bidirectional LSTM-CRF for named entity recognition. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, pages 531–540, 2018.

[30] Taiki Watanabe, Akihiro Tamura, Takashi Ninomiya, Takuya Makino, and Tomoya Iwakura. Multi-task learning for chemical named entity recognition with chemical compound paraphrasing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6244–6249, 2019.

[31] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[32] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.