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 dataset, including aspects of a given text and sentiment towards this aspect. While several datasets in English are available for this task, there is a lack of such a dataset in the Polish language. In this work, we test and compare the performance of several novel tools for sentiment analysis and aspect-based sentiment analysis on English datasets and describe the infrastructure of a framework we created for this purpose. Based on the results, we chose the most efficient one (PyABSA), which achieved a 0.78 F1 score on the SemEval dataset. We used it on an existing Polish dataset PolEmo, suitable for sentiment analysis, to create a new version of it with aspects and sentiments included. This new dataset, after manual refinement, 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. Building upon the foundation of traditional sentiment analysis, aspect-based sentiment analysis (ABSA) delves deeper by discerning the sentiment and linking it to specific aspects or attributes of a product or service.

The project's goal centres around evaluating sentiment analysis of product and service-related reviews. 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 research gap in the realm of Polish language datasets specifically tailored for aspect-based analysis in sentiment evaluation. Presently, there are available English datasets like Amazon Reviews (McAuley et al., 2015) and SemEval-2014 (Pontiki et al., 2014), which cater to aspect-based analysis, and one Polish dataset, PolEmo (Kocoń, Zaśko-Zielińska and Miłkowski, 2019), primarily designed for general sentiment analysis. However, there is a noticeable absence of a dedicated Polish dataset geared towards aspect-based sentiment analysis. Creating such a dataset constitutes a pivotal contribution, as it fills a crucial gap in the domain. Although our dataset still needs some manual refinements, 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.

Another important contribution is creating and thoroughly describing the infrastructure of a unified framework for comparing different tools for overall and aspect-based sentiment analysis, available as a Python package `pysent`.

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 datasets 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, Section 6 - conclusion and suggestions for further work and Section 7 - some ethical considerations.

2 Related work

Sentiment analysis, a fundamental aspect of natural language processing, has garnered significant attention due to its applications across various domains. This analysis is typically binary, classifying sentiments as positive or negative, and sometimes even neutral, though it can also cover a range of emotions like happiness, anger, or sadness (Cambria et al., 2012) . 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 (Rambocas et al., 2018).

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). For instance, in a restaurant review, ABSA helps distinguish the customer's sentiment about various aspects such as food quality, service, ambience, and price. This granular approach provides a more detailed sentiment overview, which is crucial for businesses and service providers aiming to pinpoint strengths and areas of improvement based on customer feedback.

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 Novel Sentiment Analysis Tools

Several tools are available to perform the sentiment analysis, both overall and aspect-based. Below is the summary of tools chosen to be evaluated in this study, as they seemed suitable for our use case, popular and quite easy to use.

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 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. The development of SentiStrength involved analyzing an initial dataset of 2,600 human-classified MySpace comments, followed by evaluation using an additional random sample of 1,041 MySpace comments. It achieved 60.6% accuracy on recognizing positive emotion and 72.8% accuracy on recognizing negative emotion.

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

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 languages, including English and Polish, and trained pipelines for models provided for each language. It is designed to be suitable for large-scale problems and extraction tasks, ensuring appropriate processing speed – both CPU and GPU hardware options are supported.

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

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 apply 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 their specific engineering challenges. 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].

2.1.4 ChatGPT

ChatGPT (2021) is a novel tool perfect for processing text data and 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 user's prompts. The dialogue format allows the model to answer follow-up questions, admit mistakes, and reject inappropriate requests. However, the unstructured response format poses a problem. It requires careful prompt engineering to achieve the response in the desired format and then pars it to a more structured data form.

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 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 datasets, but it also allows for extensions and use of your own models and datasets. 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 dataset. 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.

2.2 Datasets

For this project, we utilized 3 datasets containing reviews: two datasets in English (one with and one without gold standard aspects present) and one dataset in Polish (without aspects).

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

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

2.2.1 Amazon dataset

The Amazon Reviews dataset (McAuley et al., 2015) is a collection of reviews written in English by customers for products purchased on Amazon. It is one of the largest and most commonly used datasets for sentiment analysis and natural language processing tasks. This dataset includes reviews spanning various product categories, providing a broad range of vocabulary and topics.

2.2.2 SemEval dataset

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 – Extracting 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 dataset is included in Section 4.1.2.

2.2.3 PolEmo dataset

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). Ambivalent can show mixed feelings or contradictory emotions, both positive and negative. An example of "ambiwalnetny" could be "I'm excited about the new job, but also scared about the changes" and a neutral statement could be "I have a job interview at 3pm tomorrow" .The preprocessing of this dataset is covered in Section 4.1.3.

3 Our approach and research methodology

ABSA (Zhang et al., 2022) 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 understanding complex relationships in the text.

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. As a standard Python package, this repository holds all of the codes 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 datasets as an estimator of how good the tool is in working with Polish sentences.

3.2 Ways of using chosen tools

Below we summarize the way each of the tools described in section 2.1 is used in our project. Due to the large number of used models and the small amount of options to tune them, we decided to use default parameters for each of the selected tools.

- **SentiStrength**

This tool is designed for overall Sentiment Analysis and 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 ABSA, where the text is first split into chunks, or the aspects of the text are extracted using another tool.

- **Spacy**

  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 the project since pipelines with SpaCy as the first-step tool did not achieve the best result.

- **Flair**

  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.

- **ChatGPT**

  It is used 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.

- **PyABSA**

  PyABSA is used 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.

3.3 Limitations of chosen tools

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

- **ChatGPT**

  According to OpenAI Documentation, the 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 datasets using this tool. We present the results only for overall sentiment analysis, and even for this task, we use a subset of the data available. Based on our calculations, training the Amazon electronics subset dataset of 5000 reviews would cost around $14, and $58666 for the entirety of the dataset. 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.

- **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 intended purpose.

3.4 Description of the package

The package is built to provide users with easily accessible pretrained models allowing annotation, 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. An annotator here, means, a tool that will provide an annotation for the text, which in this case means, a tool that shall provide the sentiment.

3.4.1 OverallAnnotator

OverallAnnotator is a class that as an input has a tool class that inherits from *OverallAnnotatorAbstract* (described further). *OverallAnnotator* 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 structure of *AspectAnnotator* and to keep it easily extensible.

Overall annotators are classes that inherit from *OverallAnnotatorAbstract* class. The *OverallAnnotatorAbstract* class is an interface that has meth-

ods *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.

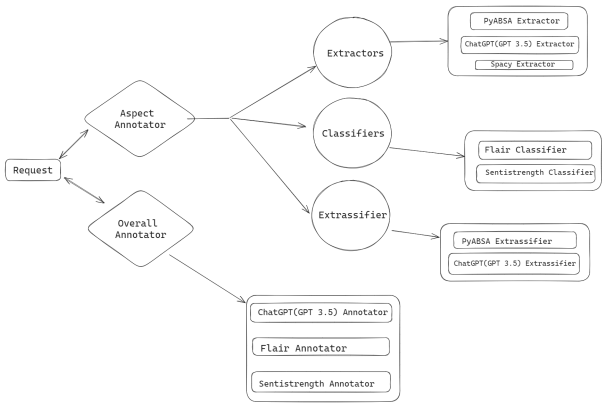

Figure 1: Breakdown of classes and flow in Pysent

3.4.2 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:

- Extracting 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 their 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, which is a parameter in this class.
- *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. The prompt is as follows:

> For text below provide me a sentiment analysis label and score in the format:
> Label: <label you suggest >
> Score: <score you suggest >
> Text: "the text provided for sentiment analysis"

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 tool itself. To cover this, we have decided to add this functionality and contribute to this open-source package. We added a keywords parameter that allows the possibility to perform ABSA with the supplied keyword. 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 options.

Extensive descriptions of all of the classes can be found in the documentation provided here

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

- *ExtractedAspect* - a dataclass representing the output of extraction aspect tools. Has an aspect keyword and context in text.
- *SentimentAnnotation* - a dataclass containing 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 (sentiment).

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

3.5 Metrics used

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

3.5.1 Overall sentiment

Apart from global accuracy, we calculate metrics for the multi-class sentiment classification using both macro and micro approaches 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:

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

- Micro precision:

$$Precision_{micro} = \frac{\sum_{i=1}^{n} TP_{class_i}}{\sum_{i=1}^{n} TP_{class_i} + \sum_{i=1}^{n} FP_{class_i}}$$

- Micro recall:

$$Recall_{micro} = \frac{\sum_{i=1}^{n} TP_{class_i}}{\sum_{i=1}^{n} TP_{class_i} + \sum_{i=1}^{n} FN_{class_i}}$$

- Micro F1:

$$F1_{micro} = 2 \cdot \frac{Precision_{micro} \cdot Recall_{micro}}{Precision_{micro} + Recall_{micro}}$$

Where:

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

3.5.2 Aspect-based sentiment

For aspect-based sentiment, the ‘confusion matrix‘ is defined in a slightly different way than in regular multi-class 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 dataset,

- 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 datasets shall be carried out. For each dataset, 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 dataset. However, at first the general characteristics of the 3 datasets shall be displayed so that they can be compared conveniently:

Dataset	Avg length	sentiment task	label count
Amazon	3987	overall	5
PolEmo 2.0	758	overall	4
SemEval '14	100	aspect-based	4

Table 1: Basic statistics of the datasets

Additionally, prior to discussing each dataset separately a figure comparing distribution of lengths:

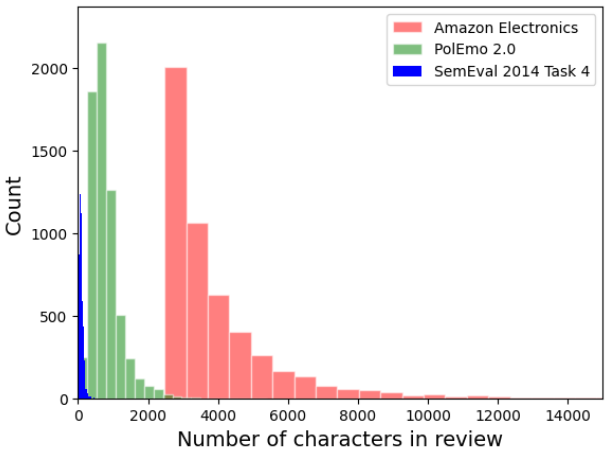

Figure 2: Reviews' lengths distribution

Figure 2 and table 2.2 show that the reviews in the Amazon - Electronics dataset has the longest reviews while the reviews in the Sem Eval dataset are the shortest and they are the only one dedicated to the aspect-based sentiment analysis task.

4.1.1 Amazon dataset

As described in Section 2.2, 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 product reviews. Therefore, in this project, its subset of 5 000 observations - chosen with stratified sampling meaning that there is even number of observations for each of the 5 available labels - is used and this part of the dataset is described below.

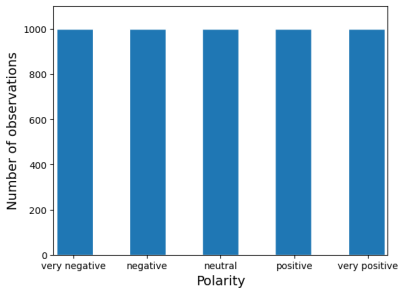

Figure 3: Distribution of reviews from the Amazon Electronics dataset by the gold-standard sentiment

Figure 3 indicates that the taken subset was sampled evenly for each gold-standard label of the reviews. Hence, there are exactly 1,000 reviews that were marked as very positive, 1,000 as positive and the same in the case of neutral, negative and very negative.

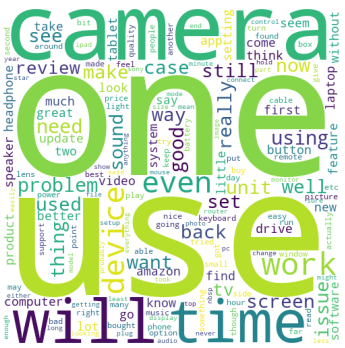

Figure 4: Amazon - word cloud

Figure 4 presents the most significant words that were used in this dataset. 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 dataset

The second dataset 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	# of reviews	# of aspects
Laptops	1482	2359
Restaurants	2019	3693
Total	3501	6051

Table 2: SemEval - reviews & aspects count

Table 2 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.

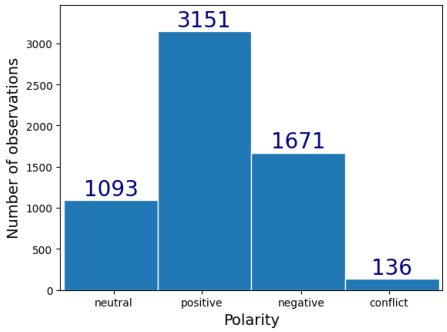

Figure 5: Distribution of reviews from the SemEval dataset by the gold-standard polarity

Figure 4.1.2 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.

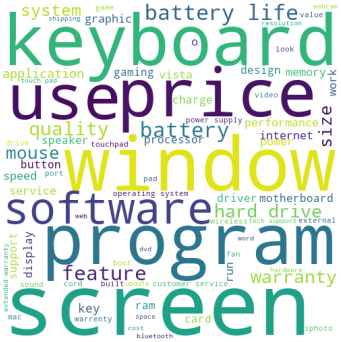

Figure 6: SemEval laptops - word cloud

Figure 7: SemEval restaurants - word cloud

This time 2 word clouds are shown so that differences between the two parts of the dataset are visible.

4.1.3 PolEmo dataset

The third dataset 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: medicine, hotels, products and school so they are more varied in comparison to the previous datasets and include 6,573 reviews. PolEmo is designed for general sentiment analysis with each review being assigned a single sentiment as a whole - in contrast to the SemEval dataset where sentiments were assigned to aspects.

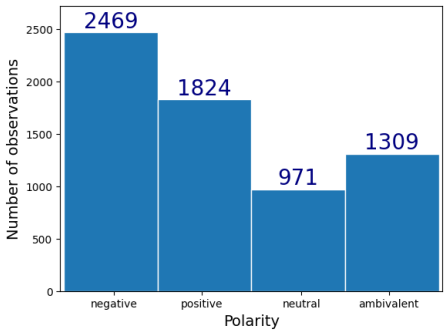

Figure 8: Distribution of reviews from the PolEmo dataset by the gold-standard polarity

Distribution of polarity as visible in Figure 8 is much more even than for the SemEval dataset since the least popular group - neutral - consists of 15% of all reviews.

Figure 9: PolEmo - word cloud

The final figure 9 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 Results

In the project, two tasks 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 10 presents the metrics for sentiment analysis using ChatGPT, Flair and SentiStrength. Results show, that Flair-based tool is the best for this task, with the highest F1, precision and recall. ChatGPT presents very similar accuracy to Flair.

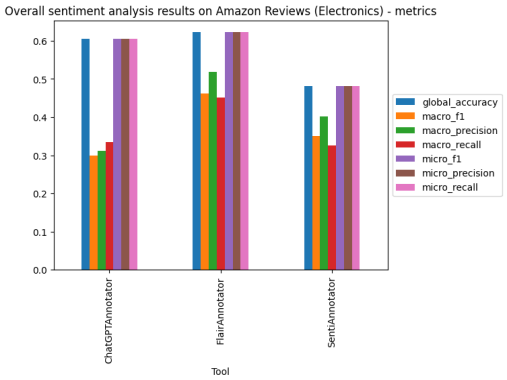

Figure 10: Accuracy, Macro and Micro metrics for overall sentiment analysis on Amazon Reviews (Electronics) dataset using ChatGPT, Flair and SentiStrength

4.2.2 Aspect-based sentiment analysis

To keep the report structure clean, detailed results will be analyzed only for the SemEval laptops subset, as the conclusions for the restaurants subset are similar. However, the final metrics will be presented for both datasets. We tested only annotators not based on the Chat-GPT because of the problems with the free tier quota.

Figure 11 presents the results of the metrics for actual labels from the SemEval laptops subset. 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 many spurious labels. Figure 12 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 dataset.

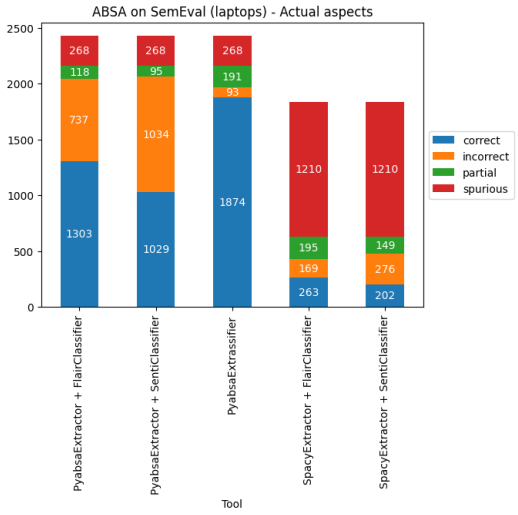

Figure 11: The division of the aspects and sentiments annotations found by different tools during ABSA. The detailed definition of actual, correct, incorrect, partial and spurious can be found in section 3.5.2

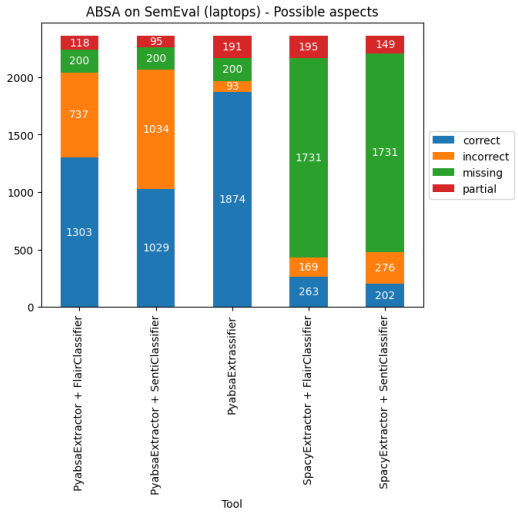

Figure 12: The division of the aspects and sentiments from the gold standard annotations during ABSA. The detailed definition of possible, correct, incorrect, missing and partial can be found in section 3.5.2

Figures 13 and 14 present the final metrics for each dataset, which confirms observations already written.

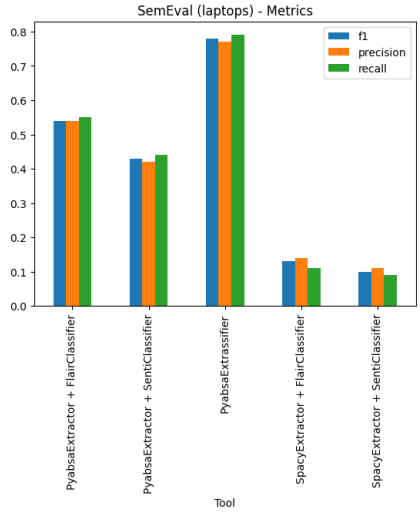

Figure 13: Comparison of global metrics achieved on SemEval (laptops) dataset by different tools used for ABSA

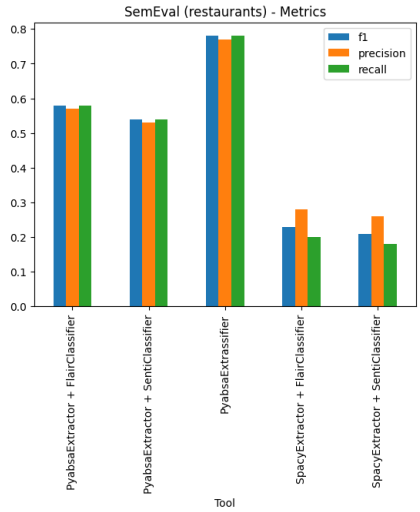

Figure 14: Comparison of global metrics achieved on SemEval (restaurants) dataset by different tools used for ABSA

4.3 Example results

Examples presented below are provided to understand the results better. In the tables aspects are presented in the following manner - ”example aspect” (S), where S is a classified sentiment (P - positive, N - negative, Ne - neutral).

Example 1: ”I would recommend this computer to anyone searching for the perfect laptop, and the battery life is amazing.

In the Table 3 there are presented the extracted aspects. The main mistake of tools based on SpaCy extraction is taking only ”life” (without

”battery”) as an aspect. The second issue is taking unnecessary ”computer” aspect. This one is ambiguous since one can stand that because it is preceded by the phrase ”I would recommend”, it should be treated as an aspect and should be extracted and classified as positive.

Tools used	Aspects
PyAbsa + Flair	”battery life” (P)
PyAbsa + SentiStrength	”battery life” (P)
SpaCy + Flair	”computer” (P) ”life” (positive)
SpaCy + SentiStrength	”computer” (P) ”life” (positive)
PyAbsa	”battery life” (P)

Table 3: Comparison of tools - example 1

Example 2: When I called Sony the Customer Service was Great.

The second example presented Table 4 is simple, SpaCy based tools takes incorrect word as a aspect.

Tools used	Aspects
PyAbsa + Flair	”Customer Service” (P)
PyAbsa + SentiStrength	”Customer Service” (P)
SpaCy + Flair	”Sony” (P)
SpaCy + SentiStrength	”Sony” (P)
PyAbsa	”Customer Service” (P)

Table 4: Comparison of tools - example 2

4.3.1 Time comparison

Table 5 presents the results of average time (measured on 100 texts) per annotation. For the overall annotation tools, Flair turned out to be faster than SentiStrenght, but surprisingly the results are the opposite in aspect annotation. Tools based on PyAbsa are much slower than the rest. Overall, all of the tools are working very fast, but for some real-time analysis with big data, usage of the faster, yet worse ones could be justified.

4.3.2 Polish dataset annotations

Having compared different approaches on two English datasets, the final part of this project could be carried out which is the preparation of a Polish dataset for aspect-based sentiment analysis. To accomplish that, the PolEmo dataset was used, although it should be remembered that in its original form, it was not designed for aspect-based

Exercise Tools used	Aspects	Time
Overall annotation	Flair	0.01235
Overall annotation	SentiStrength	0.04280
Aspect annotation	PyAbsa + Flair	0.22319
Aspect annotation	PyAbsa + SentiStrength	0.19688
Aspect annotation	SpaCy + Flair	0.01773
Aspect annotation	SpaCy + SentiStrength	0.00714
Aspect annotation	PyAbsa	0.21856

Table 5: Comparison of tools' time to annotate

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. By transforming the dataset it is meant that the aspects need to be extracted from the reviews and then, sentiment shall be assigned to these labels - this is exactly the process know and described in this report as aspect-based sentiment analysis.

Before discussing the results, two limitations should be mentioned. The first limitation that had to be somehow addressed was the language difference between the datasets 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 to Polish data. Still, 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 also be successful for Polish data. The second limitation was rather technical since it turned out that without very expensive training for which we did not even 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 2, it can be noticed that almost all of these opinions were shorter than that. Therefore, some pre-processing was needed, and out of all 6,573 reviews that were available in the PolEmo dataset, only 806 that were the shortest were chosen.

Finally, having in mind these two remarks the ultimate Polish dataset for aspect-based sentiment analysis was generated. From these 806 reviews, 2,489 aspects were extracted that were assigned to one of 3 groups:

- 940 negative,

- 182 neutral,
- 1367 positive.

In order to properly assess the quality of the generated Polish dataset the following checking procedure was chosen:

1. Out of the annotated dataset that has 806 reviews a random sample of 100 reviews was chosen - this subset contains 398 aspects that were labelled as positive, neutral or negative.
2. Each of these 398 aspects was manually labelled by one person.
3. The manually assigned labels were compared with those generated automatically to assess accuracy. The resulted value was 80.402%.

Since the dataset that was prepared is designed for aspect based sentiment analysis, then the calculated accuracy is also accuracy of labels assigned to aspects.

Finally, because the aim of the project was to prepare a Polish dataset for aspect-based sentiment analysis tasks, a manually corrected version of this transformed dataset was needed. For that purpose the same subset of the dataset annotated with PyABSA was used as the one used for accuracy calculation. The labels were then corrected manually and such a smaller dataset can be treated as a basis for further development (remaining parts of automatically annotated dataset can be corrected in the same way) and also sharing as open-source.

An example of two reviews with labels generated by the algorithm and the corrected one:

1. - **Review** Sympatyczny pobyt 3 , dniowy . Pogoda była okropna . ale hotelik śliczny i miła obsługa - szczególnie damski personel zrekompensowało mi brak słońca . Jedzenie zdecydowanie ładniej i smaczniej wyglada na materiałach reklamowych , ale nie jestem obżartuchem wiec nie ma problemu . Nie wybacze jednak ceraty pod prześcieradłem , dlatego odejmuje jedna gwiazdke ,
   - **Label** damski personel
   - **Automatically assigned sentiment** positive
   - **Manually assinged sentiment** positive
2. - **Review** Sympatyczny pobyt 3 , dniowy . Pogoda była okropna . ale hotelik śliczny i miła obsługa - szczególnie damski personel zrekompensowało mi brak słońca . Jedzenie zdecydowanie ładniej i smaczniej wyglada na materiałach reklamowych , ale nie jestem obżartuchem wiec nie ma problemu . Nie wybacze jednak ceraty pod prześcieradłem , dlatego odejmuje jedna gwiazdke ,
   - **Label** Jedzenie
   - **Automatically assigned sentiment** positive
   - **Manually assinged sentiment** neutral

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 based on the Figure 11 and 12. Therefore, it was then used to annotate the PolEmo dataset 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 dataset can be used, for example, for training some new algorithms, and therefore the first idea for further work is to make a proper manual correction of the prepared dataset. Additionally, algorithms that performed slightly worse in the experiments part will be used for the PolEmo dataset 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 utilize the PyABSA algorithm to automatically annotate 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.

7 Ethical considerations

In our work, we focus on sentiment analysis of different kinds of reviews of products and services. Such analysis can help gain insight into the satisfaction and experiences of customers. It is important to note, however, that one could the tools described in this work to identify critical or unfavorable content. The risks associated with that can include the ability to remove such reviews or even punish their creators by blocking them from the platform where the reviews can be submitted. This can be a valid action if the reviews contain inappropriate language or other harmful or untrue statements. In other cases, however, we do not condone such actions and encourage the service providers and product creators to pay attention to the constructive criticism that the reviews may contain.

8 Contributions

The responsibilities of individual contributors during conducting the project are listed in the tables below:

8.1 Anish Gupta

Codes for fine-tuning BERT and DistilBert	7 h
Execution of prepared notebook OverallSentiment	4 h
Research	4 h
Report: *Introduction*, *Discussion*	3 h
Report checkup and addressing comments by Prof.	3 h
Presentations preparation & delivery	2 h
Overall	**23 h**

8.2 Martyna Majchrzak

Codes for full ChatGPT, ChatGPT + Flair and ChatGPT + Sentistregth in the Python package	9 h
Results visualisations	3 h
Report: *Abstract*, *Introduction*, *Related work*, *Ways of using chosen tools*, *Ethical considerations*, *Contributions*, *Rebuttal* sections	8 h
Report corrections and checkup	2 h
Presentations preparation & delivery	2 h
Overall	**24 h**

8.3 Bartosz Rożek

Python package codes, structure and documentation	14 h
Research	4 h
Report: *Our approach and research methodology*, *Sentiment analysis tests*, *Time comparison*	5 h
Presentations preparation & delivery	2 h
Overall	**25 h**

8.4 Konrad Welkier

Codes for datasets preparation	4 h
EDA	4 h
Polish dataset generation + assessment + manual correction	5 h
Overall sentiment metrices	3 h
Report: *Explorative data analisys*, *Polish dataset annotations*, *Conclusion and further work*, *Appendices*;	5 h
Report check-up	1 h
Presentations preparation & delivery	2 h
Overall	**24 h**

9 Rebuttal

We have received multiple suggestions for improvements to this report and our codes from Prof. Anna Wróblewska and other students attending the NLP course who reviewed our work. Below we address some of those questions.

We have received inquiries about the lack of saved models, setting random state in our codes and the parameters used for training. We want to underline that all of the tools used in this project contained pre-trained models for sentiment analysis that do not require further training, so there was no need to set a random state or save models, as no new model was created. As we did not have a dataset in Polish suitable for this task, so training a model specifically for this task from scratch was not an option. We acknowledge, however, that setting random state in research projects is a good practice and we will remember about it in our future work.

We have conducted a more thorough comparison between the tools, including examples of aspects found (4.2.2) and time of execution (4.3.1) and some further manual assessment of the created Polish dataset (4.3.2).

Initially, we have not included the micro metrics in overall sentiment analysis results, as they all seem to have the same value. Per request, we have included them in figure 10.

We added the 'Ethical considerations' section 7 to underline the potential risks associated with sentiment analysis of reviews and a simple datasheet in the Appendix A.

We also introduced the following editorial changes:

- more concise descriptions in the Exploratory data analysis section 4.1
- grammatical corrections (such as Polish written with a capital letter)
- added references for ChatGPT4.0, ChatGPT3.5 an PyABSA
- adding `wordcloud` package to the requirements.txt file in code
- table of contributions was added
- longer captions under plots and overall improved plot layout in section 4.1 and 4.2
- moving the descriptions of way the tools were used in our project from section *Related work* 2 to *Our approach and research methodology* 3
- some introduction at the beginning of the 'Datasets' section 2.2
- the *Introduction* section 1 was shortened and some details were moved to *Related work* 2
- added *Pysent* package workflow diagram 1

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.

[Cambria et al.2012] Erik Cambria, Andrew Livingstone, and Amir Hussain. 2012. *The Hourglass of Emotions*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7403 LNCS, pages 144–157.

[Rambocas et al.2018] Meena Rambocas and Barney G. Pacheco. 2018. *Online sentiment analysis in marketing research: a review*. Journal of Research in Interactive Marketing, Vol. 12 No. 2, pages 146–163.

[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. Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023 ACM *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.

[Zhang et al.2022] Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, Wai Lam. 2022. *A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges*. In ar5iv.org.

Appendices

A Dataset

Summary of the dataset developed in the project.

- DATA STATS – 100 reviews, 398 aspects with each of them labelled as *positive*, *negative* or *neutral*
- DATA DOWNLOAD – the dataset can be downloaded from the link: https://github.com/welkierk/NLP-BAMK-project/blob/main/data/polemo_labelled_corrected.xlsx
- NEW DATA DESCRIPTION – The dataset is created based on the PolEmo dataset that can be downloaded from here: https://clarin-pl.eu/dspace/handle/11321/710. The process involved a few steps that were: automatic aspect extraction and labelling with the PyABSA tool, random sampling 100 reviews and then manually correcting the automatically generated labels.
- DATA LANGUAGES – all of the reviews and aspects are in Polish while all of the labels are in English (due to the fact that the tool used to generate them operates primarily in English)