This laptop has great coffee: Training a Dutch ABSA model from customer reviews

FIEKE MIDDELRAAD, University of Twente, The Netherlands

Aspect-based sentiment analysis (ABSA) finds the sentiment of a feature in a text. It can be useful for both businesses and customers to get an overview of product reviews. In this paper, I explore the possibility to use data extracted from Dutch Bol.com customer reviews rather than annotated data, since annotating data is a labour-intensive task.

I try two different methods: in the first one, I fine-tune a Dutch GPT-2 model to generate aspects together with a sentiment label. In the second, I fine-tune the same GPT-2 model to generate only an aspect. I then fine-tune a Dutch BERT model (BERTje) to classify the sentiment. Using both an automatic and human evaluation, I find that the combination of GPT-2 and BERT outperforms the multi-tasked GPT-2.

Additional Key Words and Phrases: aspect-based sentiment analysis, ABSA, Dutch, GPT-2, BERT, BERTje, customer reviews, aspect generation, aspect sentiment classification, fastText word embeddings

1 INTRODUCTION

Natural language processing is increasingly more important in current society. The internet is overflowing with data, more than a human could ever process. A large amount of the retail industry has moved partly or entirely online, allowing consumers to widely share their experiences with places and products. For businesses, this is very valuable information, but to have all these reviews analysed by hand would take an immense amount of time. Therefore, teaching a computer to read and summarize a set of reviews can help businesses review their customer's opinions much faster. Likewise, it can help customers get an overview of a product's reviews without having to read them all.

Sentiment analysis, also called opinion mining, aims to identify the sentiment of a text. It can classify text as positive or negative, or rank it on a specified scale. Aspect-based sentiment analysis (ABSA) effectively does the same thing, only for much smaller pieces of text. It aims to find the sentiment of different aspects of the reviewed entity. For example, consider the sentence *"The chair is very comfortable, but I don't like the colour."* The overall sentiment of this sentence could be viewed as neutral, or perhaps conflicting since it holds both a negative and positive element. This information is not very specific or informative. However, when looking at the different aspects mentioned in the sentence, the information becomes more distinct. The comfort of the chair is positive, but the colour is not. This information allows a business to review whether they should update their product to attract more customers, and if so, in what way. Furthermore, it can summarize the reviews to help other customers choose what to buy.

TScIT 37, July 8, 2022, Enschede, The Netherlands
© 2022 University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In contrast to most previous work, this research focuses on ABSA in Dutch. Most research in the field of natural language processing is done in English. I do not use an annotated dataset, but rather a large number of unannotated customer reviews. To the best of my knowledge, this has not been tried for ABSA before. The results of this research are evaluated by both an automatic and human evaluation. Since these evaluations use the same data, I compare them to see to what extent they agree with each other.

An issue of aspect-based sentiment analysis is the need to have a large amount of annotated data. Annotating data by hand is very time-consuming and presents the issue that different people annotate in different ways. This causes inconsistencies in the data and can therefore be a problem in training a classifier. The Dutch language has the disadvantage of having a relatively small amount of speakers, compared to e.g. English. This means there is less research into natural language processing tasks in Dutch, therefore not many annotated datasets are available. It also means that it is harder to find annotators via crowdsourcing, thus it is harder to create new datasets. I try to bypass this issue by using customer reviews from Bol.com[1]. Bol.com asks customers to add plus and minus points when writing a product review. I regard these plus and minus points as aspects with a sentiment label, which means that I can use a large amount of unannotated data as if it were annotated.

ABSA generally consists of two tasks: aspect extraction and aspect sentiment classification. Aspect extraction is the task of identifying mentioned aspects of an entity. In the example above, this entity is the chair. For that sentence, aspect extraction would aim to find the words 'comfortable' (or more general: comfort), and 'colour'. Aspect sentiment classification is the task of determining the sentiment of these aspects. For the chair review, these would be positive and negative, respectively. These tasks are typically performed separately. However, some studies on the English, Chinese and Vietnamese languages found that combining those tasks into one, creating a multi-task learning model, can produce promising results [11, 15, 16]. Therefore, this research explores two methods: using a multi-task model and performing the tasks separately. By doing so, I can try to find which method works best when using a customer review dataset.

For the first of the two methods, I fine-tune a Dutch GPT-2 model[3] to generate an aspect when prompted with a review text. In this model's training data, each aspect is preceded by a tag with its sentiment. Consequently, the model generates aspects with a sentiment label. The second method starts with a similar fine-tuned GPT-2 model. However, this model's training data does not contain sentiment tags, thus it only generates aspects. I fine-tune BERTje[4] to determine the aspect's sentiment. The predictions of both methods are evaluated using both an automatic and human evaluation.

[1] Bol.com is a Dutch webshop and marketplace

1.1 Research question

This paper aims to answer the following question:

- How and to what extent can customer reviews that contain plus and minus points be used as training data for an aspect-based sentiment analysis model?

Considering the intended methods, the following questions arise:

- What criteria should be used to select the data?
- How can GPT-2 be fine-tuned to generate relevant aspects?
- How does a fine-tuned GPT-2's performance change when bundling aspect generation and sentiment classification into one task?
- How can BERT be fine-tuned for aspect sentiment classification of generated aspects?

2 RELATED WORK

This research uses a fine-tuned Dutch GPT-2 model as well as a fine-tuned Dutch BERT model. The following section provides some background information about the models used and their origins. Furthermore, it provides some background information about aspect-based sentiment analysis in relation to Dutch, GPT-2 and BERT.

2.1 GPT-2

GPT-2 is short for the second generation Generative Pre-trained Transformer. At the time of writing, a third generation exists (GPT3), but is unfortunately not open-source. GPT-2 was developed by OpenAI[2] in 2019 [13]. It was trained on a large amount of unannotated data, namely internet pages. GPT-2 learns and generates language based on the context of previous words. It can be fine-tuned for a specific task with additional annotated data. Some of GPT-2's abilities are summarizing, translating and question answering. It completes these tasks using text generation; given a prompt, GPT-2 will attempt to continue the text to the best of its ability.

In 2020, the Dutch university of Groningen created a Dutch version of GPT-2. They did so by taking the English GPT-2 model from OpenAI and adapting it to be used for Dutch [3]. Like the original GPT-2 model, the Dutch model is pre-trained and can be fine-tuned with annotated data to perform a specific task. The exact model I use for this paper is called 'GPT-2-small-dutch', it is available on the Hugging Face platform[3]

2.2 BERT

BERT was developed by Google AI Language[4] in 2018 [5]. It is short for Bidirectional Encoder Representations from Transformers. Similar to GPT-2, BERT was trained on a large amount of unannotated data. This data was collected from free English novel books and English Wikipedia pages. Unlike GPT-2, BERT learns language and makes predictions based on all its surrounding context at once, rather than only that of previous words (hence bidirectional). Some language tasks for which BERT achieves a good performance are question answering and next sentence prediction. Furthermore, BERT achieves high scores for sentiment classification tasks [6, 10, 14]. In 2019, 'BERTje' was developed by the Dutch university of Groningen [4]. It was trained the same way as the earlier mentioned BERT, but with a Dutch dataset. It can therefore be used or fine-tuned for Dutch language tasks. The training data mostly originates from books, news articles and Wikipedia. BERTje is available on the Hugging Face platform[5].

2.3 ABSA for Dutch

Little research has been done for Dutch aspect-based sentiment analysis. [1] presents the first full pipeline for Dutch customer reviews. This research used two Dutch annotated datasets, one on restaurants and one on smartphones. Their pipeline consists of three subtasks: aspect term extraction, aspect category classification and polarity classification. These subtasks are respectively performed by using a lexicon lookup on the surface forms and lemmas, gathering semantic information and looking at lexical features and word shapes.

Another Dutch research in ABSA is [2]. This research focuses on creating a pipeline for reviews in retail, banking and human resources. The study distinguishes itself by using service-oriented domains, rather than products. It performs a quality evaluation on models either trained in one domain or all domains. The pipeline used consists of the same steps as in [1]. Aspect term extraction is approached as a sequential IOB labelling task, category classification and polarity classification are both done using an SVM.

2.4 ABSA with GPT-2

[9] reformulates aspect-based sentiment analysis as a generation task, and uses GPT-2 to achieve it. It researches the performance of this method on the sub-tasks separately, but also as a multi-task model. It finds that ABSA as a generation task outperforms BERT on single-task polarity prediction. Furthermore, it finds that fine-tuning GPT-2 as a multi-task model improves the prediction of aspect terms and categories. The main difference with my research is that I use a GPT-2 model that was repurposed for Dutch. Therefore it is uncertain whether my model will achieve similar performance. Furthermore, this research uses an annotated dataset to fine-tune GPT-2, whereas my research uses unannotated customer reviews.

2.5 ABSA with BERT

[8] aims to accomplish both in-domain and out-of-domain aspect-based sentiment analysis and outperforms its predecessors in this task. It does so by fine-tuning BERT as an aspect classifier, a sentiment classifier and a combined model that classifies aspect and sentiment simultaneously. In this research, all aspects are classified into a predefined category. This is a significant difference compared to my research, since I do not use predefined aspect categories.

Another related research is [14]. It investigates the possibility of using BERT for aspect sentiment classification by creating auxiliary sentences. In this way, it changes the task to a sentence-pair classification task. Inspired by BERT's high-quality performance on question answering tasks and natural language inference tasks, the paper proposes four formats for those auxiliary sentences. Two of those are questions, the other two are statements. This is similar to my research in the sense that it tackles the aspect sentiment

[2]OpenAI: an AI research and deployment company
[3]https://huggingface.co/GroNLP/gpt2-small-dutch
[4]Google AI: Google's artificial intelligence research and development division
[5]https://huggingface.co/GroNLP/bert-base-dutch-cased

classification task by treating it as a sentence-pair classification task. However, my second sentence contains only the aspect, rather than a question or statement.

3 DATASET AND PREPROCESSING

The dataset consists of customer reviews about electrical appliances, retrieved from Bol.com. This website is a Dutch webshop and marketplace. With 13 million active customers and 41 million available products, Bol.com holds an enormous number of customer reviews. The domain of electrical appliances is used for this research because these products generally have a large variety of aspects (e.g. battery, waterproof, sound, etc).

3.1 Data format

Each review contains a text and a list of positive and negative aspects (hereafter referred to as 'pros' and 'cons'). Since the customer does not necessarily need to write pros or cons along with the review, the presence and quantity of both vary. Another thing to note is that customers use the ability to add pros and cons to their reviews in different ways. Some try to summarize what they wrote in the review text, others elaborate on the text. Frequently, the pros and cons form a combination of these two options: some aspects are explicitly mentioned in the review text, and some aspects provide new information. For example, consider the following review:

This laptop is super fast, but it overheats quickly.
+ good quality *- expensive*
- overheats quickly

This review contains a text and three aspects: one pro and two cons. The aspect 'overheats quickly' is clearly mentioned in the text. However, the aspects 'good quality' and 'expensive' provide new information.

3.2 Data preprocessing

In order to use this data to fine-tune a language model, I first have to make sure it is useful and relevant. I start by removing all the duplicates from the downloaded data. Bol.com shows the same reviews for slightly different products (e.g. different colours), causing a large portion of the reviews to occur more than once.

Next, I try to filter out all aspects that were not mentioned in the review text. My goal is to provide GPT-2 with a dataset containing only explicitly mentioned aspects, so that it can learn how to generate those itself. To achieve this goal, I use the pre-trained fastText Dutch embeddings [7]. I take the average vector of the words in an aspect and calculate the cosine similarity between this vector and the word vector of each word in the review text. I save the highest similarity alongside the aspect. A high cosine similarity indicates a high semantic similarity. It ranges from 0, for completely different semantics, to 1, for the exact same embeddings. If this similarity is high enough, I assume the aspect was mentioned in the review text. An example of this process is presented in Table 1.

By visual inspection of the results, I can find a threshold for the similarity value. This threshold should be high enough to filter all irrelevant aspects, but low enough to include sufficient relevant aspects. The threshold I find is 0.83, so I remove all reviews that have no aspects with a similarity score above 0.83. Unfortunately, using this threshold means filtering out many relevant aspects. For example, consider the example in Table 1. The aspect 'overheats quickly' is discarded, even though it is explicitly mentioned in the text. However, using a lower threshold would include many irrelevant aspects. Since enough data is available, I favour excluding all irrelevant aspects over including all relevant aspects in the choice of threshold.

Table 1. Example of data preprocessing

	good quality *(goede kwaliteit)*	expensive *(duur)*	overheats quickly *(oververhit snel)*
This *(Deze)*	0.492	0.301	0.369
laptop *(laptop)*	0.219	0.171	0.293
...	...	...	...
overheats *(oververhit)*	0.149	0.214	0.813
quickly *(snel)*	0.363	0.352	0.813
Highest similarity	0.492	0.352	0.813

The final step to preprocess the data is to split it into a training, validation and testing set. The training and validation sets are used to fine-tune the language models. The testing set can then be used to evaluate the models with new, unseen data. I take 1,000 reviews and set them aside as the unseen testing set. For the rest of the reviews, I first remove all aspects with a similarity score below 0.83. Since I already removed all reviews without any aspects above this threshold, all reviews in this dataset should have at least one aspect left. I do not remove any aspects from the testing set because I want to keep those for the evaluation. I split the remaining reviews into a training and validation set by randomly selecting 15% of the reviews as validation data. An overview of the amount of data along each step of preprocessing can be found in Table 2.

4 METHOD

I attempt to answer the research question by using two different methods. Both methods cover the aspect extraction and aspect sentiment classification tasks, but they do so in different ways. The first method generates aspects with a sentiment label using a fine-tuned GPT-2 model. In other words, it tackles both tasks in one go. For the second method, I fine-tune GPT-2 to generate aspects and fine-tune BERTje to label their sentiment. The following section goes into more detail on the two methods. The code and data used to fine-tune the models can be found on GitHub[6].

4.1 Method 1: Multi-task GPT-2

Method one uses the Dutch GPT-2 model (see Section 2.1) for both aspect extraction and classification. I fine-tune the model using 3 epochs and a learning rate of $5{\times}10^{-5}$, with the training and validation datasets described in Section 3.2. Each review is formatted in the following way[7]:

<startoftext> This laptop has a great screen, but a terrible keyboard. <aspects> <pro> screen <con> keyboard <endoftext>

[6] https://github.com/FM12001/DutchABSA.git

Table 2. Numeric information on the dataset during preprocessing

	Number of reviews	Number of aspects	Number of aspects per review			
			0	1	2	3+
Total downloaded	686,278	1,605,448	85,544	144,083	136,375	320,276
After removing duplicates	298,109	687,528	44,354	60,653	59,520	133,582
After removing aspects with a similarity value below 0.83	29,848	36,291	0	25,556	3,124	1,168
In the training/validation data	28,848	32,694	0	25,484	2,936	428
In the testing data	1,000	3,597	0	72	188	740

After training the model, I prompt it to generate aspects by formatting the testing data as follows:
<startoftext> This phone has a good battery. <aspects>
Ideally, when feeding this to the model, it outputs something similar to:
<startoftext> This phone has a good battery. <aspects> <pro> battery <endoftext>
From this output, I can extract the generated aspects together with their sentiment label.

4.2 Method 2: GPT-2 and BERT

For the second method, the aspect extraction using GPT-2 happens quite similar to the first method. The only difference is that instead of specifying aspects as either 'pro' or 'con', they are all labelled as 'aspect'. This means the training data is formatted as follows[7]:
<startoftext> This laptop has a great screen, but a terrible keyboard. <aspects> <aspect> screen <aspect> keyboard <endoftext>
I fine-tune the model using the same data and parameters as for the first method (3 epochs, $5{\times}10^{-5}$ learning rate). After fine-tuning, the model can be prompted the same way as in method 1. However, the ideal output now looks like this:
<startoftext> This phone has a good battery. <aspects> <aspect> battery <endoftext>
From this output, I can extract the aspect 'battery'. However, unlike in method 1, it does not have a sentiment label yet. To create this label, I fine-tune the pre-trained BERTje model (see Section 2.2). The training data for BERTje contains the same pros and cons as used for the GPT-2 fine-tuning, each linked to their corresponding review. They are also linked to a label: 1 for a positive aspect, or 0 for a negative aspect. The result has the following format:
("This laptop has a great screen, but a terrible keyboard.", "screen")
Label: 1
("This laptop has a great screen, but a terrible keyboard.", "keyboard")
Label: 0

The model is again fine-tuned using 3 epochs and a learning rate of $5{\times}10^{-5}$. An important detail is the maximum input size of a BERT model, which is 512 tokens. If the testing data is longer than this, only the first part of the data may be truncated. Losing part of the review text is not necessarily problematic, but losing the aspect is. After fine-tuning BERTje on this data, it can be prompted with an input like:
("This phone has a good battery.", "battery")
Ideally, it should now output the label 1, to indicate the aspect in this input is positive according to the review.

5 EVALUATION AND RESULTS

Ideally, I would like to evaluate all testing data with high quality. Realistically, this is not possible, since it would take too much time and evaluators. Therefore, the two methods are evaluated by both an automatic and human evaluation. The automatic evaluation can assess a large number of results, but the quality is limited. In contrast, the human evaluation can only assess a smaller part of the results, but with higher quality. The following section presents both the evaluation processes, their limitations and the gathered results.

5.1 Automatic Evaluation: Aspect Extraction

The aspect extraction task evaluation is similar to the selection of the training data, namely by using word embeddings and their cosine similarity. I ignore the sentiment labels for now, which means I compare each generated aspect as if its sentiment is unknown. To evaluate the aspect generation, I compose two different scores for each generated aspect: the customer aspect similarity (*cas*) and the review text similarity (*rts*).

The customer aspect similarity represents the highest similarity found between the generated aspect and the aspects written by the customer. In other words, I try to find out whether the generated aspect was also mentioned by the customer as a pro or con. As mentioned in Section 3.2, the testing dataset contains all the customer aspects, including the ones with a similarity below the threshold. In this way, if a relevant customer aspect fell below the threshold, I can still use it to see if a generated aspect is relevant. For example, this could be the case for the aspect 'overheats quickly' in Table 1, which is a relevant aspect but fell below the threshold. I calculate the cas score by calculating the cosine similarity of the generated aspect vectors and the customer aspect vectors. I then save the highest value for each generated aspect. By visual inspection I find a threshold of 0.80. This means that I assume all generated aspects with a cas score above 0.80 are mentioned in the review, thus I consider them relevant aspects. However, it does not necessarily mean that all generated aspects with a cas score below the threshold are invalid aspects. Consider the following examples:

[7] For the purpose of this paper, English examples are used. The actual models are trained on Dutch data.

Table 3. Automatic evaluation: aspect generation results

	GPT-2 and BERT	Multi-task GPT-2
Total number of generated aspects	1000	1046
Threshold rts score	0.83	0.83
Threshold cas score	0.80	0.80
Mean average of rts score	0.702	0.694
Standard deviation of rts score	0.181	0.193
Mean average of cas score	0.695	0.645
Standard deviation of cas score	0.264	0.251
Number of aspects that are …		
above both thresholds	361	289
only above rts threshold	132	163
only above cas threshold	54	56
above either threshold	547	508
Minimal accuracy	0.547	0.485

cosine_similarity("past precies", "past prima") = 0.801
(fits perfectly), (fits fine)
cosine_similarity("werkt goed", "werkt naar behoren") = 0.759
(works well), (works properly)

Both these examples hold two differently expressed aspects. The aspects are the same, but the expressions differ in adverbs and adjectives. Whether something fits 'fine' or 'perfectly', the aspect remains about the size. However, only the first example has a cas score above 0.80. That means that the first example is a relevant aspect. The second example's cas score falls below the threshold, but it is a relevant aspect too. This means I cannot know for sure whether it is relevant when evaluating it automatically.

The review text similarity score represents the highest similarity between the generated aspect and a word in the review text. With this score, I aim to find out whether the generated aspect was mentioned in the review text. The process of finding this score is the same as the data preprocessing in Section 3.2. See Table 1 for an example. Apart from the process, the goal is the same too: I want to know if the aspect is mentioned in the review. Therefore, it makes sense to use the same threshold of 0.83. I consider any generated aspect with an rts score above this threshold

After calculating the scores for each generated aspect, I count how many aspects I can assume to be relevant. Next, I use this number to calculate the accuracy of the aspect generation of each model. I call this accuracy the 'minimal accuracy', since any aspects with scores below the thresholds are not necessarily invalid. The minimal accuracy therefore holds the highest accuracy that I can be certain of. The results can be found in Table 3.

5.2 Automatic Evaluation: Aspect Sentiment Classification

Evaluating the aspect sentiment classification is slightly more challenging than evaluating the aspect generation. This is mostly because the results are dependent on whether or not the aspects were generated correctly. If the review is about a smartwatch, it is hard to determine whether the aspect 'remote control' was assigned the right sentiment. An additional thing to note is that the customer aspects do not necessarily cover everything mentioned in the review text. For example, consider the aspect 'fast' from the review in Section 3.1). I need to know the actual sentiment of a generated aspect to check if the prediction is correct, but I can only find that sentiment if the same aspect was mentioned in the customer aspects. Therefore, I limit the evaluation to the aspects that have a customer aspect similarity (*cas*) above the threshold of 0.80 (see Section 5.1). This leaves 415 generated aspects for the GPT-2 and BERT method, and 345 generated aspects for the multi-task GPT-2 method.

Table 4. Automatic evaluation: aspect sentiment classification results

	GPT-2 and BERT	Multi-task GPT-2
True negatives	18	27
True positives	387	305
False negatives	1	6
False positives	9	7
Total	415	345
Accuracy	0.976	0.962
Precision	0.977	0.978
Recall	0.997	0.981
F1	0.987	0.979

For the multi-task GPT2 method, the aspects are already linked to a sentiment label. For the GPT-2 and BERT method, I first need to let my fine-tuned BERTje model predict the sentiment of the generated aspects. After this, I have the predicted sentiment label for the generated aspects of both models. However, I do not yet know the actual sentiment of these aspects. In order to find this, I compare the generated aspect vector to each pro vector and save the highest cosine similarity. Next, I repeat this for each con vector. Since I only evaluate aspects with a cas score higher than 0.80, either the highest pro similarity or the highest con similarity has to be above 0.80, if not both. I assume that the highest similarity I find is from the matching customer aspect. Therefore, if the highest similarity is with a pro, I assume the actual sentiment of the generated aspect is positive, and vice versa. Now that I have both the predicted sentiment label and the actual sentiment label of each generated aspect, I can easily count the number of true and false negatives and positives. With these numbers, I calculate the accuracy, precision, recall and F1 score of both models. The results can be found in Table 4.

5.3 Human Evaluation

Using word vectors is a great way to gain a large number of results. However, the quality of those results is limited. To get an indication of said quality, and to get a high-quality assessment for both the aspect extraction and aspect sentiment classification tasks, I also perform a human evaluation. This evaluation consists of an online survey. The questions have the following format[8]:

What can you say about [aspect] in this review?
[Review text]
- [aspect] is not mentioned
- [aspect] is positive
- [aspect] is negative

This question forms two questions in one. It asks the respondent

[8]For the purpose of this paper, the example is in English. The actual survey is in Dutch

Table 5. Human evaluation: aspect extraction results

	GPT-2 and BERT	Multi-task GPT-2
Total number of generated aspects	100	100
Number of aspects without split division	95	98
Number of relevant aspects	60	54
Accuracy	0.632	0.551

whether the aspect is mentioned, and if it is, whether it is positive or negative. For this reason, the results are presented in two parts as well.

The survey consists of 100 reviews. Each review is annotated for two aspects (one from each model), meaning the survey contains 200 questions. The respondent does not see the two questions for the one review at the same time, they are annotated separately. The survey reached 82 respondents, which led to a total amount of 1210 annotations, 605 for each model. Each review is annotated by at least three respondents. On average, the reviews received 6 annotations.

The final answer to each question was found by taking the majority vote. A generated aspect is assumed to be irrelevant if more than half of the responses said it was not mentioned in the review text. If the aspect is assumed relevant, its sentiment is again found by taking the majority vote. Any aspects with a split division of answers are left out of the results (this is the case for 8 aspects).

Since the human evaluation is done with a subset of the testing data of the automatic evaluation, it is interesting to see whether the evaluations agree with each other on the generated aspects. This is shown and discussed in Appendix A.

5.3.1 Human Evaluation: Aspect Extraction. The results can be found in Table 5. The Multi-task GPT-2 method scored an accuracy of 0.551. This is 0.08 lower than the accuracy of the GPT-2 and BERT method, which is 0.632.

To see whether the difference in accuracy is enough to say the models differ, I use an approximate randomization test [12], a non-parametric test which requires minimal assumptions. This test is based on the idea that if one model scores higher on a test metric t (in my case, the accuracy), swapping random predictions of the two models should almost never result in a better performance of that model. If this does happen often, then either the models are not that different, or not enough predictions are available. I use this test by taking the 100 reviews as the independent variable, and the accuracy as the test metric (i.e. the statistic). After 10,000 iterations, the test results in a 2-tailed p-value of 0.393. This value is higher than the standard alpha value of 0.05 (0.025 for a 2-tailed test), which means I cannot say for certain that the two models differ in performance.

5.3.2 Human evaluation: Aspect Sentiment Classification. The results of the human evaluation for the sentiment classification can be found in Table 6. Both methods achieved approximately the same F1-score. The GPT-2 and Bert method got an F1-score of 0.945. The Multi-task GPT-2 method got an F1-score of 0.943.

Since the difference in results is very small for this task, the approximate randomization test is not used.

Table 6. Human evaluation: aspect sentiment classification results

	GPT2 and BERT	Multi-task GPT2
True negatives	5	9
True positives	52	41
False negatives	0	0
False positives	6	5
Total	63	55
Accuracy	0.905	0.909
Precision	0.897	0.891
Recall	1.0	1.0
F1	0.945	0.943

6 DISCUSSION AND FUTURE WORK

For the automatic evaluation of the aspect extraction task, the GPT-2 and BERT method scored a minimal accuracy of 0.55. The multi-task GPT-2 model scored a minimal accuracy of 0.49. For the human evaluation, the methods scored an accuracy of 0.63 and 0.55, respectively. The GPT-2 and BERT method appears to have the best performance for this task. As expected, the automatic evaluation is too strict in declaring aspects irrelevant, since both methods score higher on the human evaluation. The approximate randomization test failed to prove that the difference in accuracy for the human evaluation is significant. However, since the automatic evaluation shows a similar difference in performance, I assume that GPT-2 and BERT still outperforms the multi-task GPT-2.

Unfortunately, I have no automatic way to find the precise accuracy of the results of the methods. I call the accuracy that I did find the 'minimum accuracy' because my evaluation method only finds which aspects are probably relevant (due to the chosen high threshold). It does not tell me which aspects are probably irrelevant. Instead, for all aspects that fall below the threshold, I simply do not know much about their relevance. I assume the worst-case scenario and count them all as invalid. Consequently, there is an error margin from the calculated accuracy going upwards until 1. However, the advantage of choosing a relatively high threshold is that I can be certain the models perform with at least the found accuracy.

For the aspect sentiment classification task, both methods score similarly and high. GPT-2 and BERT slightly outperforms multi-tasked GPT-2, with the respective F1-scores 0.99 and 0.98 for the automatic evaluation. The human evaluation scores are slightly lower, namely 0.95 and 0.94. It is interesting to note that both methods achieve F1 scores very close to 1, meaning they labelled all sentiments almost perfectly. However, that does bring into question the validity of the evaluation. Since I can only evaluate generated aspects that are mentioned in the review according to the extraction evaluation, I can only evaluate a relatively small part of the testing data. This is the part that was mentioned explicitly in the review, which could mean it is also the easier part to find the sentiment of.

Overall, the GPT-2 and BERT method outperforms the multi-task GPT-2 method. This could be because this method uses two separate models for the two separate tasks. In other words, the model is dedicated entirely to generating aspects. The multi-task GPT-2 model has to divide its capabilities over two tasks. Hence, it makes sense that it performs a little less well.

The methods score well on aspect sentiment classification, however, their scores on aspect extraction could be improved. I think using Bol.com reviews rather than annotated data has the potential to work very well, but it would be interesting to look into different ways of doing so. Perhaps a better method to extract the aspects from the review texts can be found. Furthermore, it can be useful to design a better way to preprocess the data so that more relevant aspects can stay with the reviews. This way the reviews would provide more data to train on. Lastly, the automatic evaluation method could be more precise, especially for the aspect extraction task (see Appendix A). It could be practical to design a better automatic evaluation method in the future, to properly compare the performances of various methods.

7 CONCLUSION

In this paper, I explore the possibility of fine-tuning language models for aspect-based sentiment analysis without using annotated data. Instead, I use Bol.com customer reviews with plus and minus points, which I regard as aspects with a sentiment label. I try two methods, namely a multi-task GPT-2 and a combination of GPT-2 (for aspect extraction) and BERT (for aspect sentiment classification). I use the pre-trained fastText Dutch embeddings to preprocess the data and automatically evaluate the results. Furthermore, I use an online survey to have a subset of the results evaluated by humans. From both evaluations, I find that the GPT-2 and BERT method outperforms the multi-task GPT-2 method. Both models perform very reliably on the task of aspect sentiment classification. However, the accuracy scores on the aspect extraction tasks can still be improved. It would be interesting to find a different way to extract the aspects, and then use BERT to label their sentiment. Overall, using customer reviews with plus and minus points rather than annotated data has the potential for success. The methods I propose could be improved for the task of aspect extraction, but the scores are high enough to show that it is possible to train for this task using Bol.com data. Furthermore, both methods show that customer review data can successfully be used to fine-tune a language model for an aspect sentiment classification task.

REFERENCES

[1] Orphée De Clercq and Véronique Hoste. 2016. Rude waiter but mouthwatering pastries! An exploratory study into Dutch Aspect-Based Sentiment Analysis. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Portorož, Slovenia, 2910–2917. https://aclanthology.org/L16-1465

[2] Orphée De Clercq, Els Lefever, Gilles Jacobs, Tijl Carpels, and Véronique Hoste. 2017. Towards an integrated pipeline for aspect-based sentiment analysis in various domains. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, Copenhagen, Denmark, 136–142. https://doi.org/10.18653/v1/W17-5218

[3] Wietse de Vries and Malvina Nissim. 2021. As Good as New. How to Successfully Recycle English GPT-2 to Make Models for Other Languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.74

[4] Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *CoRR* abs/1912.09582 (2019). http://arxiv.org/abs/1912.09582

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/ARXIV.1810.04805

[6] Zhengjie Gao, Ao Feng, Xinyu Song, and Xi Wu. 2019. Target-Dependent Sentiment Classification With BERT. *IEEE Access* 7 (2019), 154290–154299. https://doi.org/10.1109/ACCESS.2019.2946594

[7] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. https://doi.org/10.48550/ARXIV.1802.06893

[8] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-Based Sentiment Analysis using BERT. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*. Linköping University Electronic Press, Turku, Finland, 187–196. https://aclanthology.org/W19-6120

[9] Ehsan Hosseini-Asl, Wenhao Liu, and Caiming Xiong. 2022. A Generative Language Model for Few-shot Aspect-Based Sentiment Analysis. https://arxiv.org/abs/2204.05356

[10] Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained Sentiment Classification using BERT. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, Vol. 1. 1–5. https://doi.org/10.1109/AITB48515.2019.8947435

[11] Hy Nguyen and Kiyoaki Shirai. 2018. A Joint Model of Term Extraction and Polarity Classification for Aspect-based Sentiment Analysis. In *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*. 323–328. https://doi.org/10.1109/KSE.2018.8573340

[12] Sebastian Padó. 2006. User's guide to sigf: Significance testing by approximate randomisation. https://nlpado.de/~sebastian/software/sigf.shtml

[13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. http://www.persagen.com/files/misc/radford2019language.pdf

[14] Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. *CoRR* abs/1903.09588 (2019). http://arxiv.org/abs/1903.09588

[15] Dang Van Thin, Duc-Vu Nguyen, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen, and Anh Hoang-Tu Nguyen. 2020. Multi-task Learning for Aspect and Polarity Recognition on Vietnamese Datasets. In *Computational Linguistics*, Xuan-Hieu, Hasida Kôiti, Tojo Satoshi Nguyen Le-Minh, and Phan (Eds.). Springer Singapore, 169–180.

[16] Heng Yang, Biqing Zeng, Jianhao Yang, Youwei Song, and Ruyang Xu. 2021. A multi-task learning model for Chinese- oriented aspect polarity classification and aspect term extraction. *Neurocomputing* 419 (2021), 344–356. https://doi.org/10.1016/j.neucom.2020.08.001

Table 7. Comparing the evaluation methods

AE	Auto +	Auto -	Total
Human +	84	30	114
Human -	18	68	86
Total	102	98	200
ASC	Auto +	Auto -	Total
Human +	88	5	93
Human -	6	19	25
Total	94	24	118

A APPENDIX A

Since the human evaluation is done using a subset of the reviews from the automatic evaluation, it is possible to see to what extent the two evaluation methods agree with each other. I created two confusion matrices: one for each subtask. The evaluation results of both the Multi-task GPT-2 and GPT-2 and BERT method are added together. The matrices are presented in Table 7.

For the task of aspect extraction (AE), the agreement for the two evaluations has an F1-score of 0.778. This is high enough to assume that the automatic evaluation results give a useful insight into the performance of the models. However, the score is not high enough to render the human evaluation unnecessary.

For the aspect sentiment classification (ASC) task, the evaluation methods seem to almost always agree with each other. They got an F1-score of 0.941. This score is high enough to consider leaving

out the human evaluation for this task in a future repetition of the experiment.