Applying Contrastive Learning to an Attention Neural Model in a Multilingual Context

Philipp Gottschalk, Flavius Frasincar(✉)[0000−0002−8031−758X], and Eyo Herstad

Erasmus University Rotterdam, Burgemeester Oudlaan 50,
3062 PA Rotterdam, the Netherlands
philipp.lukas.gottschalk@gmail.com, {frasincar,herstad}@ese.eur.nl

Abstract. This research contributes to the field of Aspect-Based Sentiment Classification (ABSC) of Web data by proposing new cross-, multi- and unilingual ASBC models. We do this by improving the state-of-the-art mLCR-Rot-hop++ attention neural model and its variations. We introduce different multilingual XLM-R embedders to replace the multilingual BERT (mBERT) embedder found within the mLCR-Rot-hop++ model. Furthermore, we add two distinct contrastive learning methods to the existing mLCR-Rot-hop++ model. The first approach integrates sentiment-level contrastive learning, adapted to instances rather than individual token embeddings, into the mLCR-Rot-hop++ model. Our second approach considers the high-level opinion representations of the mLCR-Rot-hop++ model within the contrastive loss function. Our findings indicate that replacing the mBERT embedder with an XLM-R_{base} embedder generally improves performance. Furthermore, sentiment-level contrastive learning usually improves the performance of various models, especially compared to representation-level contrastive learning.

Keywords: ABSC · ML-ABSC · XL-ABSC · UL-ABSC · LCR-Rot-hop++

1 Introduction

With the ever-expanding presence of online marketplaces, review platforms, and social media on the Web, opinionated text has become ubiquitous. As a result, there is an escalating interest in being able to aggregate large swaths of opinionated text into useful data, leading to a surge of research in sentiment analysis.

Sentiment analysis concerns itself with the automated classification of opinionated text. In particular, there has been an expanded exploration of Aspect-Based sentiment analysis (ABSA), which focuses on detecting the sentiment of a certain entity, or aspect of an entity, within a given text [17]. ABSA can be subdivided into multiple tasks. Our investigation deals with the subtask of Aspect-Based Sentiment Classification (ABSC), which consists of assigning labels and sentiment scores to previously extracted aspects [2].

A significant challenge when classifying online opinionated texts is the diverse range of languages used. Consequently, there is a vested interest in creating

models which can be trained in resource-rich source languages and then applied to resource-poor target languages. This approach is often referred to as Cross-Lingual ABSA (XL-ABSA). As specified by [16], XL-ABSA trains models on a labelled source language and then applies these models to an unlabelled target language. Another possible approach is to train models on a range of different source languages, thereby creating a model which is relatively language-agnostic. This approach is usually referred to as Multilingual ABSA (ML-ABSA). Models trained in the same language later used to evaluate them are referred to as Unilingual ABSA (UL-ABSA) models.

In this paper, we contribute to the ABSC field by proposing new XL-ABSA and ML-ABSA models inspired by those introduced in [5], which use the state-of-the-art ABSC LCR-Rot-hop++ method [13] as a backbone. The improvements that we propose are two-fold. First, we replace the mBERT embedder that the model currently uses with two versions (base and large) of the XLM-RoBERTa (XLM-R) embedder, which have been shown to outperform mBERT on numerous cross-lingual benchmarks [3]. This gives us the $\text{XLMR}_{\text{base}}$-LCR-Rot-hop++ and XLMR-LCR-Rot-hop++ models.

Moreover, we integrate contrastive learning, proposed for XL-ABSA usage by [6], into the framework of the multilingual LCR-Rot-hop++ models. As mentioned in [6], contrastive learning works by shortening the distance between so-called anchor points and *positive* samples and increasing the distance between anchor points and *negative* samples. We utilise contrastive learning for the sentiment labels of entire instances, leading to the CLS-$\text{XLMR}_{\text{base}}$-LCR-Rot-hop++ model. This contrasts [6], which uses contrastive learning for the sentiment labels of tokens. Inspired by the method in [7], we also introduce a novel contrastive learning approach which uses the concatenated high-level representations from LCR-Rot-hop++ model for contrastive learning, giving us the CLR-$\text{XLMR}_{\text{base}}$-LCR-Rot-hop++ model. The paper's source code is found on Github at https://github.com/P-Gottschalk/CL-XLMR_base-LCR-Rot-hop-plus-plus.git.

The rest of the paper is constructed as follows. In Sect. 2, an overview of the current state of the research is provided. Section 3 describes the data utilised throughout this research. Section 4 then details the proposed methodology. The results of the investigation can be found in Sect. 5. In Sect. 6 we provide our research conclusion and suggestions for future work.

2 Related Work

[11], an ABSA survey pre-dating [17], states that research on ABSA falls into three categories: Aspect Detection (AD), Aspect-Based Sentiment Classification (ABSC), and joint AD and ABSC. A summary of AD is provided by [12]. Our investigation focuses on ABSC [2] and its application to a multilingual setting.

As noted by [2], ABSC can generally be categorised into three major categories: knowledge-based, machine learning, and hybrid models. Knowledge-based models classify sentiments using pre-determined rules, relations, and lexicalisations. Machine learning approaches are trained to extract sentiments using a

training dataset of feature vectors labelled with sentiments. Hybrid approaches combine knowledge bases and machine learning approaches, aiming to use knowledge bases where a lack of data hinders machine learning approaches.

In our research, we focus on extending a state-of-the-art machine learning approach. [18] introduces a Left-Center-Right separated neural network with Rotary attention (LCR-Rot). [14] utilises this model in a two-stage sentiment analysis algorithm called the Hybrid Approach ABSA (HAABSA) model. A lexicalised domain ontology is used to predict the sentiment, and LCR-Rot-hop, which runs multiple iterations of the rotary attention mechanism of LCR-Rot, is used as a backup model. HAABSA++ is subsequently introduced by [13]. This new model adds hierarchical attention to LCR-Rot-hop and replaces non-contextual word embeddings with deep contextual word embeddings, resulting in LCR-Rot-hop++. [5] then utilises the LCR-Rot-hop++ procedure for cross- and multilingual ABSC, replacing the previously used BERT embedder [4] by mBERT, a multilingual version of BERT trained using 104 languages [4].

We address the issue of low-resource languages by considering cross-lingual sentiment analysis [15] and multilingual sentiment analysis [1]. Both approaches aim to alleviate the issue of low-resource languages as a target language. Here, cross-lingual models are strictly trained on one source language and then applied to different target languages, whilst multilingual models are trained on multiple languages. It should be noted this distinction is often less pronounced in the literature, with multilingual sentiment analysis often serving as an umbrella term for both approaches. We further consider unilingual sentiment analysis for non-English languages, as in [5]. Since ABSA research is generally unilingual, this is rarely isolated as a distinct field of research in a multilingual context.

Numerous approaches to XL-ASBA and XL-ABSC are suggested in the literature. To augment available data, [16] creates an aspect code-switching mechanism, which switches aspect terms between instances in the source language and translated cases in the target language, using a combined dataset to train the model. [6] uses contrastive learning to achieve a convergence of semantic spaces across different languages. As described in Sect. 1, this is done by adjusting the distance between anchor points and corresponding *positive* and *negative* samples. Whilst applying contrastive learning to ABSA is increasing in popularity, to the best of our knowledge, [6] is one of the very few investigations to utilise contrastive learning cross-lingually. Unilingually, [7] presents a token-based approach to contrastive learning in ABSA. Instead of using probability distributions of the predicted sentiment within the contrastive loss function [6], [7] uses aspect-oriented sentiment representations. This gives a more fine-grained model, as the aspect-oriented sentiment representations contain more information than the sentiment probability distribution. [10] proposes a multi-layer network with divided attention to perform XL-ABSC. This method extracts Part-of-Speech (POS) information–grammatical properties such as nouns, adjectives, and verbs–and feeds this information to an attention-based convolutional neural network. [10] further leverages bilingual dictionaries to map converted tokens

across languages. As previously stated, [5] adapts the LCR-Rot-hop++ method to both a cross-lingual and a multilingual context.

Multilingual Masking Language Models (MLMs) are also being developed significantly. Improving on the widely utilised baseline model mBERT, [3] proposes XLM-R_{base} and XLM-R, two multilingual versions of Facebook's RoBERTa [8]. Whilst only trained on 100 languages [3], compared to the 104 languages in mBERT's training set, XLM-R_{base} and XLM-R have a larger vocabulary (250k tokens), compared to mBERT's (110k tokens).

3 Data

In this paper we use the SemEval-2016 dataset, developed by [9]. This dataset is widely employed in ABSA research, and is therefore an appropriate benchmark for model evaluation.

We use the Task 5, Subtask 1 (SB1) data of the SemEval-2016 dataset. SB1 is focused on sentence-level ABSA and the identification of opinion tuples from the following three types of information: Aspect Category (AC), Opinion Target Expression (OTE), and Sentiment Polarity (SP). The dataset covers multiple topics, including hotels, consumer electronics, and restaurants. For this investigation, we use the restaurant dataset, as this dataset spans the most languages– English, French, Spanish, Dutch, Turkish, and Russian–out of the available data. Hence, it is therefore most appropriate for investigations into cross- and multilingual sentiment classification.

Firstly, note that Russian uses the Cyrillic alphabet and is consequently ill-suited for investigations into cross- and multi-lingual investigations, as similarities with the other languages are relatively low, so Russian is removed. We also drop Turkish from our dataset due to its comparatively small test sample. As seen in [9], the test set is limited to 39 sentences with 144 sentiments. Comparatively, the second smallest dataset, English test data, has 90 sentences and 676 expressed sentiments, an almost fivefold increase in sentiments compared to Turkish.

Further, we clean the data according to the methodology set out by [5]. Specifically, we remove any sentiment labels that are related to hidden aspects, as the used LCR-Rot-hop++ method introduced by [13], which serves as the foundation for this research, is not equipped to deal with implicit aspects.

The data files provided are in XML format. An example of a sentence from the dataset is provided in Fig. 1. Here, we see a specific sentence, with attached opinions, is provided. Each includes the target phrase of the opinion, the category of the target phrase, and an attached sentiment polarity.

```
<sentence id="1349391:0">
  <text>sometimes i get good food and ok service.</text>
  <Opinions>
    <Opinion target="food" category="FOOD#QUALITY" polarity="positive" from=
    "21" to="25"/>
    <Opinion target="service" category="SERVICE#GENERAL" polarity="neutral"
    from="33" to="40"/>
  </Opinions>
</sentence>
```

Fig. 1. SemEval-2016 example sentence.

Summary statistics for the dataset are provided in Table 1, including the frequency of sentiment polarities and their percentage. The data cleaning results in a loss of up to 35.7% for individual datasets.

Table 1. Summary statistics for our used data. The parentheses indicate the number of removed polarity labels.

	English		**French**		**Spanish**		**Dutch**	
	Train	Test	Train	Test	Train	Test	Train	Test
# Total	1880 (627)	650 (209)	1770 (706)	718 (236)	1937* (783)	731 (341)	1283 (577)	394 (219)
% Positive	70.2	74.3	50.9	50.7	70.6	71.3	59.1	62.2
% Neutral	3.83	20.8	42.5	39.7	24.7	24.1	31.7	31.7
% Negative	26.0	4.92	6.55	9.61	4.60	4.65	9.20	6.09

*A further opinion was removed during embedding due to an unknown polarity label.

4 Methodology

4.1 XLMR$_{\text{base}}$-LCR-Rot-hop++

The XLMR$_{\text{base}}$-LCR-Rot-hop++ model is based on the previously proposed mLCR-Rot-hop++ model introduced by [5]. This model serves as a basis for the remainder of the investigation, consistently achieving strong results when tested against state-of-the-art machine learning ABSC models. The model is trained on English data, embedded using a multilingual embedder. The test set used for the model is the test set for each language.

We use the two pre-trained multilingual configurations first introduced in [3]: XLM-R$_{\text{base}}$ and XLM-R, which sets us apart from the works of [5] and of [13], which use mBERT and BERT for the respective embeddings. The XLM-R embedder is the "larger" of the two models, containing approximately twice the number of parameters, double the number of layers, and 30% hidden states than XLM-R$_{\text{base}}$[1]. While XLM-R outperforms the XLM-R$_{\text{base}}$ in [3], it is worth testing our model with both embedders, as XLM-R$_{\text{base}}$ has similar model parameters

[1] Model sizes, written as {L, H, A, # param} [3]: mBERT = {12, 768, 12, 172M}; XLM-R$_{\text{base}}$ = {12, 768, 12, 270M}; XLM-R = {24, 1024, 16, 550M}.

to mBERT. Thereby, it could be more suited to the already existent LCR-Rot-hop++ framework, as the larger size of XLM-R embeddings may increase the quantity of data needed to train our model.

The embeddings are subsequently fed into the LCR-Rot-hop++ framework. This model is a neural network with a rotary attention mechanism, which operates at the sentence level. Each instance fed into the model is split into a left context, a target phrase, and a right context. The two-step rotary attention mechanism is then applied: first, the context representations for the left and right contexts are computed using a target representation from a pooling layer. We then introduce hierarchical attention to the model by tuning the context representations with respect to each other. Secondly, we compute target aspect representations using the right and left context representations found previously. Finally, the target representations are tuned using hierarchical attention. This step can be repeated n (here, $n = 3$) times, where the pooling layer target representations are replaced by the computed target representations after these are first computed. Once this attention mechanism has been repeated a sufficient number of times, the four context representations are concatenated to form a representation vector v_i for instance i, and the sentiment polarity is computed using a softmax function. Prediction p_i is evaluated using the cross-entropy loss function:

$$\underset{1\times1}{L_{CE}} = -\sum_{i=1}^{K} \underset{|C|\times1}{y_i} \times \log \underset{|C|\times1}{(p_i)} + \lambda||\Theta||^2 \tag{1}$$

where K denotes the size of the batch of training opinions, y_i the sentiment vector of x_i, p_i the prediction vector for instance x_i, $|C|$ the number of different sentiment categories, and λ the L_2 regularisation term for the parameter set Θ. We initialise the weights and bias terms using a uniform distribution and update using stochastic gradient descent with a momentum term. We tune hyperparameters using a Tree-structured Parzan Estimators (TPE) algorithm.

4.2 Variations on XLMR$_{\text{base}}$-LCR-Rot-hop++

The base mLCR-Rot-hop++ is trained on the English dataset from SemEval-2016. We will use this configuration as a baseline comparison for the XLMR$_{\text{base}}$-LCR-Rot-hop++ and XLMR-LCR-Rot-hop++ models, as it is also trained on the English datasets and outperforms most XL-ABSC and ML-ABSC models when tested on the French and Spanish datasets [5]. Moreover, we present further adaptations of the XLMR$_{\text{base}}$-LCR-Rot-hop++ model, which are used as comparisons to other well-performing models described in [5]. Table 2 shows the classifications of the variations to the standard XLMR$_{\text{base}}$-LCR-Rot-hop++ which which we propose in this paper.

Table 2. The classification of the models that are proposed.

	Model Type		
	ML-ABSC	XL-ABSC	UL-ABSC
$XLMR_{base}$-LCR-Rot-hop++	-	x	-
$XLMR_{base}$-LCR-Rot-hop-XX++	-	-	x
$XLMR_{base}$-MLCR-Rot-hop++	x	-	-
$XLMR_{base}$-LCR-Rot-hop-ACS_{XX}++	-	x	-

4.2.1 $XLMR_{base}$-LCR-Rot-hop-XX++. $XLMR_{base}$-LCR-Rot-hop-XX++ is a UL-ABSC model. The model is very similar to that of $XLMR_{base}$-LCR-Rot-hop++, with the key distinction being that rather than English, the model is trained on language XX, where XX ∈ {FR, ES, NL}. We use this model due to the strong performance of the comparable mLCR-Rot-hop-XX++ in [5].

4.2.2 $XLMR_{base}$-MLCR-Rot-hop++. $XLMR_{base}$-MLCR-Rot-hop++ is an ML-ABSA model. For this model, we create one large dataset containing the instances in the training data from all the available languages–English, French, Spanish, and Dutch–and concatenate them into a single dataset. We then train the $XLMR_{base}$-LCR-Rot-hop++ model on this large dataset and test the model's performance separately for each language.

4.2.3 $XLMR_{base}$-LCR-Rot-hop-ACS_{XX}++. This model utilises the ACS methodology introduced in [16]. The methodology inflates the size of a dataset both by translation and through code-switched bilingual sentences, thereby increasing the size of the dataset by an approximate factor of four.

We start with a single instance in English. We then translate this instance into a target language. To do so, we utilise Alignment-free Label Projection [16], which aims to obtain pseudo-labeled data in the target language. This involves marking the aspect terms with special symbols before translating the instance. Similar to [5], we utilise Google API, which supports over 130 languages.

After the translation, we extract the aspects from the translated instance, utilising the previously mentioned markings. Note that multiple aspects are not an issue, as different special symbols are used for each subsequent marking. We then assign the sentiment labels to the translated aspect, giving us bilingual data from a singular instance. When running our models, we remove instances with an empty target following the embedding process.

We subsequently focus on specific Aspect-term Code Switching [16]. This involves taking the two instances obtained from the above steps, one in English and one in the target language, and switching the aspects between the two instances, leaving us with four different instances. These datasets are combined to form a single large dataset, on which we then train our model. Fig. 2 shows the structure of the $XLMR_{base}$-LCR-Rot-hop-ACS_{XX}++ model.

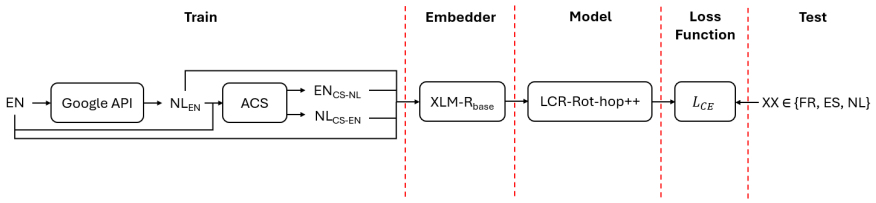

Fig. 2. A diagram of the $\text{XLMR}_{\text{base}}$-LCR-Rot-hop-$\text{ACS}_{\text{XX}}$++ model.

4.3 CLS-$\text{XLMR}_{\text{base}}$-LCR-Rot-hop++

Contrastive Learning Sentiment $\text{XLMR}_{\text{base}}$-LCR-Rot-hop++ (CLS-$\text{XLMR}_{\text{base}}$-LCR-Rot-hop++) fuses the previously defined XLMR-LCR-Rot-hop++ model and its variations with a contrastive learning approach, adapted from [6]. The same steps as in the $\text{XLMR}_{\text{base}}$-LCR-Rot-hop++ model–or for one of its variations– are carried out to obtain a sentiment prediction vector p_i. We combine the cross-entropy function from Equation 1 with a contrastive loss function and take a weighted sum of the two loss functions to evaluate the model.

Our approach is distinct from the work [6], which compares the sentiment labels of tokens, as we instead focus on comparing the sentiment labels of an entire instance. Let us denote our group of sample instances and the matching labels found in a given batch by $\{x_i, y_i\}_{i \in I}$, where set $I = \{1, ..., K\}$ represents the indices of a batch of size K. For all CLS models, we set $K = 32$. We define the positive set of all indices of the instances with the same label as the instance with index i, such that $P_i = \{j : j \in I, y_i = y_j \wedge i \neq j\}$. Here, $y_i \in Y_{\text{sen}}$, with Y_{sen} denoting the set of possible sentiments such that $Y_{sen} = \{\text{POS}, \text{NEU}, \text{NEG}\}$.

We then define the contrastive loss function for every $i \in I$ such that

$$L_{CLS_i} = -\sum_{j \in P_i} \log \frac{\exp(\text{sim}(p_i, p_j)/\tau)}{\sum_{k \in I/i} \exp(\text{sim}(p_i, p_k)/\tau)} \tag{2}$$

where τ is the temperature hyperparameter–set to 0.07, as in [6]–and $\text{sim}(\cdot)$ is the cosine similarity function. The contrastive loss function for the entire batch of size K can then be written as follows:

$$L_{CLS} = \sum_{i=1}^{K} \frac{1}{|P_i|} L_{CLS_i} \tag{3}$$

We combine Equations 1 and 3 to obtain our final model loss function:

$$L = (1 - \beta) \cdot L_{CE} + \beta \cdot L_{CLS} \tag{4}$$

where β is a hyperparameter used to weight the cross-entropy and sentiment-level contrastive loss functions.

4.4 CLR-XLMR$_{\text{base}}$-LCR-Rot-hop++

As mentioned in Sect. 2, another contrastive learning model is described in [7], using the aspect oriented sentiment representations. This approach is not directly applicable to the LCR-Rot-hop++ model structure, which outputs a concatenated representation vector v to feed into the final MLP layer. To deal with this incompatibility, we propose a contrastive learning methodology which directly utilises the high-level sentiment representation vector v in its contrastive learning function, to decrease the space between representation vectors with the same label. This model exploits the increased information found in the high-level opinion vectors, which may lead to more fine-grained contrastive learning. We call this model Contrastive Learning Representation XLMR$_{\text{base}}$-LCR-Rot-hop++ (CLR-XLMR$_{\text{base}}$-LCR-Rot-hop++) model.

As in Sect. 4.1, v_i is the concatenated representation vector of the instance x_i in a given batch of size K. Note, however, that due to the significant increase in the input size of the vectors utilised in contrastive learning, we decrease the batch size to $K = 10$ for the sake of computational feasibility, as larger batch sizes failed to run on the T4 GPU used for this project. We then define the individual loss function

$$L_{CLR_i} = -\sum_{p \in P_i} \log \frac{\exp(\text{sim}(v_i, v_p)/\tau)}{\sum_{k \in I/i} \exp(\text{sim}(v_i, v_k/\tau)} \tag{5}$$

with $\tau = 0.07$. The batch loss function and the final loss function are the same as in Equations 3 and 4, with L_{CLS_i} being replaced by L_{CLR_i}.

5 Results

For each model described in Sect. 4, we run the model using both the XLM-R$_{\text{base}}$ and the XLM-R embedder. We compare these to results we obtain from running the models first proposed by [5]. We subsequently also show and explain the results of each model using our two proposed contrastive learning approaches. The models are evaluated using two performance measures: the accuracy score and the (macro-)F_1 score. Whilst the accuracy score is the more interpretable evaluation measure, the macro-F_1 helps account for potentially unbalanced datasets, for example by punishing majority classification.

5.1 XLMR$_{\text{base}}$-LCR-Rot-hop++

Table 3. Results for XLMR$_{\text{base}}$-LCR-Rot-hop++ and comparable models.

	English		**French**		**Spanish**		**Dutch**	
	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1
mLCR-Rot-hop++	80.6	**0.593**	62.5	0.445	73.6	0.425	68.8	0.445
XLMR$_{\text{base}}$-LCR-Rot-hop++	**82.0**	0.504	**63.4**	**0.469**	**76.6**	**0.462**	**71.1**	**0.476**
XLMR-LCR-Rot-hop++	74.3	0.284	50.7	0.224	71.3	0.277	62.2	0.256

From Table 3, we notice that, across all languages, the XLMR$_{\text{base}}$-LCR-Rot-hop++ tends to outperform the other models, both when evaluated on the accuracy and the F_1 scores. The improvement in accuracy is consistent but small in magnitude. It is striking that the accuracy scores found for the XLMR-LCR-Rot-hop++ model are the same as the percentage of positive instances found in the test data, as shown in Table 1. Coupled with F_1 scores that are significantly lower than in models with similar accuracy, this strongly indicates that XLMR-LCR-Rot-hop++ cannot distinguish between different sentiments and reverts to classifying all instances as positive opinions. This reduced performance may be due to the model struggling to classify the larger XLM-R embedding vectors. The higher performance of the mLCR-Rot-hop++ and XLMR$_{\text{base}}$-LCR-Rot-hop++ on English language data likely stems from the fact that these models are trained on English data, whilst all other models are XL-ASBC.

We only apply contrastive learning to mLCR-Rot-hop++ and XLMR$_{\text{base}}$-LCR-Rot-hop++, as these outperform XLMR-LCR-Rot-hop++.

Table 4. Results for the models with contrastive learning.

	English		**French**		**Spanish**		**Dutch**	
	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1
CLS-mLCR-Rot-hop++	81.9	**0.660**	**67.4**	**0.471**	75.8	0.445	71.6	0.476
CLS-XLMR$_{\text{base}}$-LCR-Rot-hop++	**84.5**	0.534	62.8	0.453	76.6	**0.473**	**73.1**	**0.497**
CLR-mLCR-Rot-hop++	78.0	0.454	62.5	0.445	**76.9**	0.464	70.1	0.454
CLR-XLMR$_{\text{base}}$-LCR-Rot-hop++	83.2	0.521	61.0	0.453	75.7	0.461	67.5	0.449

From Table 4, we observe that sentiment-level contrastive learning outperforms representation-level contrastive learning in terms of accuracy and F_1 score. Adding sentiment-level contrastive learning to a model almost always increases the performance across both measures. In contrast, for the CLR-mLCR-Rot-hop++ model, the performance improves over the mLCR-Rot-hop++ model when classifying Dutch and Spanish data, stays constant for French data, and decreases for English data. Furthermore, CLR-XLMR$_{\text{base}}$-LCR-Rot-hop++ only improves relative to XLMR$_{\text{base}}$-LCR-Rot-hop++ when tested on the English data, but underperforms for all other languages. This lacklustre performance may be explained by the representation-level vectors containing too much information to effectively decrease the semantic space, instead increasing the noise in the model.

5.2 XLMR$_{\text{base}}$-LCR-Rot-hop-XX++

Table 5 presents the results of the unilingual models. Note that Table 5 also includes the results of the XLMR$_{\text{base}}$-LCR-Rot-hop++ model for English, as the XLMR$_{\text{base}}$-LCR-Rot-hop++ and XLMR$_{\text{base}}$-LCR-Rot-hop-XX++ are identical when trained and tested on an English dataset.

Table 5. Results for $\text{XLMR}_{\text{base}}$-LCR-Rot-hop-XX++ and comparable models.

	English		**French**		**Spanish**		**Dutch**	
	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1
mLCR-Rot-hop-XX++	80.6	**0.593**	**69.8**	**0.562**	**80.6**	**0.519**	71.1	0.467
$\text{XLMR}_{\text{base}}$-LCR-Rot-hop-XX++	**82.0**	0.504	66.2	0.486	71.8	0.427	**83.3**	**0.566**
XLMR-LCR-Rot-hop-XX++	74.3	0.284	69.5	0.509	71.3	0.277	62.2	0.256

In contrast to the results found in Table 3, Table 5 gives no obvious indication of a best-performing model. We notice that mLCR-Rot-hop++ performs strongly when applied to French and Spanish data, with this boosted performance being particularly noticeable in the Spanish model's accuracy measures. However, this performance does not carry over to the Dutch dataset, where $\text{XLMR}_{\text{base}}$-LCR-Rot-hop-XX++ is the best model by a substantial margin. The XLMR-LCR-Rot-hop-XX++ model again seems to classify by majority vote, with an exception for the French model. A possible explanation for this is that the sentiment polarities are more evenly distributed in the French data, with 50.9% of observations being positive. This indicates the model may perform better for the other languages if their training data were more evenly distributed.

Table 6. Results for the models with contrastive learning.

	English		**French**		**Spanish**		**Dutch**	
	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1
CLS-mLCR-Rot-hop-XX++	81.9	**0.660**	66.4	**0.569**	73.9	0.437	68.5	0.540
CLS-$\text{XLMR}_{\text{base}}$-LCR-Rot-hop-XX++	**84.5**	0.534	**68.9**	0.506	**85.1**	**0.554**	**81.2**	**0.708**
CLR-mLCR-Rot-hop-XX++	78.0	0.454	67.3	**0.569**	80.9	0.508	74.6	0.543
CLR-$\text{XLMR}_{\text{base}}$-LCR-Rot-hop-XX++	83.2	0.521	50.7	0.224	71.3	0.277	62.2	0.256

Table 6 shows that the CLS-$\text{XLMR}_{\text{base}}$-LCR-Rot-hop-XX++ model performs best across the board, delivering better results than its counterpart without contrastive learning. The performances of the two contrastive models utilising mBERT embedders are more mixed. Whilst the CLS-mLCR-Rot-hop++ model mostly has better F_1 scores than mLCR-Rot-hop++, its accuracy is relatively worse. From our results, it is unclear whether the CLR-mLCR-Rot-hop++ model improves over mLCR-Rot-hop++. The results in Dutch, French and Spanish indicate that the CLR-$\text{XLMR}_{\text{base}}$-LCR-Rot-hop-XX++ model scores poorly, again seeming to revert to majority classification. However, unbalanced data may not cause this phenomenon, as CLR-$\text{XLMR}_{\text{base}}$-LCR-Rot-hop-XX++ outperforms $\text{XLMR}_{\text{base}}$-LCR-Rot-hop-XX++ in English, despite 70.2% of the English training data being positive instances.

5.3 XLMR$_{\text{base}}$-MLCR-Rot-hop++

Table 7. Results for XLMR$_{\text{base}}$-MLCR-Rot-hop++ and comparable models.

	English		French		Spanish		Dutch	
	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1
MLCR-Rot-hop++	69.4	**0.523**	61.4	0.513	76.5	**0.556**	65.5	**0.524**
XLMR$_{\text{base}}$-MLCR-Rot-hop++	**80.5**	0.497	**71.5**	**0.519**	**78.8**	0.501	**76.9**	0.523
XLMR-MLCR-Rot-hop++	79.1	0.472	69.6	0.505	75.8	0.456	70.1	0.471

Table 7 shows that the models using XLM-R$_{\text{base}}$ and XLM-R embedders are consistently more accurate than MLCR-Rot-hop++. This is especially clear in the performance of the XLMR$_{\text{base}}$-MLCR-Rot-hop++. However, evaluating by the F_1 measure tells a starkly different story. Here, the MLCR-Rot-hop++ largely outperforms the models using XLM-R type embedders. The accuracy differences between models seem to be more substantive than the F_1 differences. This is most clearly seen in the Dutch results, where XLMR$_{\text{base}}$-MLCR-Rot-hop++ increases accuracy over MLCR-Rot-hop++ by 11.4 percentage points, whilst the F_1 score of MLCR-Rot-hop++ is only 0.001 higher. The comparitively higher accuracy when testing on English and Spanish data may be due to the higher fraction of English and Spanish data in the multilingual training set.

Table 8. Results for the models with contrastive learning.

	English		French		Spanish		Dutch	
	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1
CLS-MLCR-Rot-hop++	**80.0**	**0.583**	73.8	0.621	82.1	0.618	78.2	**0.604**
CLS-XLMR$_{\text{base}}$-MLCR-Rot-hop++	74.9	0.507	**77.6**	**0.692**	**84.5**	**0.718**	**83.5**	0.575
CLR-MLCR-Rot-hop++	75.1	0.534	71.0	0.592	78.7	0.544	74.4	0.596
CLR-XLMR$_{\text{base}}$-MLCR-Rot-hop++	74.5	0.426	51.3	0.404	71.3	0.277	62.2	0.256

In Table 8, we notice that sentiment-level contrastive learning is extremely effective, regardless of the utilised embedder. The CLS-MLCR-Rot-hop++ outperforms the corresponding MLCR-Rot-hop++ across all languages in both measures. Similarly, the CLS-XLMR$_{\text{base}}$-MLCR-Rot-hop++ consistently scores higher than the original XLMR$_{\text{base}}$-MLCR-Rot-hop++ model. Representation-level contrastive learning fails to perform to the same degree. The CLR-MLCR-Rot-hop++ outperforms the MLCR-Rot-hop++ model for most measures, albeit to a lesser degree than the CLS-MLCR-Rot-hop++ model. In contrast, CLR-XLMR$_{\text{base}}$-MLCR-Rot-hop++ delivers a subpar performance, with the accuracy and F_1 measures suggest that the model reverts to majority classification when classifying Spanish and Dutch data.

5.4 XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++

For ACS cross-lingual models, we do not include the English test data for our ACS methodology, as English is used to generate our translations.

Table 9. Results for XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++ and comparable models.

	French		**Spanish**		**Dutch**	
	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1
mLCR-Rot-hop-ACS$_{\text{XX}}$++	66.4	0.526	71.0	0.506	61.4	**0.581**
XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++	**78.7**	**0.550**	**83.5**	**0.535**	68.8	0.467
XLMR-LCR-Rot-hop-ACS$_{\text{XX}}$++	75.2	0.525	81.0	0.522	**73.9**	0.492

From Table 9 we find that, generally, the XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++ has the highest accuracy and F_1 scores. The only exception is in the Dutch language test data, where the mLCR-Rot-hop++ model has a substantially higher F_1 and the XLMR-LCR-Rot-hop-ACS$_{\text{XX}}$++ model has the highest accuracy. Note that the difference in evaluation measures is relatively lower between the two XLM-R embedding models compared to the other models. This could be attributable to the "artificial" size increase induced by the ACS methodology, which approximately quadruples the size of our dataset. Given that the XLM-R embedder provides larger embeddings than XLM-R$_{\text{base}}$ and mBERT, we hypothesize that the subsequent LCR-Rot-hop++ model requires more training data to provide a similar–or potentially superior–level of accuracy.

Table 10. Results for the models with contrastive learning.

	French		**Spanish**		**Dutch**	
	Acc. (%)	F_1	Acc. (%)	F_1	Acc. (%)	F_1
CLS-mLCR-Rot-hop-ACS$_{\text{XX}}$++	65.7	0.522	76.5	0.513	69.8	0.585
CLS-XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++	**74.9**	**0.664**	74.8	**0.538**	**86.3**	**0.586**
CLR-mLCR-Rot-hop-ACS$_{\text{XX}}$++	65.9	0.550	75.7	0.518	66.8	0.425
CLR-XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++	70.2	0.562	**80.4**	0.516	68.8	0.464

The results in Table 10 indicate that sentiment-level contrastive learning tends to improve a model's performance. The CLS-mLCR-Rot-hop-ACS$_{\text{XX}}$++ improves over the mLCR-Rot-hop-ACS$_{\text{XX}}$++ across all languages except for French, whilst the CLS-XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++ has a consistently better F_1 score than the XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++. Representation-level contrastive learning performs less well, although some improvements over the mLCR-Rot-hop-ACS$_{\text{XX}}$++ model are noticeable, for example in Spanish. The CLR-XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++ model is not a substantial improvement over the XLMR$_{\text{base}}$-LCR-Rot-hop-ACS$_{\text{XX}}$++ model.

6 Conclusion

We contribute to the literature by proposing several extensions to the existing mLCR-Rot-hop++ model. The original mBERT embedder is exchanged for an XLM-R_{base} or an XLM-R embedder. Furthermore, we integrate sentiment-level and representation-level contrastive learning into our proposed models.

Replacing the mBERT embedder with an XLM-R_{base} improves the results for the majority of the proposed models. The XLM-R embedder achieves relatively poor results, only outperforming the other embedders in a single measure across all four proposed model variations. This lacklustre performance by the XLM-R embedder may be caused by the size of the training dataset. As previously mentioned, XLM-R embeddings are larger than mBERT and XLM-R_{base} embeddings. Hence, it can be reasoned that the LCR-Rot-hop++ method could require a larger training dataset when fitted on XLM-R embeddings. Indeed, the performance difference between the XLM-R and XLM-R_{base} embedder decreases for models trained on comparatively larger datasets.

Across all languages, we notice that models which combine sentiment-level contrastive learning with mBERT embedders and XLM-R_{base} embedders tend to outperform the respective base models. In contrast, representation-level contrastive learning only occasionally improves model performance and usually performs worse than sentiment-level contrastive learning. Note here that high-level opinion representations contain significantly more information than the sentiment probability vectors, presumably making it considerably harder to decrease the sentiment space between these vectors, as the likelihood of these vectors displaying any similarity is significantly lower. Additionally, the computational limits to the batch size may also present a hurdle, as larger batches allow more comparisons to occur. Hence, sentiment-level contrastive learning is currently the most compatible with the LCR-Rot-hop++ method in a multilingual context.

A prospective research direction is the introduction of POS tagging into the LCR-Rot-hop++ structure. As explained in [10], POS denotes the grammatical properties of words in an instance, which likely have a strong connection with aspect-based sentiments and could augment model performance.

References

1. Agüero-Torales, M.M., Salas, J.I.A., López-Herrera, A.G.: Deep learning and multilingual sentiment analysis on social media data: An overview. Applied Soft Computing **107**, 107373 (2021)
2. Brauwers, G., Frasincar, F.: A survey on aspect-based sentiment classification. ACM Computing Surveys **55**(4), 65:1–65:37 (2023)
3. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020). pp. 8440–8451. ACL (2020)
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: 2019 Conference of the North

American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019). pp. 4171–4186. ACL (2019)
5. Horst, S., Frasincar, F.: Multilingual, cross-lingual, and unilingual models for ABSC. In: 25th International Conference on Web Information Systems Engineering (WISE 2024). LNCS, vol. 15436, pp. 89–101. Springer (2024)
6. Lin, N., Fu, Y., Lin, X., Zhou, D., Yang, A., Jiang, S.: CL-XABSA: Contrastive learning for cross-lingual aspect-based sentiment analysis. IEEE/ACM Transactions on Audio, Speech, and Language Processing **31**, 2935–2946 (2023)
7. Lingling, X., Weiming, W.: Improving aspect-based sentiment analysis with contrastive learning. Natural Language Processing Journal **3**, 100009 (2023)
8. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
9. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., Clercq, O.D., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N.V., Kotelnikov, E.V., Bel, N., Zafra, S.M.J., Eryigit, G.: Semeval-2016 task 5: Aspect based sentiment analysis. In: 10th International Workshop on Semantic Evaluation (SemEval 2016). pp. 19–30. ACL (2016)
10. Sattar, K., Umer, Q., Vasbieva, D.G., Chung, S., Latif, Z., Lee, C.: A multi-layer network for aspect-based cross-lingual sentiment classification. IEEE Access **9**, 133961–133973 (2021)
11. Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering **28**(3), 813–830 (2016)
12. Truşcă, M.M., Frasincar, F.: Survey on aspect detection for aspect-based sentiment analysis. Artificial Intelligence Review **56**(5), 3797–3846 (2023)
13. Truşcă, M.M., Wassenberg, D., Frasincar, F., Dekker, R.: A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention. In: 20th Conference on Web Engineering (ICWE 2020). LNCS, vol. 12128 of LNCS, pp. 365–380. Springer (2020)
14. Wallaart, O., Frasincar, F.: A hybrid approach for aspect-based sentiment analysis using a lexicalized domain ontology and attentional neural models. In: 16th Extended Semantic Web Conference (ESWC 2019). LNCS, vol. 11503 of LNCS, pp. 363–378. Springer (2019)
15. Xu, Y., Cao, H., Du, W., Wang, W.: A survey of cross-lingual sentiment analysis: Methodologies, models and evaluations. Data Science and Engineering **7**(3), 279–299 (2022)
16. Zhang, W., He, R., Peng, H., Bing, L., Lam, W.: Cross-lingual aspect-based sentiment analysis with aspect term code-switching. In: 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021). pp. 9220–9230. ACL (2021)
17. Zhang, W., Li, X., Deng, Y., Bing, L., Lam, W.: A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. IEEE Transactions on Knowledge and Data Engineering **35**(11), 11019–11038 (2023)
18. Zheng, S., Xia, R.: Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention. arXiv preprint arXiv:1802.00892 (2018)