Products opinions and news
Project report for NLP Course, Winter 2023

Anish Gupta
Warsaw University of Technology
01175535@pw.edu.pl

Martyna Majchrzak
Warsaw University of Technology
martyna.majchrzak.stud@pw.edu.pl

Bartosz Rożek
Warsaw University of Technology
01142140@pw.edu.pl

Konrad Welkier
Warsaw University of Technology
01144707@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

Training and evaluating models for aspect-based sentiment analysis is a task that requires a specific type of data set, including aspects of a given text and sentiment towards this aspect. While there are several data sets in English available for this task, there is a lack of such a data set in the Polish language. In this work, we test and compare the performance of several state-of-the-art tools for sentiment analysis and aspect-based sentiment analysis on English data sets. Based on their results, we choose the most efficient one (PyABSA) and use it on an existing Polish data set suitable for sentiment analysis to create a new version of it with aspects and sentiments included. This new data set can be used by researchers to train and evaluate models for aspect-based sentiment analysis in Polish.

1 Introduction

Sentiment Analysis, a pivotal aspect of Natural Language Processing (NLP), involves computationally identifying and categorizing opinions expressed in a piece of text to determine the writer's attitude towards a particular topic, product, or service. This analysis is typically binary, classifying sentiments as positive or negative, though it can also cover a range of emotions like happiness, anger, or sadness. The significance of sentiment analysis lies in its ability to parse through vast amounts of data - from product reviews to social media content - enabling businesses and researchers to gauge public opinion, conduct market research, and enhance customer service.

Building upon the foundation of traditional sentiment analysis, aspect-based sentiment analysis (ABSA) delves deeper by not only discerning the sentiment but also linking it to specific aspects or attributes of a product or service. For instance, in a restaurant review, ABSA helps in distinguishing the customer's sentiment about various aspects such as food quality, service, ambience, and price. This granular approach provides a more detailed sentiment overview, crucial for businesses and service providers aiming to pinpoint strengths and areas of improvement based on customer feedback.

ABSA can be approached typically through two methodologies. The first is the two-step approach, consisting of an Aspect Extractor followed by a Sentiment Classifier. In this method, the system first identifies the relevant aspects mentioned in the text and then determines the sentiment expressed for each aspect. This sequential processing, while methodical, can sometimes lead to compounded errors where inaccuracies in the first step (aspect extraction) adversely affect the second step (sentiment classification). An alternative is the one-step approach, where aspect and sentiment analysis are conducted simultaneously. This integrated method often leverages advanced machine learning and deep learning models to capture the nuances of both aspects and their associated sentiments in one go. The one-step approach can be more efficient and might reduce the error propagation seen in the two-step process. However, it requires robust model architectures capable of un-

derstanding complex relationships in the text.

The project's goal centres around evaluating sentiment analysis within product-related news articles, encompassing sentiments at both the comprehensive text level and their components. The focus lies on delineating sentiment, particularly in user opinions and news texts, through an aspect-based sentiment analysis approach.

The significance of this project is underscored by the existing void in the realm of Polish language data sets specifically tailored for aspect-based analysis in sentiment evaluation. Presently, there are available English data sets like Amazon Reviews (McAuley et al., 2015) and SemEval-2014 (Pontiki et al., 2014), which cater to aspect-based analysis, and one Polish data set, PolEmo (Kocoń, Zaśko-Zielińska and Miłkowski, 2019), primarily designed for general sentiment analysis. However, there is a noticeable absence of a dedicated Polish data set geared towards aspect-based sentiment analysis. The creation of such a data set constitutes a pivotal contribution, as it fills a crucial gap in the domain. This endeavour will pave the way for researchers, developers, and practitioners in the field to explore and advance their methodologies in aspect-based sentiment analysis within the Polish language.

This report contains a detailed description of the project conducted as part of the NLP Course at the Faculty of Mathematics and Information Science at Warsaw University of Technology. Section 2 contains a literature review, as well as information about state-of-the-art tools and data sets that were used. In section 3 we describe the motivation for our approach, the structure of a Python package created as part of the project and metrics used to evaluate the results. The experiments and their results are presented in section 4. Section 5 contains the discussion of the results and section 6 - conclusion and suggestions for further work.

2 Related work

Sentiment analysis, a fundamental aspect of natural language processing, has garnered significant attention due to its applications across various domains. Traditionally, sentiment analysis primarily focused on determining the overall sentiment polarity of text, classifying it as positive, negative, or neutral. However, the evolving landscape of sentiment analysis has shifted towards a more nuanced approach known as aspect-based sentiment analysis (ABSA).

Studies by (Liu and Zhang, 2012) laid the groundwork for ABSA, introducing the idea of exploring sentiments toward specific aspects or features within a text. This paradigm shift led to the exploration of sentiment analysis beyond document-level polarity, allowing for a more granular understanding of opinions within different aspects or entities mentioned in the text.

Recent studies by (Pontiki et al., 2014) have explored aspect-based sentiment analysis methodologies in English. They introduced techniques like aspect extraction and sentiment classification for fine-grained sentiment analysis, laying the groundwork for subsequent research in this field.

Sentiment analysis is a task highly connected to the language of the given text. This challenge can be addressed by creating multilingual models for such a task, such as a corpus proposed by (Augustyniak et al., 2023).

2.1 State-of-the-art

There are several tools available to perform the sentiment analysis, both overall and aspect-based. Below the summarize some of them, chosen to be evaluated in this study.

2.1.1 SentiStrength

SentiStrength (2010) is a text analysis tool designed specifically for sentiment analysis. It is capable of assigning a sentiment strength score to text, which is useful for detecting positive and negative sentiments in short texts. Created specifically with informal communication found in social network posts, blogs, and discussion forums in mind, SentiStrength employs a sentiment word dictionary along with associated strength measures. It leverages various unconventional spellings and other prevalent textual methods to express sentiments. The development of SentiStrength involved analyzing an initial data set of 2,600 human-classified MySpace comments, followed by evaluation using an additional random sample of 1,041 MySpace comments.

Python 3 Wrapper for SentiStrength is available on the Github repository[1] as a Python package.

This tool is designed for overall Sentiment Analysis and it is not suited for performing ABSA. In this project we use it in two ways: for overall sentiment analysis and as a second-step tool in

[1] https://github.com/zhunhung/Python-SentiStrength

ABSA, where the text is first split into chunks or the aspects of the text are extracted using another tool.

2.1.2 SpaCy

SpaCy (2017) is a free, open-source library for advanced Natural Language Processing (NLP) in Python. It can be used to perform Tokenization, Part of speech tagging, NER and other NLP tasks. It contains trained pipelines in 26 different languages, including English and Polish, and trained pipelines for each of those models. It is designed to be suitable for large-scale problems and extraction tasks, ensuring appropriate speed of processing - both CPU and GPU hardware options are supported.

In this project, we use the English "en_core_web_sm" model as a first-step tool (an extractor) for ABSA. "pl_core_news_sm" is suitable to use for tasks in Polish, but it was not used in project since pipelines with SpaCy as first-step tool did not achieve the best result.

2.1.3 Flair

Flair (2019) is an NLP framework designed to streamline the training and deployment of cutting-edge sequence labeling, text classification, and language models. It was first introduced in (Akbik et al., 2018) as a framework to enable the application of contextual string embeddings for sequence labeling. Its primary aim is to simplify the integration of various word and document embeddings by offering a unified interface. The framework addresses the complexity associated with different types of embeddings, hiding the specific engineering challenges they pose. This abstraction enables researchers to seamlessly combine and utilize diverse embeddings without extensive modifications to model architectures.

Traditional word embeddings, although beneficial for NLP tasks, come with limitations. They require specific adjustments to model architectures, especially when incorporating additional features like subword structures or contextualized embeddings. FLAIR aims to mitigate these challenges by providing a straightforward interface that allows researchers to mix different embeddings effortlessly within a single model architecture.

Moreover, Flair offers functionalities such as data fetching modules for easy access to NLP datasets, simplifying the setup of experiments. It includes standard model training procedures and hyperparameter selection routines, streamlining the training and testing workflows for NLP models. Additionally, Flair comes equipped with a collection of pre-trained models, enabling users to readily apply state-of-the-art NLP models to their applications.

The tool is made available as a Python package on the well-documented Github repository[2].

In this project, Flair will be used for overall Sentiment Analysis and as a second-step tool in ABSA, where the text is first split into chunks or the aspects of the text are extracted using another tool and then those aspects are classified with Flair.

2.1.4 ChatGPT

ChatGPT (2021) is a state-of-the-art tool, that is perfect for processing text data, and it can be leveraged to do multiple things. It allows access to the models from the GPT (Generative Pretrained Transformer) family. The free tier version uses GPT3.5-Turbo and the ChatGPT Plus (paid version) uses GPT-4. It can be accessed as an online service or as a REST API through OpenAI Python API library.

The model is trained to interact conversationally, answering the prompts provided by the user. The dialogue format allows the model to answer follow-up questions, admit its mistakes and reject inappropriate requests. However, the unstructured format of the response poses a problem and requires careful prompt engineering to achieve the response in the desired format and then parsing the said response to a more structured data form.

In this project we use it to perform overall sentiment analysis using one type of prompt and for aspect-based sentiment annotations in two ways:

- to divide sentences into chunks, that are later processed using Flair 2.1.3
- to extract keywords from sentences, which are later processed with SentiStrength 2.1.1.

2.1.5 BERT

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) is a groundbreaking model in the field of natural language processing (NLP). Developed by Google, BERT revolutionized the understanding of how

[2]https://github.com/flairNLP/flair

deep learning could be applied to language understanding. It was a significant shift from the previous models that predominantly focused on unidirectional or sequential processing.

BERT is based on the Transformer architecture, which relies on attention mechanisms to understand the context of a word in relation to all other words in a sentence, unlike traditional models that process words in order. This bidirectional context understanding is a key feature of BERT.

2.1.6 PyABSA

PyABSA (2023) is an open source Framework for reproducible Aspect-based Sentiment Analysis built on PyTorch. It supports several ABSA subtasks, including aspect term extraction, aspect sentiment classification, and end-to-end aspect-based sentiment analysis. It integrates 29 different models and 26 additional data sets, but it also allows for extensions and use of your own models and data sets. The motivation behind it was to create an easy-to-use solution that would allow beginners to reproduce the results of a model with a specific data set. The authors released a range of trained checkpoints, which can be accessed through the Transformers Model Hub (powered by Huggingface Space), for users who need exact reproducibility. The tool is available on the Github repository[3], but it can also be downloaded as a Python package.

In this project, we used PyABSA for Aspect-based Sentiment Analysis in two ways:

- as an end-to-end tool for both aspect term extraction and sentiment classification
- as a first-step tool to extract aspects to be classified using Flair 2.1.3 or SentiStrength 2.1.1.

2.2 Data sets

2.2.1 Amazon data set

The Amazon Reviews data set (McAuley et al., 2015) is a collection of reviews written in English by customers for products purchased on Amazon. It's one of the largest and most commonly used data sets for sentiment analysis and natural language processing tasks. This data set includes reviews spanning various product categories, providing a broad range of vocabulary and topics. We will be using the *Electronics* subset of the Amazon reviews, as it is a very large data set, and the authors themselves encourage us to make use of a subset of it. The preprocessing of this dataset is covered in section 4.1.1.

[3]https://github.com/yangheng95/PyABSA

2.2.2 SemEval data set

The SemEval 2014 Task 4 dataset (Pontiki et al., 2014) is an English benchmark dataset used to evaluate systems for aspect-based sentiment analysis. It can be used for different subtasks described in the original paper:

- Subtask 1: Aspect Term Extraction - Extract the explicit aspect term (e.g., "battery life") from the sentence.
- Subtask 2: Aspect Term Polarity- Determine the sentiment polarity of the aspect term mentioned in a sentence.
- Subtask 3: Aspect Category Detection - Identify the category of the aspect (e.g., "food", "service") that is mentioned in a sentence
- Subtask 4: Aspect Category Polarity - Determine the sentiment polarity for the aspect category.

The Exploratory Data Analysis of this data set in included in section 4.1.2.

2.2.3 Polemo data set

This dataset was introduced in a paper (Kocoń, Zaśko-Zielińska and Miłkowski, 2019). It consists of Polish reviews from four domains: medicine, hotels, products, and schools. Each review is labelled with one of the following Polish labels: "negatywny", "ambiwalnetny", "neutralny", and "pozytywny" (negative, ambivalent, neutral and positive). The preprocessing of this dataset is covered in section 4.1.3.

3 Approach and research methodology

3.1 Motivation of the approach

To cover all of the aspects of the task and simultaneously keep the code as clean and reproducible as possible we decided to encapsulate all of the codes in the Python package called 'pysent' that is located on a different repository - Github link. This repository, as a standard Python package, holds all of the codes that are used in the annotation process, without the code execution which is done in

the notebooks at the main repository. The whole documentation of the package is hosted on the GCP (Google Cloud Platform) bucket -link. What is crucial in this task, is the fact that all of the tools are prepared to be used with both, Polish and English language. Hence, we use results obtained on the English data sets as an estimator of how good is the tool in working with Polish sentences.

3.2 Limitations of chosen tools

While working with the tools described in 2.1 we have encountered some problems that forced us to limit the use of some tools or resign from using them in the final solution.

3.2.1 ChatGPT

According to OpenAI Documentation free-tier rate limit for gpt-3.5-turbo model allows for only 3 RPM (Requests Per Minute), we were not able to process the entire data sets using this tool. We present the results only of it for overall sentiment analysis and even for this task we use a subset of data available. The appropriate functions for ABSA with ChatGPT are available in the 'pysent' package and the computations of their results are one of our ideas for future works.

3.2.2 BERT

Training BERT on a personal computer presents significant challenges due to its considerable computational and memory requirements. BERT, being a large and complex model, demands high-end GPUs and substantial RAM for efficient processing, which exceeds the capabilities of most standard PCs. This is why we could not use BERT for the purpose we intended it for.

3.3 Description of the package

The package is built to provide users with easily accessible already-done annotators, as well as to be easily extendable with newly created annotators by the user. It contains two main classes - *OverallAnnotator* and *AspectAnnotator* which act as wrappers for pipelines in those two tasks.

3.3.1 OverallAnnotator

OverallAnnotator is a class that as an input has a tool class that inherits from *OverallAnnotatorAbstract* (described further). It has methods that allow to generate sentiment annotation for text and test the supplied annotator with given gold standard annotations. Its structure is a bit too complicated since all of the methods could be implemented in the *OverallAnnotatorAbstract* class, but we decided to follow the path of *AspectAnnotator* and to keep it easily extensible.

Overall annotators are classed that inherit from *OverallAnnotatorAbstract* class. The *OverallAnnotatorAbstract* class is an interface that has methods *check_arguments* and *classify*. We implemented three tools that can assign sentiment to the given text:

- *FlairAnnotator* - a class based on the *Flair* Python package
- *SentiAnnotator* - a class based on the *SentiStrength* tool and *sentistrength* Python package which is a CLI wrapper for *SentiStrength*
- *ChatGPTAnnotator* - a class based on the OpenAI Python package that allows querying ChatGPT from Python and prompt engineering done by us.

3.3.2 AspectAnnotator

AspectAnnotator is a class that takes a list of tools (named further "pipeline") that inherit from *AspectExtractor*, *AspectClassifier* or *AspectExtrassifier* classes. Methods implemented allow users to get annotations as well as to test the supplied pipeline. Two pipeline types are possible:

- Extract aspects (tools named **extractors**) and assign sentiment (tools named **classifiers**) to those aspects with context (one element list supplied as a pipeline)
- Do the whole process (tools named **extrassifiers**) in one run (two elements list supplied as a pipeline)

Extractors are classes that inherit from the interface *AspectExtractor* and implement method *extract* which extracts aspects from sentences with context. In the package, there are three extractors implemented:

- *SpacyExtractor* - a class based on the Python package *spacy*, takes out aspects by part of speech in the sentence, the context is taken out as a set number of words surrounding the aspect.
- *PyabsaExtractor* - a class based on the Python package *pyabsa*, the context is taken

out as a set number of words surrounding the aspect.

- *ChatGPTExtractor* - a class based on the Python package *OpenAI* and prompt engineering.

Classifiers are classes that inherit from the interface *AspectClassifier* and implement method *classify* which for the given context and aspect returns sentiment label. In the package, there are three classifiers implemented:

- *FlairClassifier* - a class based on the Python package *flair*, assigns sentiment to the given context, the aspect is taken from the extractor.

- *SentiClassifier* - a class based on the tool *SentiStrength* and Python package *sentistrength*, the aspects is taken from the extractor, but the context is extracted by *SentiStrength* itself.

Originally, *sentistrength* Python wrapper does not provide the user with the option to do an aspect-based analysis which is possible in the SentiStrength tools itself. To cover this, we have decided to add this functionality and contribute to this open-source package. Our pull request is currently opened - github link.

Extrassifiers are classes that inherit from the interface *AspectExtrassifier* and implement method *classify* that extracts aspects and assigns sentiment to them. The name is taken as a combination of two task names - "extract" and "classify". In the package, there are two Extrassifiers implemented:

- *PyabsaExtrassifier* - a class based on the Python package *pyabsa*.

- *ChatGPTExtractor* - a class based on the Python package *OpenAI* and prompt engineering.

To sum up, we have implemented 3 tools for overall sentiment analysis. For aspect-based analysis, we have 3 extractors, 2 classifiers and 2 Extrassifiers which gives us 8 possible tools.

Extensive descriptions of all of the classes can be found in the documentation provided in the previous section.

To keep the appropriate structure, we decided to introduce several data classes placed in the file *data_structures.py*:

- *ExtractedAspect* - class representing the output of extraction aspect tools. Has an aspect keyword and context in text.

- *SentimentAnnotation* - Contains information about single sentiment annotation.

- *AspectAnnotation* - Contains information about aspect sentiment annotation.

- *OrdinaryResults* - Contains results for the overall annotation where only the label is predicted.

- *AspectBasedResults* - Contains results for the aspect-based annotation where the model predicts the place of the annotation and the label.

We have also created some additional functions that are used to convert input data set into the appropriate form, all are stored in the *transforms.py*.

3.4 Metrics used

We decided to use standard metrics for the annotation process, different for overall sentiment and aspect-based.

3.4.1 Overall sentiment

Apart from global accuracy, we calculate metrics for the multi-class sentiment classification using the macro approach of aggregating the results in different classes.

- Global accuracy:

$$GlobalAccuracy = \frac{correct\ classifications}{all\ classifications}$$

- Macro precision:

$$Precision_{macro} = \frac{1}{n}\sum_{i=1}^{n} Precision_{class\ i}$$

- Macro recall:

$$Recall_{macro} = \frac{1}{n}\sum_{i=1}^{n} Recall_{class\ i}$$

- Macro F1:

$$F1_{macro} = \frac{1}{n}\sum_{i=1}^{n} F1_{class\ i}$$

Where:

- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F1 = 2 \cdot \frac{precision \cdot recall}{precision+recall}$

3.4.2 Aspect-based sentiment

For aspect-based sentiment, the ‘confusion matrix‘ is defined in a slightly different way than in regular multiclass classification.

- correct (COR) - if the observation and its label are the same as the gold-standard annotation
- incorrect (INC) - if the observation is the same as the gold-standard annotation but has an incorrect label
- partial (PAR) - if the observation partially overlaps the gold-standard annotation and has the correct label
- missing (MIS) - if a gold-standard annotation does not occur in the result data set
- spurious (SPU) - if the observation does not occur in the gold-standard annotation

Using those terms we can define:

- possible (POS) - the number of annotations in the gold standard that contribute to the final score
  $POS = COR + INC + PAR + MIS = TP + FN$
- actual (ACT) - the total number of annotations produced by the system
  $ACT + COR + INC + PAR + SPU + TP + FP$

Finally, using those values we can define the actual metrics:

- $precision = \frac{correct}{actual}$
- $recall = \frac{correct}{possible}$
- $F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$

4 Experiments and Results

4.1 Explorative data analysis

In the first few paragraphs of this section, a proper and thorough introduction to the used data sets shall be carried out. For each data set, this has a similar structure that is: the distribution of reviews by their polarity, their length as well as word clouds highlighting the most important words in each data set.

4.1.1 Amazon data set

As described in the 2.2 section Amazon Reviews is a very big source of data and even its subset concerning product reviews from the *Electronics* field consists of about 20 000 000 opinions. Therefore, in this project, its subset of 5 000 observations is used and this part of the data set is described below.

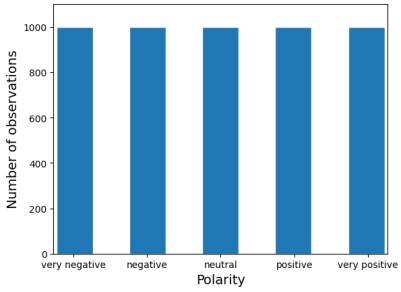

Figure 1: Amazon - reviews by polarity

The first figure 1 indicates that the taken subset was sampled evenly for each polarity of the reviews. Hence, there are exactly 1000 reviews that were marked as very positive, 1000 as positive and the same in the case of neutral, negative and very negative.

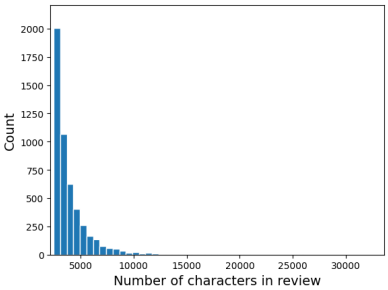

Figure 2: Amazon - reviews by length

The second graph 2 for the Amazon Reviews data set shows that the vast majority of the opinions about products are shorter than 10 000 characters although the distribution is right-skewed and there are some reviews with as many as 30 000 characters.

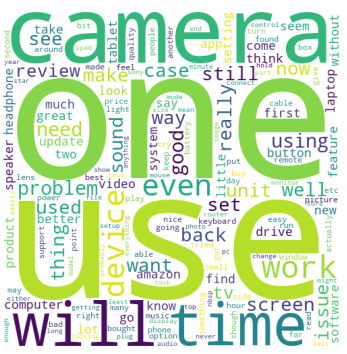

Figure 3: Amazon - word cloud

The final figure 3 in this subsection presents the most significant words that were used in this data set. Apart from words that are quite common for any written text it is worth noting that some of the listed elements are *camera*, *speaker*, *headphone*, *device*, *laptop* or *computer*. This list indicates that indeed those reviews focus on Electronics.

4.1.2 SemEval data set

The second data set is oriented on aspects and it is not as homogeneous as in the case of the previous one since it contains reviews of laptops as well as restaurants' reviews.

	Domain	Number of reviews	Number of aspects
0	Laptops	1482	2358
1	Restaurants	2019	3693
2	Total	3501	6051

Figure 4: SemEval - reviews & aspects count

The first figure 4 in this subsection confirms that the reviews and aspects are distributed between two domains with the majority in both groups being from the *Restaurants* part. Generally, there are approximately 1,73 aspects for each review.

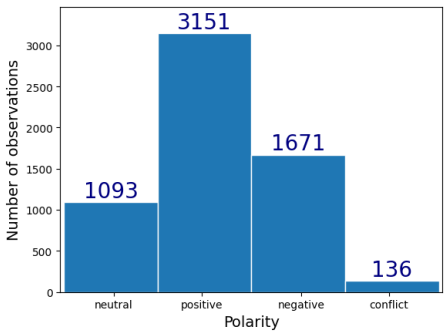

Figure 5: SemEval - aspects by polarity

The next figure 5 shows that this time there is no even distribution of observations - that is aspects instead of whole reviews - among polarities. Over half of the aspects have positive polarity while for only about 2% the polarity is conflicting.

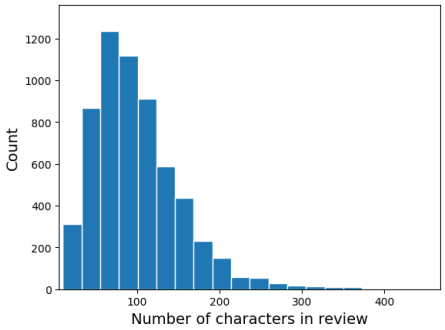

Figure 6: SemEval - observations by length

An interesting conclusion can be learnt from analysis of the figure 6. It turns out that almost all observations are shorter than 300 characters and it is a significantly lower value than in the case of Amazon Electronics data set.

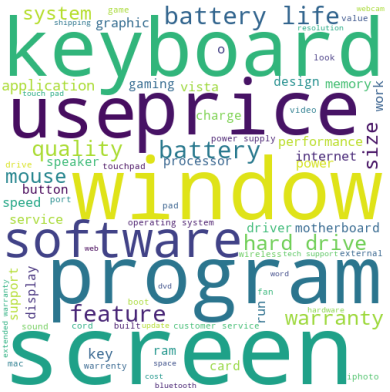

Figure 7: SemEval laptops - word cloud

Figure 8: SemEval restaurants - word cloud

This time 2 word clouds are shown so that differences between the two parts of the data set are visible. Figure 7 presents that in the laptops subset the most significant words are, for example, *keyboard*, *screen* or *program*, while from figure 8 it can be gathered that words *food*, *service* or *price* are among the most important in the Restaurants subset. Such choice of the most crucial words in each part of the SemEval data set quite well describes the nature of each of them.

4.1.3 PolEmo data set

The third data set is also the only one that is in Polish. As described in the introduction part of this paper it consists of reviews in 4 domains so they are more varied in comparison to the previous data sets and include 6573 reviews. PolEmo is designed for overall sentiment analysis with each review being assigned a separate polarity.

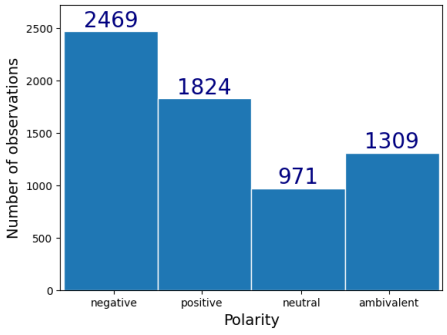

Figure 9: PolEmo - reviews by polarity

Distribution of polarity as visible in figure 9 is much more even than for the SemEval data set since the least popular group - neutral - consists of 15% of all reviews.

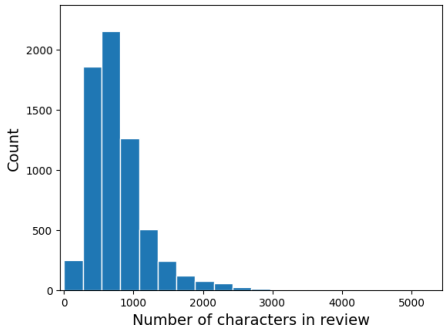

Figure 10: PolEmo - observations by length

Figure 10 shows that the majority of reviews are shorter than 1000 characters which means that on average opinions from the PolEmo data set are shorter than in the Amazon Electronics data set and longer than in the SemEval source.

Figure 11: PolEmo - word cloud

The final figure 11 in the Exploratory Data Analysis part paradoxically is interesting although the most significant words are usually conjunctions. However, it is expected since the reviews are from as many as 4 different domains and hence do not have that much in common.

4.2 Sentiment analysis tests

In the project, two task were tested - overall sentiment analysis and aspect based sentiment analysis. Due to the free-tier limitation on ChatGPT, a small subset of the dataset was used to calculate the results using this tool, for ABSA, no ChatGPT-based tools were tested.

4.2.1 Overall sentiment analysis

Figure 12 presents the metrics for sentiment analysis using ChatGPT, Flair and SentiStrength. Results shows, that Flair-based tool is the best for this task, with the highest F1, precission and recall. ChatGPT presents very similar accuracy to Flair, but it is not conclusive in this task and is probably a result of imbalance in classes.

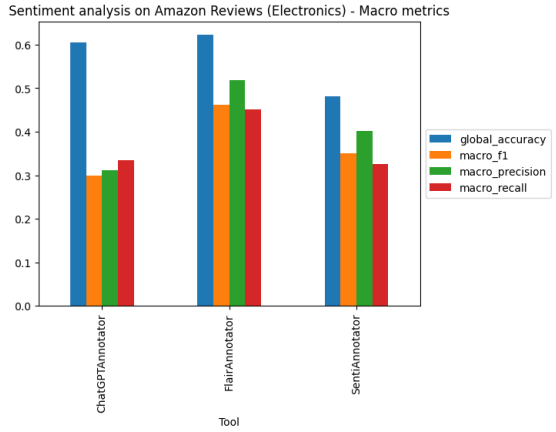

Figure 12: Sentiment analysis on Amazon Reviews (Electronics) - Macro metrics

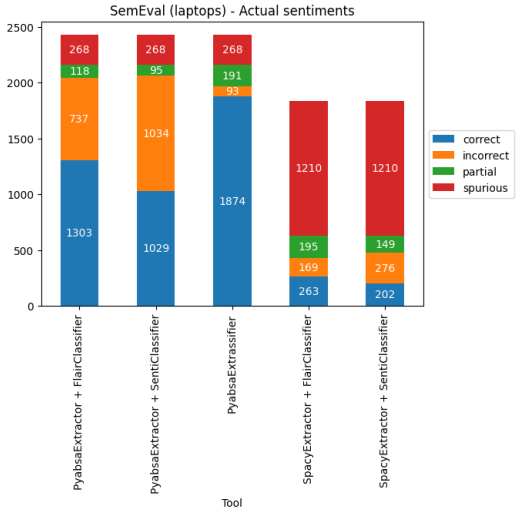

Figure 13: Actual labels for ABSA

4.2.2 Aspect-based sentiment analysis

To keep the report structure clean, detailed results will be analyzed only for one data set, because the conclusions are similar. However, the final metrics will be presented for both data sets. We tested only annotators not based on the Chat-GPT because of the problems with the free tier quota.

In Figure 13 are presented results of the metrics for actual labels from the Laptops data set. As can be observed, the PyABSA-based tools are achieving much better results. Among them, the full PyABSA workflow is the best and Flair is doing better than the SentiStrength classifier. The remaining two achieved much worse results and obtained a large amount of spurious labels. Figure 14 shows the same and highlights the problem of pipelines based on spaCy - the extractor is not working properly, and captures many not existing aspects, but at the same time omits aspects that are in the gold-standard data set. Figures 15 and 16 present the final metrics for each data set, which confirms observations already written.

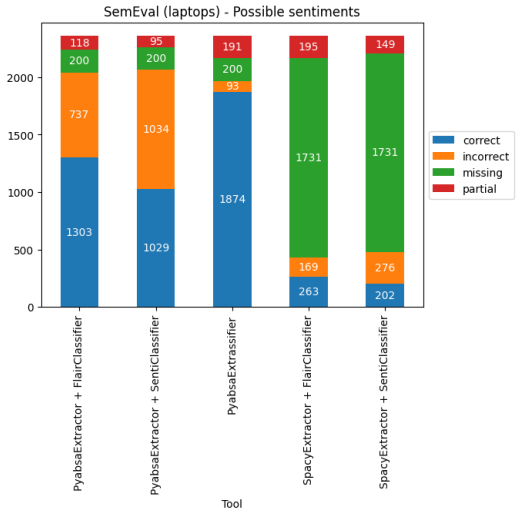

Figure 14: Predictions for ABSA

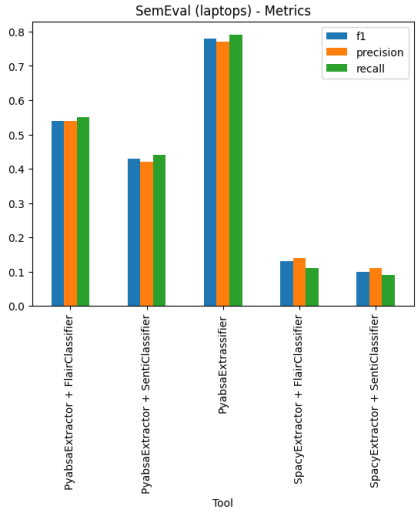

Figure 15: Metrics for laptops

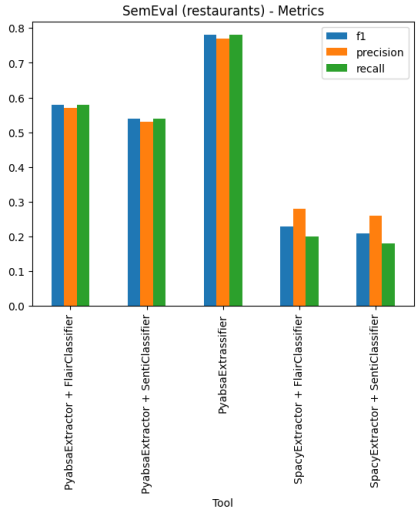

Figure 16: Metrics for restaurants

4.2.3 Polish data set annotations

Having compared different approaches on 2 English data sets the final part of this project could be carried out which is the preparation of a Polish data set for aspect-based sentiment analysis. To do that the PolEmo data set was used, although it should be remembered that in its original form, it was not designed for aspect-based sentiment analysis. Therefore, there was a need to transform it and for that purpose, the best tool from the carried out tests was chosen - PyABSA.

Before discussing the results two limitations should be raised. The first limitation that had to be somehow addressed was the language difference between the data sets for which the comparison of the tools was performed and PolEmo. Since the ultimate goal was to prepare a Polish dataset it would be ideal to compare the tools on Polish data but in reality, there was no other choice than to trust that if the documentation of, for example, PyABSA and ChatGPT, claim that the tool works in both English and polish and on the other hand it significantly outperforms its competitors on the test English sets then it will be also successful for polish data. The second limitation was rather technical since it turned out that without very expensive training for which we even didn't have data PyABSA cannot do aspect extraction and sentiment analysis for longer reviews than 375 characters. While training on SemEval data it was not observed since from figure 6 it can be noticed that almost all of these opinions were shorter than that. Therefore, some pre-processing was needed where out of 6573 reviews only 806 that were the shortest were chosen.

Finally, having in mind these two remarks the ultimate Polish data set for aspect-based sentiment analysis was generated. From these 806 reviews, 2489 aspects were extracted that were assigned to one of 3 groups:

- 940 negative
- 182 neutral
- 1367 positive.

At this point, some manual testing was performed. To do that 20 aspects from the data set were chosen at random and it was checked for how many of them the label was assigned properly. This simple test aimed to lay some foundations for more thorough testing in the future but even more importantly it was needed to assert that the whole process led to some satisfying results. It turned out that for these 20 aspects, as many as 18 of them were assigned correctly and that is an impressive result.

5 Discussion

This project's exploration into sentiment analysis, particularly aspect-based sentiment analysis (ABSA), using various NLP tools, presents several insights.

The major task achieved in the project was a comparative tool analysis where multiple pipelines to perform ABSA on English text were implemented. We could either perform ABSA in

two steps, first extract the aspects and then perform the sentiment analysis, or do the whole process in one run.

We observed that the PyABSA-based pipelines achieved the best results. A full PyABSA workflow gives the best results. SpaCy Extractor with Flair or Sentistrength for classification performed much worse, with Flair being better at classifying than Sentistrength.

Upon some manual inspection, we also observed a general trend that the tools working well with English datasets also worked well with Polish datasets.

These defined pipelines can then be used to annotate a Polish dataset, PolEmo. PolEmo in itself is not a dataset for ABSA, so it needs to be transformed into one, with correctly annotated aspects and sentiments. We chose the best-performing pipeline, PyAbsa, to carry out the annotation task. 18 out of 20 randomly chosen aspects had the aspect and the sentiment assigned to them correctly. A test like this, while certainly not thorough, lays a stepping stone for more testing in the future.

6 Conclusion and further work

To sum up the work that has been done in this project, after a thorough investigation of a few approaches where tools for aspect-based sentiment analysis were compared it was decided that PyABSA performs the best. Therefore, it was then used to annotate the PolEmo data set to prepare a source for aspect-based sentiment analysis in Polish. After simple manual testing, it was concluded that such an approach fulfilled expectations since for 20 checked aspects 18 of them were annotated correctly.

However, such a test is not enough to claim that this data set can be used for example for training of some new algorithms and therefore the first idea for further work is to make a proper manual correction of the prepared data set. Additionally, algorithms that performed slightly worse in the experiments part will be used for the PolEmo data set so that the resulting sets can be manually compared with one another because maybe the other tools perform even better with the Polish data.

We also intend to perform the computations using ChatGPT for ABSA, which were too time-consuming to do in the given timeframe.

A different idea that can also highlight the direction in which further work could go is the preparation of an application that would utilise the PyABSA algorithm to perform automatic annotation of some data and then expose a UI to non-IT people who can then easily perform manual corrections. However, this idea is probably second in line to be checked and will require more technical work related to application preparation.

References

[Augustyniak et al.2023] Łukasz Augustyniak, Szymon Woźniak, Marcin Gruza, Piotr Gramacki, Krzysztof Rajda, Mikołaj Morzy and Tomasz Kajdanowicz. 2023. *Massively Multilingual Corpus of Sentiment Datasets and Multi-faceted Sentiment Classification Benchmark*.

[Liu and Zhang2012] Bing Liu and Lei Zhang 2012. *A SURVEY OF OPINION MINING AND SENTIMENT ANALYSIS*.

[McAuley et al.2015] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. *Image-based recommendations on styles and substitutes*. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 43–52.

[Pontiki et al.2014] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. *SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

[Kocoń, Zaśko-Zielińska and Miłkowski2019] Kocoń, Jan and Zaśko-Zielińska, Monika and Miłkowski, Piotr 2019. *PolEmo 2.0 Sentiment Analysis Dataset for CoNLL*, CLARIN-PL digital repository

[Devlin et al.2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805.

[Akbik et al.2019] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. *FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pages 54–59.

[Akbik et al.2018] Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2019. *Contextual String Embeddings for Sequence Labeling* . Proceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649.

[Thelwall et al.2010] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas.

2010. *Sentiment Strength Detection in Short Informal Text*. Journal of the American Society for Information Science and Technology, 61(12):2544–2558.

[OpenAI2021] OpenAI. 2021. *ChatGPT: Optimizing Language Models for Dialogue*. `https://openai.com/blog/chatgpt`.

[Honnibal and Montani2017] Matthew Honnibal and Ines Montani. 2017. *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*. To appear.

[Yang and Zhang and Li2023] Matthew Honnibal and Ines Montani. 2023. *PyABSA: A Modularized Framework for Reproducible Aspect-based Sentiment Analysis*.

[Sanh et al.2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv preprint arXiv:1910.01108.