Multilingual, Cross-lingual, and Unilingual Models for ABSC

Steinar Horst and Flavius Frasincar(✉) [0000−0002−8031−758X]

Erasmus University Rotterdam, Burgemeester Oudlaan 50, 3062 PA Rotterdam, the Netherlands
steinar.horst@gmail.com, frasincar@ese.eur.nl

Abstract. With opinionated text becoming an overabundant resource due to the Web, the need for processing this resource is becoming more and more relevant. That is why the field of Aspect-Based Sentiment Classification (ABSC) has seen rapid development. However, most research has been focused on English texts. Therefore, this paper adds to the field of Multilingual Aspect-Based Sentiment Classification (MABSC) and Cross-lingual Aspect-Based Sentiment Classification (XABSC). We take a state-of-the-art ABSC model, LCR-Rot-hop++, and repurpose it for MABSC and XABSC. We propose MABSC models mLCR-Rot-hop++ and MLCR-Rot-hop++, which use mBERT and a multilingual dataset, respectively. For XABSC we propose MLCR-Rot-hop-XX_{en}, which uses translation techniques from English to another language (XX) to form the training data. Furthermore, we make use of Aspect-Code-Switching (ACS) to further extend the training data and make it bilingual. Last, we also introduce Unilingual ABSC (UABSC) models, which are models trained on resource-poor languages. These models are called mLCR-Rot-hop-XX++. The best performance for MABSC is shown by MLCR-Rot-hop++. Furthermore, mLCR-Rot-hop++ is our best model for XABSC. The UABSC models mLCR-Rot-hop-XX++ are the best performing overall but these are trained and tested on resource-poor languages.

Keywords: ABSC · MABSC · XABSC · UABSC

1 Introduction

Nowadays, text has become an overabundant resource. Moreover, since the rise of the Web, people have been able to easily voice their opinions on any matter. For example, user reviews can now be easily made for films, restaurants, or any location on online map services. Therefore, it has become more and more important to be able to analyze these large bodies of text in a rapid manner. One way to analyze texts is by evaluating the sentiment of a text towards a certain entity, like a review of the food in a restaurant. The field that has arisen to perform this task, is known as Sentiment Analysis [4]. More specifically, sentiment analysis has been done using aspects, which can be any characteristic or property of the entity in question [6]. Hence, this subfield is called Aspect-Based Sentiment Analysis (ABSA).

ABSA has two main components, Aspect Detection (AD) and Aspect-Based Sentiment Classification (ABSC) [6]. This research focuses on ABSC [1], as our aim is to be able to correctly determine a text's sentiment without being mislead by a language's intricacies. Furthermore, as most research has been done on only English texts [5], we want to develop a model that gives good results for ABSC of texts in different languages.

Another concern for ABSC lies in the fact that not all languages have an equal amount of resources. So-called resource-poor languages lack prepped datasets with annotated text, needed for ABSC. This increases the need to develop ABSC for this type of languages.

We focus on three main ways to address the previous concerns. The first is the MABSC task. An MABSC task needs a model to make sentiment predictions for texts of different languages. To this end, our main proposal is the use of mBERT, a word encoder that is already trained in 104 languages [2]. Moreover, we train the model on multilingual data. Such a solution makes the resulting models more language-agnostic.

The second classification of models is XABSC. XABSC is different from MABSC in that it requires sentiment prediction of text in one target language while only having training data in another source language. Therefore, it is possible with such a model to train only using data of a resource-rich language, such as English, and predict on a text of a resource-poor language, such as Dutch. Moreover, translation techniques and Aspect-Code-Switching (ACS) [8] can be applied to enrich the training data. The enriched dataset can then enhance the bilingual capabilities of the model.

We also define a third task, which is called Unilingual Aspect-Based Sentiment Classification (UABSC). As its name suggests, the type of model for this task is geared towards a single language, meaning training on one language and testing on that same language. The usual ABSC models would fall under this classification, as they are solely trained in English and tested in English. However, in this paper, models trained solely on English data are not included under UABSC, as these are not our primary focus.

The models for the three tasks are based on LCR-Rot-hop++, the deep learning ABSC model of [7], which achieved state-of-the-art results for the ABSC task. This model was previously trained only in English.

First, for MABSC, we utilize mBERT to train a multilingual version of LCR-Rot-hop++, named mLCR-Rot-hop++, using only English data. Then we employ data in different languages, combine them to form a multilingual dataset, and train mLCR-Rot-hop++ on it. The resulting model is called MLCR-Rot-hop++.

Second, for XABSC we use mLCR-Rot-hop++ as an XABSC model as well, since it does not require training data in a language other than English. Furthermore, we train mLCR-Rot-hop++ on data translated from English to the target language. These models are specific to the language they were trained for and are encapsulated by the general name mLCR-Rot-hop-XX_{en}++. XX stands for one of the three resource-poor languages that we use, namely NL for Dutch, FR

for French, and ES for Spanish. Then, using ACS we construct models under the name mLCR-Rot-hop-ACS$_{xx}$++, where aspects in different languages (codes) are swapped for enlarging the original dataset.

Last, the UABSC models called mLCR-Rot-hop-XX++ were formed using mLCR-Rot-hop++ and training it on Dutch, French, and Spanish individually.

The contribution of this paper is twofold. First, we strengthen the capabilities for MABSC by using mBERT into a state-of-the-art ABSC model, LCR-Rot-hop++, and by training it on annotated data in four languages, producing models which are language-agnostic, which is one of the goals of this paper. Second, as not all languages are resource-rich when it comes to ABSC, we help to circumvent this issue by applying mLCR-Rot-hop++ to XABSC as well as MABSC. However, since the performance of basic XABSC models mostly depends on the quality of translation services, we also implement a data-enriching technique, ACS. Last, we focus on unilingual models for our considered languages, but English. The Python source code is publicly available at `https://github.com/Steinar2049/mLCR-Rot-hop-plus-plus.git`.

The remainder of the paper is structured as follows. Section 2 describes the data used in this research. Then, the inner workings of mLCR-Rot-hop++ and its various extensions are explained in Sect. 3. Afterward, the main results of our evaluation are presented in Sect. 4. Last, we present our conclusions and suggestions for future work in Sect. 5.

2 Data

The data used in this research hails from the SemEval-2016 dataset, which provides labeled training and test data in eight languages. It makes available data on four domains: restaurants, electronics, hotels, and telecom. We only consider the data on restaurant reviews, as it is the most supported across languages. The data for the restaurant reviews are available in six languages, namely English, Dutch, French, Russian, Spanish, and Turkish. Turkish, however, is not well supported, as the total data consists of only 339 sentences. Moreover, Russian has the Cyrillic alphabet, which is different from the remaining languages. As the four other remaining languages are quite related to each other, it allows us to more easily compare the different approaches further discussed in this paper, so choosing languages that are very different could lead to very different results in different methods, which we leave as a topic for another research. Hence, we focus on English, Dutch, French, and Spanish for our research.

The data consists of labeled restaurant reviews in XML format. An example of a sentence in a review is shown in Fig. 1. Each sentence contains a"text" element and an "Opinions" element. The text element contains the writings of the reviewer and may or may not include an aspect. The "Opinions" element consists of "Opinion" elements corresponding to the aspects present in the sentence. Each "Opinion" element has a list of attributes, containing the aspect, its category, its polarity, and its boundary positions. For example, as shown in Fig. 1, the sentence is "Salads are a delicious way to begin the meal.". The aspect in

the sentence is “Salads” and is part of the category of food quality. Furthermore, the sentiment towards this aspect is positive, as given by the polarity attribute, and the aspect is found in the sentence at index 0 to and not including 6.

```
<sentence id="16611043:1">
<text>Salads are a delicious way to begin the meal.</text>
<Opinions>
<Opinion target="Salads" category="FOOD#QUALITY"
         polarity="positive" from="0" to="6"/>
</Opinions>
</sentence>
```

Fig. 1. A sentence from the restaurants SemEval-2016 dataset.

Next, Table 1 summarizes the data for each language. In the table, the data per language is divided into the training and the test data. For each type of dataset, the number of aspects is shown, hence also the number of sentiment predictions to be made. Furthermore, the percentages of positive, negative, and neutral labels are given for each dataset. Notably, all datasets contain more positive labels than negative or neutral. Moreover, the English and Spanish data have relatively more positive labels compared to the French and Dutch data.

Table 1. Statistics of the restaurants SemEval-2016 dataset.

Language/Aspects	Train	Positive	Negative	Neutral	Test	Positive	Negative	Neutral
English	1880	70.160%	26.011%	3.830%	650	74.308%	20.769%	4.923%
Dutch	1283	59.080%	31.723%	9.197%	394	62.183%	31.726%	6.091%
French	1770	50.904%	42.542%	6.554%	718	50.696%	39.694%	9.610%
Spanish	1937	70.676%	24.729%	4.595%	731	71.272%	24.077%	4.651%

As to the data cleaning, we remove any sentiment labels pertaining to hidden aspects (less than 25% of the data). The numbers shown in Table 1 are after removing these labels. We remove these labels, as our models are not made to deal with hidden aspects. Furthermore, since we want to use ACS, we need to have the aspects present in the sentence to be able to switch them.

Furthermore, we removed any labels that were not valid anymore after the translation techniques or after applying the ACS method. It could often occur that an aspect disappeared after translation, or sometimes the aspect boundaries became incorrect. Hence, we removed the labels for which these events occurred. The total amount of aspect labels remaining per language are shown in Table 2 with the number of removed labels between brackets. The dataset per language is divided into two phases: a training phase and test phase. The training phase

has four subcategories: {XX, XX_{en}, $EN_{acs\text{-}xx}$, $XX_{acs\text{-}en}$}. XX is the normal set corresponding to one of the four languages. XX_{en} is the set corresponding to the English data translated to XX, one of the other three languages. $EN_{acs\text{-}xx}$ corresponds to the set of English data of which the aspects are code-switched with the aspects of XX_{en}. $XX_{acs\text{-}en}$ is the set of translated data XX_{en} with the aspects code-switched with the English dataset.

Table 2. Number of opinions per type of dataset and number of removed opinions in brackets.

Phase	Type	EN	NL	FR	ES
Train	XX	1880 (627)	1283 (577)	1770 (760)	1937 (783)
	XX_{en}	-	1861 (646)	1849 (658)	1851 (656)
	$EN_{acs\text{-}xx}$	-	1880 (627)	1880 (627)	1880 (627)
	$XX_{acs\text{-}en}$	-	1861 (646)	1849 (658)	1851 (656)
Test	XX	650 (209)	394 (219)	718 (236)	731 (341)

3 Methodology

In this section we showcase our models, the ideas behind them, and how to implement them. First, in Subsect. 3.1 we explain our base model, mLCR-Rot-hop++, in detail. Second, in Subsect. 3.2 we show the MABSC models. Third, in Subsect. 3.3 the XABSC models are presented. Fourth, the UABSC models are explained in Subsect. 3.4. Last, in Subsect. 3.5 we give the performance measure used for model evaluation. In Table 3 an overview is provided of all the models and under which classification they fall.

Table 3. Overview of the proposed models and their classification.

Model/Classification	MABSC	XABSC	UABSC
LCR-Rot-hop++	-	-	-
mLCR-Rot-hop++	x	x	-
MLCR-Rot-hop++	x	-	-
mLCR-Rot-hop-XX_{en}++	-	x	-
mLCR-Rot-hop-ACS_{xx}++	-	x	-
mLCR-Rot-hop-XX++	-	-	x

3.1 Base Model: mLCR-Rot-hop++

First, we introduce the base model, the backbone for all the other models presented in this paper. The model, mLCR-Rot-hop++, is heavily based on LCR-

Rot-hop++ [7], but with the important distinction that mLCR-Rot-hop++ uses mBERT as embedder instead of monolingual BERT [2]. The model itself is an attention-based neural network model that operates on sentence level and notably takes the context of the target aspect and assesses it from both sides of the target, the left context, and the right context. These elements of the sentence are embedded using mBERT so that further processing can be done. The network is not simply a feedforward neural network but uses rotary attention to repeatedly adjust the sentence representations of the model. Another addition to the model is that it introduces hierarchical attention, which exchanges local information with the other side of the network to improve the forming of the vector representations in a set number of iterations (hops).

After the hops in the rotary attention mechanism, the final vector representations are concatenated and supplied to the MLP layer. The MLP layer then makes a final prediction for the sentiment. Probability vectors p_j are evaluated using a cross-entropy loss function with an L2 regularization term,

$$\underset{1\times1}{L} = -\sum_{j=1}^{P} \underset{3\times1}{s_j} \times \log(\underset{3\times1}{p_j}) + \lambda||\beta||_2^2, \tag{1}$$

where s_j is the actual sentiment vector of instance j and λ is the penalty term for the square of the L2 norm of the parameter set β. As we have three sentiment classes, p_j and s_j have dimensions 3.

Last, note that the weights in the model are initialized using the uniform distribution and are optimized with backpropagation using stochastic gradient descent. Moreover, the model's hyperparameters are tuned using a Tree-Structured Parzen Estimator (TPE) algorithm. Hyperparameters include learning rate, dropout rate, momentum, weight decay, and number of hops. A more detailed explanation of LCR-Rot-hop++ can be found in [7].

As shown in Fig. 2, mLCR-Rot-hop++ can be applied to text in EN (English), NL (Dutch), FR (French), and ES (Spanish). Therefore, it is considered an MABSC model. The original LCR-Rot-hop++ can also be applied in this way, but as that model has no elements incorporating multilingualism, its results for languages other than English can be interpreted as chance outcomes or predictions for text vaguely reminiscent of English. Therefore, mLCR-Rot-hop++ is expected to improve upon the base model, as it utilizes pre-trained multilingual embeddings.

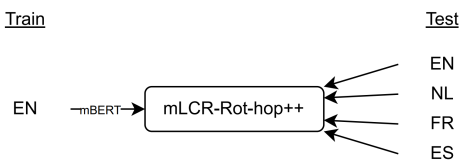

Fig. 2. mLCR-Rot-hop++.

3.2 MABSC Models

Here we use the labeled datasets of all four languages to our disposal and combine them into one large multilingual dataset and train an LCR-Rot-hop++ model on it. The resulting model is called MLCR-Rot-hop++. mLCR-Rot-hop++ also falls under MABSC category, as it can embed data of any of the four languages. mLCR-Rot-hop++ was already explained in the previous subsection, hence we do not highlight it here.

Naturally, we compare MLCR-Rot-hop++ with mLCR-Rot-hop++ to see whether the model benefits from language specific data. However, we also consider the comparison between MLCR-Rot-hop++ and the UABSC mLCR-Rot-hop-XX++ models in Subsect. 3.4 to see whether mixing the different languages in one dataset benefits or detriments the prediction results or whether there is no difference. This model is depicted in Fig. 3.

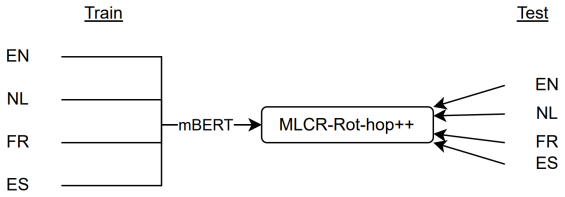

Fig. 3. MLCR-Rot-hop++.

3.3 XABSC Models

In this subsection we present the XABSC models. We have three models: mLCR-Rot-hop++, mLCR-Rot-hop-XX_{en}++, and mLCR-Rot-hop-ACS_{xx}++. As the base model, mLCR-Rot-hop++, has already been explained in Subsect. 3.1 , we skip its presentation here.

mLCR-Rot-hop-XX_{en}++. In order to perform ABSC for resource-poor languages, we make use of translation techniques. We use translation machines to translate the English dataset to a Dutch, French, and Spanish one. In this view, English is the source language, and the others are the target languages. Then, in a similar fashion to the UABSC models discussed in Subsect. 3.4, we train mLCR-Rot-hop++ on each translated dataset individually. An overview of the process is provided in Fig. 4. The individual models are called mLCR-Rot-hop-XX_{en}++, with XX $\in$ {NL, FR, ES} and are classified as XABSC models. The translated training data for language XX is denoted as XX_{en}.

We hypothesize that the mLCR-Rot-hop-XX_{en} models do not perform as well as the UABSC models, as the translated data is expected to be of lesser quality. The translated data is not manually curated, hence its quality depends on the

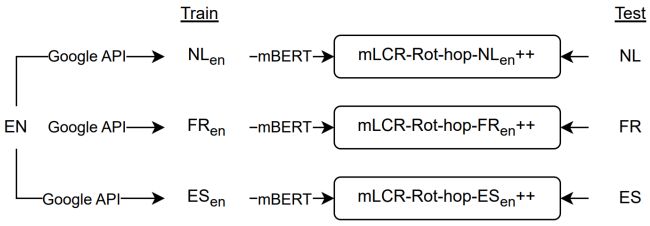

Fig. 4. mLCR-Rot-hop-XX$_{en}$++, with XX $\in$ {NL, FR, ES}.

translation service and the pseudo-labels constructed afterward. However, the focus of these XABSC models is to perform well without the use of procured data in other languages.

The translation procedure works as follows. Each sentence in the original dataset has corresponding aspect labels, denoting the aspect words in the sentence. When the sentence is translated, the aspect changes. Hence, the aspect label needs to be changed accordingly. There are multiple approaches one can take. First, it is possible to directly translate the aspect label and then to find it again in the sentence. The technique falling under this category is called Translate-then-Align (TA) [8]. However, the aspect in the sentence and in the aspect label can translate to different words because depending on the context a different word, word combination, or synonym can be used in the target language. To this end, [8] proposed Alignment-Free (AF) label projection. This technique first marks the aspects in a sentence of the source data, translates the sentence, and then uses the markings to find the translated aspect. The markings are special symbols like "[]", and are placed around the aspect. It is shown that this technique results in 25 percentage points fewer mistakes in labeling compared to TA.

The most common options for translation are Google API and DeepL. DeepL is considered to be a bit more accurate, but Google provides translation on 130+ languages, while DeepL only supports 30+ so far. However, the two services respond differently to the AF technique. For longer sentences, DeepL simply removes the markings by itself, rendering the AF technique useless. Contrastingly, Google API keeps the markings. Moreover, since the aspect is between brackets, the Google implementation seems to think the surrounding context is less important to the aspect. As a result, the surrounding words can lose a bit of context, or the aspect is translated more directly. However, as we want to implement the best alignment technique, AF, we chose Google API. Moreover, it supports more languages, and thus this engine helps ABSC for resource-poor languages.

mLCR-Rot-hop-ACS$_{xx}$++. Here we present the XABSC model using ACS. As explained before, ACS is a way to enrich the dataset. The first step in ACS is to translate the source data to a target language. In this paper, we use AF for aspect projection, as described before. After translating and obtaining the aspects for the original sentence and the translation, we swap the corresponding

aspects. The aspects are then code-switched and we obtain two more labeled sentences, both of which are bilingual.

Let us consider English and Spanish. The first bilingual sentence contains the English context but now the Spanish aspect. The second contains the Spanish context but the English aspect. An example is shown in Fig. 5, where the aspects "food and "service" in the English sentence become "comida" and "servicio" in Spanish, respectively. Afterward, these aspects are switched between sentences, creating two bilingual sentences.

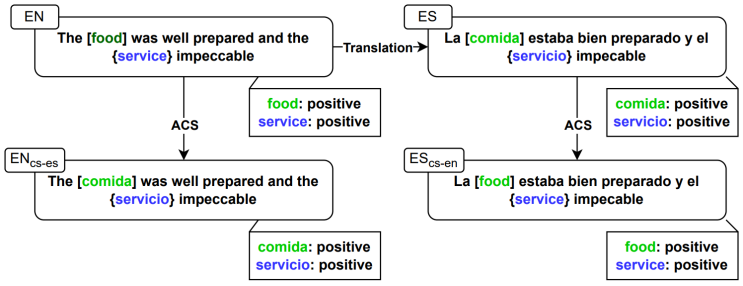

Fig. 5. ACS applied for Spanish.

Besides switching the aspects themselves, the aspect labels have to be carried with them. On account of the position index of the switched aspect being different, the new aspect has to be located again. So, once again AF is used to find the aspect and its location.

This procedure can be done for each target language and supplies each model with four times the amount of data and on top of that provides bilingual sentences. Hence, the model should become more language-agnostic on account of this method [8]. To keep track of these datasets we label the English data that was code switched as $\text{EN}_{\text{acs-xx}}$, where acs stands for aspect code-switched and xx $\in$ {nl, fr, es} for the language of the translated data. Then, we label the code-switched translated data as $\text{XX}_{\text{acs-en}}$, where XX and "en" indicate it was code-switched with English data, and XX $\in$ {NL, FR, ES}. These are depicted in Fig. 6.

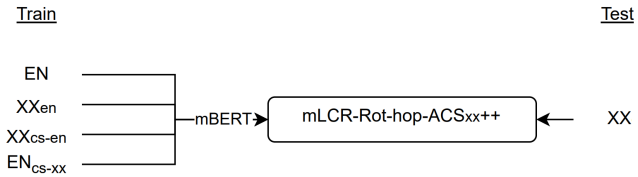

Fig. 6. mLCR-Rot-hop-ACS_{xx}++, with xx $\in$ {nl, fr, es}.

3.4 UABSC Models

Now we introduce the use of labeled datasets in languages that are not English. [5] provides labeled training and test data for the four languages we consider. Hence, we can train mLCR-Rot-hop++ on each individual dataset. Then, each model is tested only for the language it is trained in. Therefore, these models fall under the UABSC classification. and we define them as mLCR-Rot-hop-XX++, with XX $\in$ {NL, FR, ES}. These are depicted in Fig. 7.

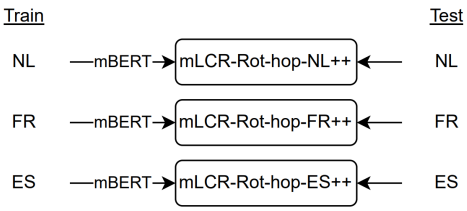

Fig. 7. mLCR-Rot-hop-XX++, with XX $\in$ {NL, FR, ES}.

3.5 Model Evaluation

To evaluate the performance of a model, we use the number of correct predictions. We define a prediction as correct when it predicts the sentiment label correctly. Accuracy is defined as the total number of correct predictions over the number of predictions.

In Sect. 2 we showed the most occurring label is positive by a high margin, especially for the English and Spanish dataset. Therefore, the accuracy score is biased upwards for models that are trained to predict positive sentiment more. However, we still choose accuracy as it is easily interpretable, and the measure used in many related studies.

4 Results

In this section, we discuss our most important results and analyze where our models went right and where they went wrong. First, we compare the accuracy across the MABSC models in Subsect. 4.1 and evaluate which option provides the best performance. Second, in Subsect. 4.2, the proposed XABSC models are evaluated and compared with benchmarks. Last, we compare the UABSC models to the MABSC models in Subsect. 4.3.

4.1 MABSC Results

In this subsection, we analyze the performance of the MABSC models. In Table 4 the accuracy scores are shown, with the best score for each language in bold.

We also show the average over the languages, to get a single indicative measure of the model's performance. Moreover, since English has been the focus of ABSC studies, and we aim to add to the multilingual capabilities of the field, we also consider the average without English.

Table 4. Accuracy scores of MABSC models.

Model/Language	English	Dutch	French	Spanish	Average	Average**
LCR-Rot-hop++	**87.38***	62.18*	50.56*	70.73*	67.71*	61.16*
mLCR-Rot-hop++	80.15	67.26	**67.13**	**77.43**	**72.99**	70.61
MLCR-Rot-hop++	78.31	**70.05**	66.57	76.20	72.78	**70.94**

*obtained with BERT embeddings **average without English

For the accuracy scores, we see that on average mLCR-Rot-hop++ performs best, with a score of 72.99. However, when removing English from the average, MLCR-Rot-hop++ obtains the highest accuracy, with a score of 70.94. Moreover, the average scores of both models are within 0.5 percentage points of each other. Hence, the two models perform quite closely. We also see this reflected in the accuracy scores for the individual languages, where mLCR-Rot-hop++ performs better for French and Spanish with scores of 67.13 and 77.43, respectively, but MLCR-Rot-hop++ performs better for Dutch with a score of 70.05. The best accuracy for English is obtained by LCR-Rot-hop++, which makes sense, as it is only trained on English data using normal BERT embeddings. That also explains the relatively weak performance of LCR-Rot-hop++ for the other languages.

That mLCR-Rot-hop++ outperforms MLCR-Rot-hop++ for French and Spanish is surprising, as MLCR-Rot-hop++ was trained on French and Spanish data, while mLCR-Rot-hop++ was not. A possible explanation could be the high similarity between these Latin based languages, which possibly confuses the model in learning meaningful patterns.

4.2 XABSC Results

In this subsection we review the results for the XABSC models and compare them. As mLCR-Rot-hop++ uses mBERT, it can be used as a XABSC model in the "zero-shot" fashion, i.e., just relying on mBERT capabilities to process different languages.

Next, we compare the accuracy scores of our models in Table 5 and we see a surprising result, as the models do not progressively get better at accuracy, the more complex they become. mLCR-Rot-hop-ACS_{xx}++ has the strongest result for Dutch, while mLCR-Rot-hop++ has the highest accuracy for French, Spanish, and on average. The accuracy gap between mLCR-Rot-hop++ and mLCR-Rot-hop-XX_{en} is around 5.5 percentage points on average and between mLCR-Rot-hop++ and mLCR-Rot-hop-ACS_{xx}++ 3.5 percentage points on average. So, we can conclude that mLCR-Rot-hop++ is our best XABSC model in terms of accuracy.

Table 5. Accuracy scores of XABSC models.

Model/Language	Dutch	French	Spanish	Average
mLCR-Rot-hop++	67.26	**67.13**	**77.43**	**70.61**
mLCR-Rot-hop-XX_{en}++	67.77	59.96	67.72	65.15
mLCR-Rot-hop-ACS_{xx}++	**68.53**	60.03	72.78	67.11

4.3 UABSC Results

In this subsection, we evaluate the UABSC models and compare the results with that of the MABSC models to see whether there is a difference in performance. We only compare UABSC with MABSC, as we are interested in whether there is a difference in performance between using a model trained and tested in one language other than English and using an all-purpose model. The XABSC models are a bit different as there is interference with the data, using translation and ACS. Hence, we do not include them in this comparison. We check the accuracy in Table 4 and Table 6 to see which models perform best overall and if there are differences across languages. From both tables we see a pattern: mLCR-Rot-hop-XX++ models perform best. The models have on average around 2.4 percentage points higher accuracy score than the second-best, MLCR-Rot-hop++. We conclude from this result that training a model on a dataset specific to one language does improve prediction performance. It is therefore not the case that making the model multilingual, as with MLCR-Rot-hop++, does not affect performance. The cause for this is likely due to interference of other languages in the training set that bring noise to the performance for the specific test language.

Table 6. Accuracy scores of UABSC models.

Model/Language	Dutch	French	Spanish	Average
mLCR-Rot-hop-XX++	**71.57**	**70.75**	**77.70**	**73.34**

5 Conclusion

To conclude, we set out to create MABSC, XABSC, and UABSC models from the state-of-the-art LCR-Rot-hop++ using various data techniques and present the best models for each task. Out of the proposed MABSC models both mLCR-Rot-hop++ and MLCR-Rot-hop++ are competitive, but MLCR-Rot-hop++ performs better when we do not consider English. As previously mentioned, we are interested in extending ABSC to languages other than English. Hence we consider MLCR-Rot-hop++ as the best MABSC model.

For our models in the XABSC category, mLCR-Rot-hop++ achieved the highest accuracy overall and so is the best XABSC model. It is interesting to

note that this model has been trained for ABSC for the English language only. Furthermore, we considered models that were trained in one specific language, the UABSC models, and we found results that confirm that a multilingual model does lose performance compared to a unilingual model. Specifically, the mLCR-Rot-hop-XX++ models outperformed the all-purpose MLCR-Rot-hop++.

The MABSC field is relatively unexplored, there are more possibilities to devise new models that include LCR-Rot-hop++. For example, the translated and ACS data can be combined into a multilingual dataset and be passed to LCR-Rot-hop++ in the same way as MLCR-Rot-hop++ is constructed. Also, in order to further improve the results of our XABSA models, we plan to investigate to use of the CL method proposed by [3].

References

1. Brauwers, G., Frasincar, F.: A survey on aspect-based sentiment classification. ACM Computing Surveys **55**(4), 65:1–65:37 (2023)
2. Devlin, J., Chang, M.W., 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 Technology (NAACC-HLT 2019) 4171-4186 ACL (2019)
3. Lin, N., Fu, Y., Lin, X., Yang, A., Jiang, S.: CL-XABSA: Contrastive Learning for Cross-lingual Aspect-based Sentiment Analysis. arXiv preprint arXiv:2204.00791 (2022)
4. Liu, B.: Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, second edn. (2020)
5. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., et al.: SemEval-2016 task 5: Aspect based sentiment analysis. In: 10th International Workshop on Semantic Evaluation (SemEval-2016). pp. 19–30. ACL (2016)
6. Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering **28**(3), 813–830 (2016)
7. 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 International Conference on Web Engineering, (ICWE 2020). LNCS, vol. 12123, pp. 365–380. Springer (2020)
8. 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)