Cross-Lingual Aspect-Based Sentiment Analysis with Multilingual BERT

Erik (Hsiang-Jen) Hou

Abstract

We applied multilingual BERT (mBERT) on the document-level laptop review dataset from SemEval-2016 for aspect-based sentiment analysis task and tested the model's ability to perform zero-shot cross-lingual learning transfer from English to Chinese. Results suggest multilingual BERT's ability to transfer its learning of complex text relationship at document-level from English to Chinese. We also demonstrated and discussed about the challenges from imbalanced data distribution for aspect-based sentiment analysis with a large number of aspect categories.

1 Introduction

The goal for this project is to experiment with cross-lingual transfer learning in the context of aspect-based sentiment analysis on customer reviews in the hopes of tapping into abundant potential applications and benefits. Aspect-based sentiment analysis (ABSA) is a subtask of sentiment analysis that aims to discover the fine-grained sentiment towards specific aspect of an entity. It is a question that has motivated many researchers in both academia and industry alike due to the benefits it can bring by letting companies and organizations understand granular information from a large amount of text reviews and comments. As NLP research advances by leaps and bounds in recent years, so has the models' ability to perform ASBA task.

However, NLP resources are still more English centric even though English speakers only account for roughly 16% of the world's population (1.2 billion out of the world's 7.5 billion population speaks some level of English)[1]. Clearly, there is a need to enable the generalization of NLP resources and model applications to other languages.

2 Background

BERT (Devlin et al. 2019) garnered great attention because of its ability to achieve state-of-the-art performance in many NLP tasks. This includes aspect-based sentiment analysis (ASBA). Sun et al. (2019) utilized BERT for ASBA by constructing auxiliary sentences and convert the ABSA task to a sentence-pair classification task and achieved state-of-the-art results on SentiHood and SemEval-2014 Task 4 datasets.

While the same method can be applied to many other languages, it would not be feasible to train and maintain separate language embedding models for individual languages. A universal language embedding is way to tap into data in other languages without having to train separate models. It also provides benefits to transfer learning from a high-resource language to a low-resource one. Cross-lingual language encoder such as multilingual BERT (mBERT) (Devlin, 2018) and XLMs (Conneau et al., 2019) were found to have the ability to achieve success in sentence classification tasks even in zero-shot learning (Wu et al., 2019; Conneau et al., 2019). With these promising results in the literature, we set out to conduct an experiment on a complex dataset by building on top of the work of Sun et al. (2019) and then attempt to transfer the model's learning from English to Chinese.

[1] https://en.wikipedia.org/wiki/List_of_countries_by_English-speaking_population

3 Methods

In this project, we will limit the experiment to transferring learning from English to Chinese because those two are the only languages the author is fluent in. Since the test set will need to be translated from English to another language to evaluate the effect of the cross-lingual transfer learning and we would like to rule out the influence of the potential incorrectness in machine translation on model performance, we will need humans to validate and correct the translation. In addition, we hope some benefits could come from looking at specific examples for error analysis and without fluency in both the source and target languages, this would not be possible.

3.1 Model Architecture

We use the NLI-M model proposed by Sun et al. (2019) whose input is the direct concatenation of tokenized entity-aspect combination and the review texts with a [SEP] token in between. The rest is the same as a BERT text classification model which has a linear classification layer on top of the encoder. According to Pires et al. (2019) even though multilingual BERT was not explicitly trained with objectives that attempt to align the embedding of corresponding vocabularies in different languages, it still exhibits surprising ability in zero-shot learning especially between two SVO (subject verb object) languages. Since Chinese is mostly an SVO language, we think it would appropriate to swap out the English only BERT-based model with the cased multilingual BERT for our transfer learning purpose. An illustration of model input and the expected output, and more details about data preprocessing which turns a review into training and test examples can be found in the supplementary materials section A.

3.2 Dataset Used

While the model built by Sun et al. (2019) achieved great results, it was tested on sentence-level reviews, where each of the sentences was labeled separately without considering the context provided by the surrounding sentences and both the SemEval 2014 and SentiHood datasets only had 5 and 8 aspect categories respectively. Since BERT supports sequence length up to 512 WordPiece tokens, we believe we could attempt to process the reviews at the document level without having to change the model's architecture. We eventually decided to use the document-level laptop reviews from SemEval 2016 task 5 after making sure the reviews can fit the sequence length limit. The reviews in this dataset were annotated at the document-level and were allowed to have 198 possible aspect categories associated with each review and 4 different sentiments associated with an identified aspect category. The rationale behind the selection of this dataset is that it will really test mBERT's ability to encode complex relationship into the final representation and transfer its learning from English to Chinese.

Our dataset contains 385 reviews for training and 80 reviews for testing, all in English. In order to evaluate the model's performance in zero-shot learning and compare it with the baselines, the 80 reviews are first translated into Chinese with machine translation by Google Translation API and then modified and validated by human for correctness.

For details about the EDA: https://github.com/erikhou45/ABSA-BERT-pair/blob/master/EDA.ipynb

3.3 Experiment Design

In order to test multilingual BERT's ability to transfer its learning on the ABSA task from English to Chinese. We split the training set into 305 reviews for training and 80 reviews for tuning both in English.

Then retrain the best performing model on the whole training set and test it on Chinese test set to compare its results with the two baselines described in Section 3.5.

3.4 Model Evaluation

To evaluate the performance of our models, we separate the aspect-based sentiment analysis task into two subtasks which are aspect category detection and sentiment polarity detection and evaluate the model's performance on them separately. The aspect category detection is about determine whether an aspect category is mentioned in a review irrespective of the sentiment predicted about it. Performant aspect detection means being able to identify as many aspect categories associated with a review correctly without making false identification. We will use the F1 score as the primary matric for this subtask. On the other hand, the sentiment polarity detection subtask is about given an aspect category and a review, predict the correct sentiment. We will use accuracy as the metric for this subtask; in addition, we will calculate the accuracy when the minority classes (conflict and neutral) are removed sequentially. This will enable us to see, if the model's performance keeps improving in the absence of less polarized classes. If it does, it means the model assigns more probability weights to the correct polarity.

3.5 Baselines

We propose two baselines in the experiment to help evaluating the models performance.

3.5.1 Naïve baseline

This baseline is language agnostic and simplistic. It is created by simply counting through the entire training set. On average there are about 5 aspect categories associated with each review. Therefore, the Naïve baseline always predicts the most common five aspect categories for every review. As for sentiment detection, we predict the most common sentiment for a given aspect category after aggregating the training set and if there is a tie, the model will predict "positive" sentiment since it is the most common sentiment across all aspect categories. This model will serve as the lower bound performance of our final cross-lingual model. We think that our final model at least needs to outperform the naïve baseline to say that it has some effect.

3.5.2 English only

This baseline is trained on the same data but evaluated on the original English test set. This is to give an idea of how the model performs the ASBA task without the language mismatch on the training and test ends. We think this baseline will be the performance upper bound of our cross-lingual transfer.

3.5 Training and Tuning

3.5.1 First iteration

In the first tuning iteration, we simply feed the training data into the model, look at the loss on the training and dev sets to make sure the model has the capacity to memorize information from the training set and that it is making reasonable predictions on data it has not seen before.

From looking at the losses, we confirmed that the model can memorize the training set. In addition, the model does a decent job detecting the sentiments. However the performance of the aspect category detection task on the development set suffered from low recall (i.e., the model fails at detecting the

entity-aspect mentioned in the reviews). Looking at the distribution of training examples in different labels, we noticed that it is very imbalanced (see table B1 in supplemental materials section B). Because most of the examples created during preprocessing have the label, "none", we think this makes the model too inclined to predict that an aspect category isn't mentioned in a review. Looking at literature, in a study using BERT for sentence classification with imbalanced data, Madabushi et al (2020) mentioned adjusting the cost function or equivalently oversampling the examples in the minority class could increase recall without hurting precision too much.

3.5.2 Second iteration

We took a grid search approach in finding a good sample proportion that will improve the model's performance on the dev dataset. Since Sun et al. had really good results in their experiment, we started the search by adjusting the sample proportion to have roughly the same proportion of examples associated with the label, "none", which is about 75%. In addition, due to large number of examples we have in the training set, each training epoch takes about an hour even when using 4 Nivida Tesla T4 GPUs (total of 60GB of memory). Therefore, we started with a mix strategy of under-sampling (randomly dropping "none" examples) and over-sampling (duplicate all non-none examples). The reason to including under-sampling is so that we could also speed up the training by reducing the number of training examples. Looking at the results, we are able to increase the recall. Yet, it was achieved at great expense of precision.

3.5.3 Final iteration

Looking at the results from the second iteration, we think there are two factors that might have caused the significant drop in precision, one is the proportion between different labels. Like what was mentioned in section 3.1, the dataset Sun et al. used only entails five aspect categories, by nature, it should have much fewer "none" examples. Adjusting our sampling to have the same distribution probably distorted the distribution too much and adversely affect the model's performance. Secondly, we realized when looking at the training data distribution across different aspect categories, that the percentage of non-none records were not evenly distributed. Therefore, under-sampling by randomly dropping none examples, might not be an efficient strategy to prompt models to detect the aspect categories where we need them to recognize the most. The distribution of non-none examples in the training set is provided in supplementary materials figure B2. Therefore, in the final iteration we only slightly increase the proportion of non-none examples by only using oversampling. We also experiment oversampling different sentiment labels with different weights such that we duplicate more "neutral", "conflict" than "negative" and "positive" examples. All the models' performances on dev set are provided in table 2. For details about the sampling strategies, please see the supplemental materials B2.

			Aspect Category Detection			Sentiment Detection
Iteration	Sampling	Model	Precision	Recall	F1-Score	Accuracy
1	NA	Basic	75%	41%	0.53	81%
2	mix	Combo-Sampling-1	50%	71%	0.58	80%
2	mix	Combo-Sampling-2	50%	71%	0.59	77%
2	mix	Combo-Sampling-3	47%	70%	0.56	70%
3	over-sampling	Over-Sampling-1	67%	59%	0.63	75%
3	over-sampling	Over-Sampling-2	63%	63%	0.63	76%

Table 2: The evaluation metrics on the models trained during tuning. We select Over-Sampling-2 model from iteration 3 to be evaluated on the test data since it has the highest F1-Score and better sentiment accuracy than Over-Sampling-1 in the same iteration. The details about the sampling strategy are provided in supplementary materials section B2.

4 Result and Discussion

The best performing model from tuning is retrained on all the examples in the original training set and then evaluated on both the English and Chinese test sets. The test results of aspect category detection task and sentiment polarity detection tasks are presented in table 3.

	Aspect Category Detection			Sentiment Detection		
Model	Precision	Recall	F1-Score	4-Class Acc.	3-Class Acc.	2-Class Acc.
Naïve Baseline	55%	41%	0.47	63%	65%	70%
English-Only Baseline	66%	57%	0.61	71%	73%	80%
Cross-Lingual	72%	44%	0.54	69%	72%	78%

Table 3: The evaluation performance of the selected model from tuning compared with the two baselines. We also provided the 3-class and 2-class accuracies to show the model's performance in sentiment detection task if rare classes such as, "conflict" and "neutral" are removed sequentially.

From table 3, can see that the model outperforms the naïve baseline in all metrics for both subtasks. Also, as predicted there is some loss in performance during the cross-lingual transferring; therefore, the zero-shot learning consistently underperforms the English-only baseline.

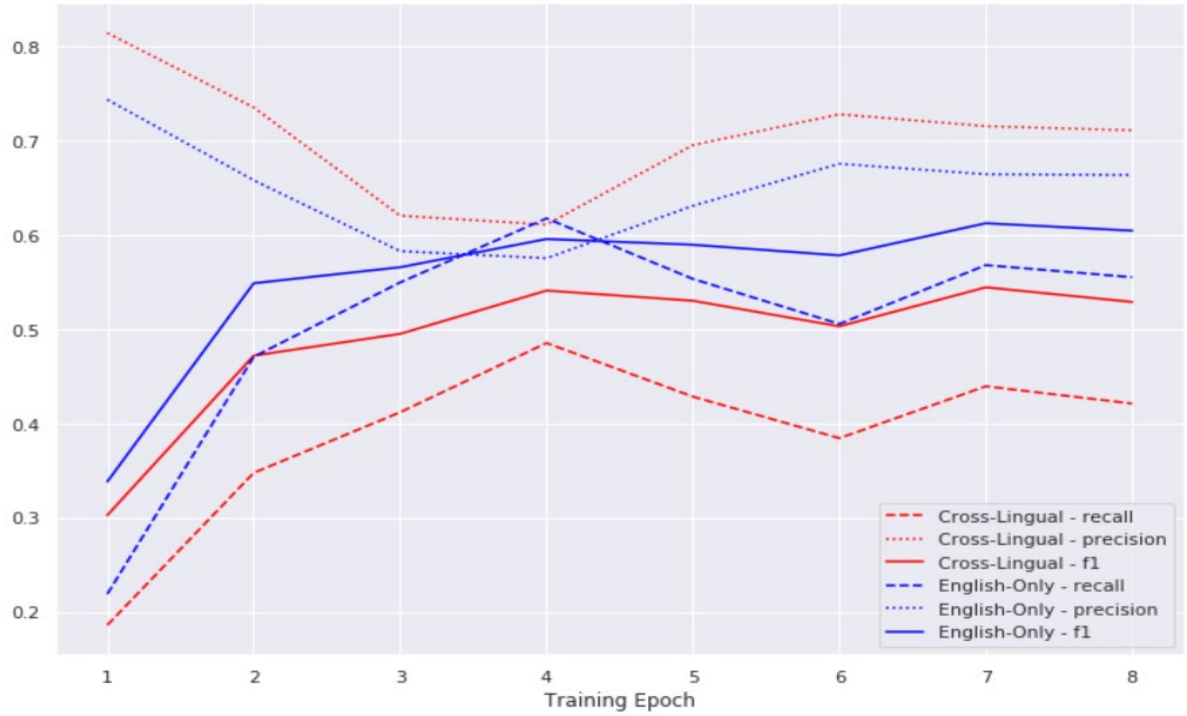


Figure 3: Aspect detection task metrics of test sets by training epoch.

In addition, we noticed some interesting trends about the model when it is evaluated on Chinese and English test sets along different training epochs (See figure 3). Firstly, both the model's evaluation on Chinese and English test set results are extremely correlated across training epochs (the corresponding lines are almost parallel). This means what is adjusted when the model is being trained in English gets reflected in the same direction and even magnitude no matter the language in which the test set is.

Secondly, the zero-shot learning seems to be consistently more conservative (always having a higher precision and a lower recall). Therefore, if the objective is to further increase F1 score of the zero-shot learning, more over-sampling of the non-none examples is likely to increase the performance of the aspect category detection task a little bit.

While the alignment and consistency in figure 3 are really amazing given multilingual BERT was never trained with language alignment as its objective explicitly. However, the strong correlation and consistency also made us think whether mBERT interprets the translated Chinese by Google Translation API in a similar way to how it interprets the original English text the Chinese text is translated from. Though we manually modify almost every review during validation for the machine translation, around 80%-90% of the vocabulary and sequences were left unchanged. We think further testing is needed such as manually translating a random subset of the test set without using Google Translate and retest to see if the strong correlation still exists.

5 Conclusion

In this study, we applied Sun et al.'s model in conjunction with multilingual BERT on document-level aspect-based sentiment analysis task. Due to the challenges caused by the imbalance in the dataset, we attempted different sampling strategies to prompt the model to make better predictions in the aspect category detection task. It results in some improvement but it is limited.

During testing, we tested the model's ability to transfer its learning from English to Chinese and the results suggest a strong correlation between model's reactions to the English and Chinese test examples. We also suggested further tests to isolate source of this correlation.

In addition, to improve the cross-lingual transfer learning, some future directions include freezing the lower-level embedding as suggested by Wu et al. (2019) or using the cross-lingual models by Conneau et al., (2019), which has been shown to outperform mBERT.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence.

Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining

Telmo Pires, Eva Schlinger, Dan Garrette. 2019. How multilingual is Multilingual BERT?

Harish Tayyar Madabushi, Elena Kochkina, Michael Castelle. 2020. Cost-Sensitive BERT for Generalisable

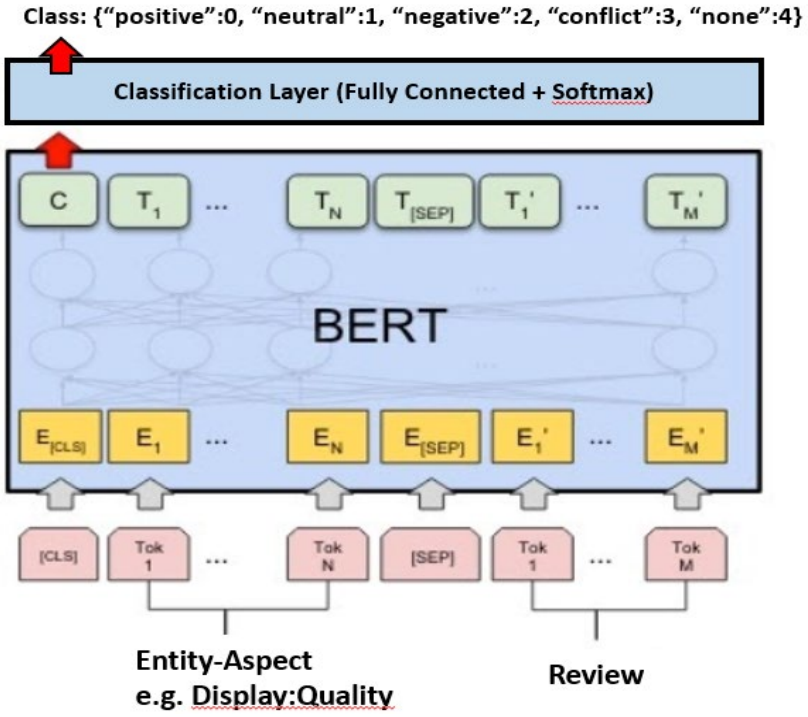
Sentence Classification with Imbalanced Data

Jacob Devlin. 2018. Multilingual BERT readme document.

Supplementary Materials

Section A: Data Flow

A1 Model Input and Output Illustration



A2 Data Preprocessing

Each review in the dataset can be associated with 22 entities about laptops which are: {*Laptop, Display, CPU, Mother Board, Hard Disc, Memory, Battery, Power Supply, Keyboard, Mouse, Fans Cooling, Optical Drives, Ports, Graphics, Multimedia Devices, Hardware, OS, Software, Warranty, Shipping,* Support, Company} and each entities can be associated with 9 aspects which are {*General, Price, Quality, Operation Performance, Usability, Design Feature, Portability, Connectivity, Miscellaneous*}. This results in 198 aspect category combinations. Therefore, each review generates 198 examples. And an example follows this following format:

[REVIEW_ID] "\t" [SENTIMENT] "\t" [ENTITY-ASPECT] "\t" [REVIEW_TEXT]

As for the sentiment, when a sentiment is associated with a certain aspect category combination for a review, the sentiment is record; otherwise, "none" is recorded to signal that a specific aspect category combination is not mentioned in a review.

Looking at a concrete example:

```
<Review rid="139">
    <sentences>
        <sentence id="139:0">
            <text>HP Pavilion DV9000 Notebook PC      When I first got this computer, it really rocked.</text>
        </sentence>
        <sentence id="139:1">
            <text>But as time went on I found it almost impossible to keep the thing on-line through wi-fi.</text>
        </sentence>
        <sentence id="139:2">
            <text>Eventually the screen went blank and the computer would not turn on.</text>
        </sentence>
        <sentence id="139:3">
            <text>HP said it was out of warranty.</text>
        </sentence>
        <sentence id="139:4">
            <text>Guess I'll stay away from HP.</text>
        </sentence>
    </sentences>
    <Opinions>
        <Opinion category="LAPTOP#GENERAL" polarity="negative"/>
        <Opinion category="LAPTOP#CONNECTIVITY" polarity="negative"/>
        <Opinion category="DISPLAY#OPERATION_PERFORMANCE" polarity="negative"/>
        <Opinion category="LAPTOP#OPERATION_PERFORMANCE" polarity="negative"/>
        <Opinion category="COMPANY#GENERAL" polarity="negative"/>
    </Opinions>
</Review>
```

This review will be turned into 198 examples. With all of them having the same [REVIEW_ID] = 139 and [REVIEW_TEXT] = the concatenation of all five sentences sequentially. Since there are five aspect categories found associated with this review. Therefore, five of the 198 examples will be:

139 “\t” negative “\t” LAPTOP-GENERAL “\t” [REVIEW_TEXT]
139 “\t” negative “\t” LAPTOP-CONNECTIVITY “\t” [REVIEW_TEXT]
139 “\t” negative “\t” DISPLAY-OPERATION_PERFORMANCE “\t” [REVIEW_TEXT]
139 “\t” negative “\t” LAPTOP-OPERATION_PERFORMANCE “\t” [REVIEW_TEXT]
139 “\t” negative “\t” COMPANY-GENERAL “\t” [REVIEW_TEXT]

Then there will be another 193 examples from the same review with the rest of [ENTITY-ASPECT] like this:

139 “\t” none “\t” [ENTITY-ASPECT] “\t” [REVIEW_TEXT]

Then before the examples are input into BERT for training or testing. The text in [ENTITY-ASPECT] and [REVIEW_TEXT] will be tokenized by WordPiece tokenizer and concatenated with a BERT [SEP] token in between. Also, the sentiment will be mapped in the following way: {“positive”:0, “neutral”:1, “negative”:2, “conflict”:3, “none”:4}.

Section B: Tuning

B1 Data Distributions

label	proportion
positive	0.014688
neutral	0.001672
negative	0.009074
conflict	0.000513
none	0.974052

Table B1: Training example distribution by class before sampling strategy adjustment

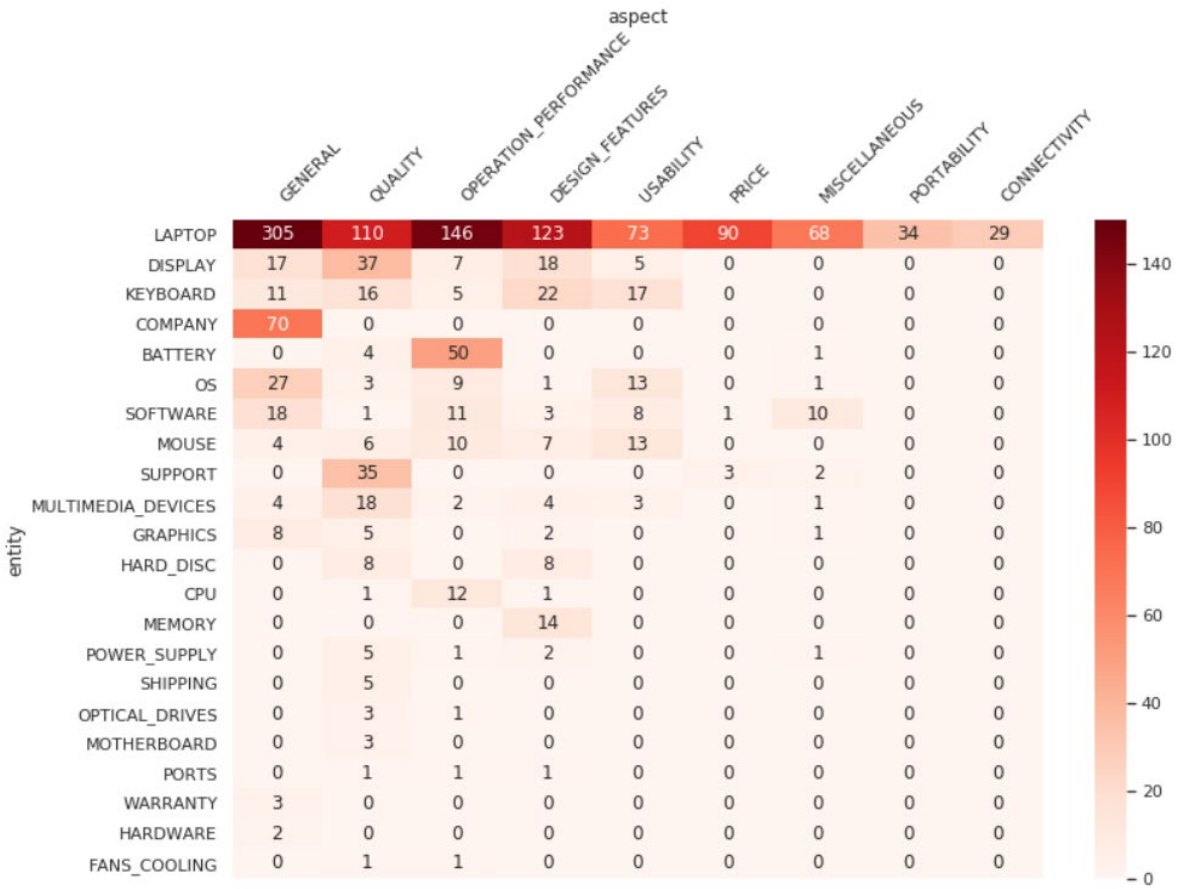


Table B2: Non-none training example distribution by aspect categories before sampling strategy adjustment. We think randomly dropping none examples isn't effective is because the majority of the aspect categories are only associated with class "none". Therefore, dropping those records wouldn't help with aspect detection. We think oversampling by duplicating non-none examples can directly emphasis the association of aspect categories with non-none classes.

B2 Sampling Strategies

Iteration 2:

Model Name: Combo-Samping-1

- Duplicate each non-none example by a factor of 2
- Randomly drop 50 percent of none examples

Model Name: Combo-Samping-2

- Duplicate each non-none example by a factor of 3
- Randomly drop 0.55 percent of none examples

Model Name: Combo-Samping-3

- Duplicate each non-none example by a factor of 5
- Randomly drop 60 percent of none examples

Iteration 3

Model Name: Over-Sampling-1

- Duplicate each non-none example by a factor of 2

Model Name: Over-Sampling-2

- Duplicate positive example by a factor of 1.5
- Duplicate neutral example by a factor of 3
- Duplicate negative example by a factor of 2.2
- Duplicate conflict example by a factor of 5