PRODUCTS OPINIONS & NEWS

BAMK

(A. Gupta, M. Majchrzak,
B. Rożek, K. Welkier)

AGENDA

Topic

Datasets

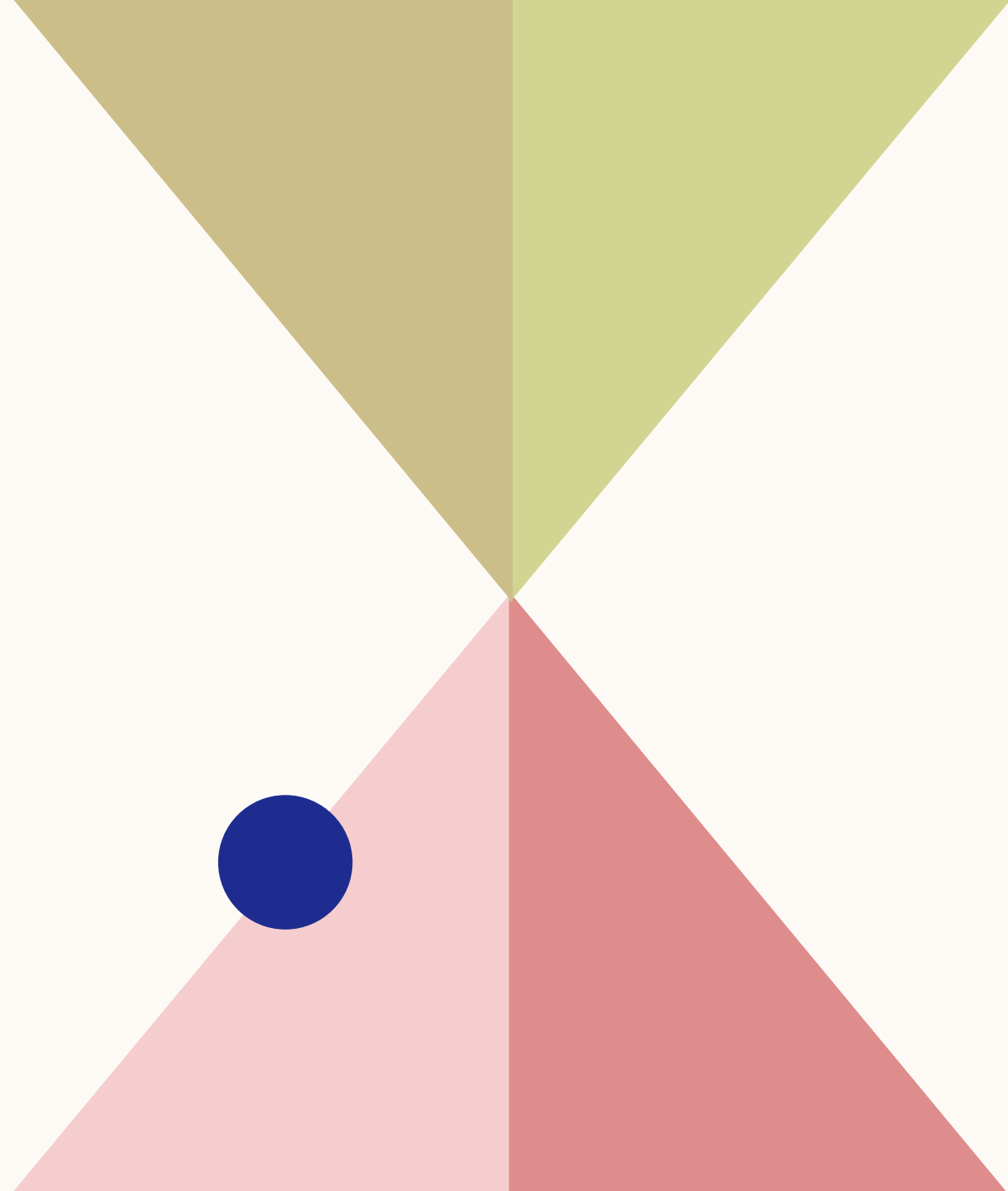
State of the art

Experiments

Metrics

Results

Ideas for Project 2



TOPIC

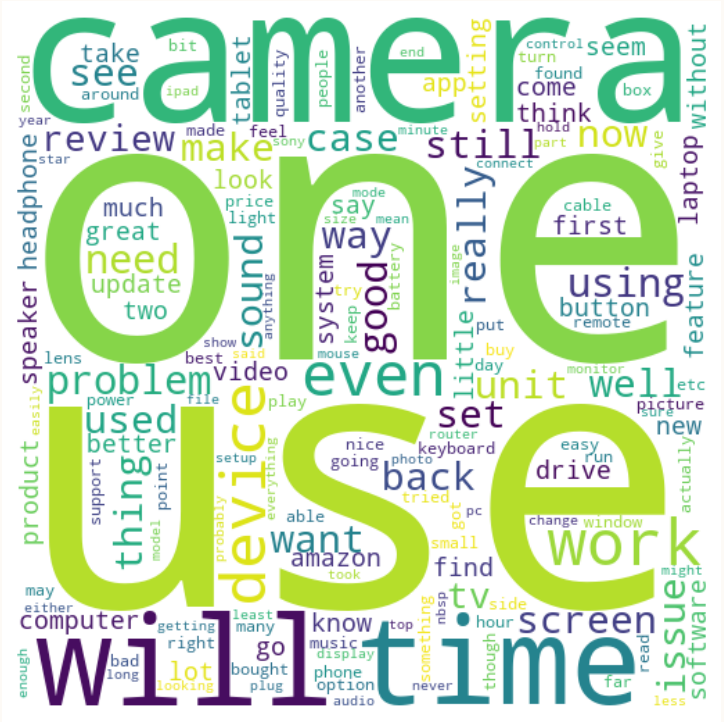
Project focuses on:

- **Sentiment** and **aspect-based** sentiment analysis of **product news articles**
- Finding **sentiment towards different mentioned aspects from the reviews**
- Identifying products and their attributes
- Preparation of a **polish dataset for aspect-based sentiment analysis**

DATASETS

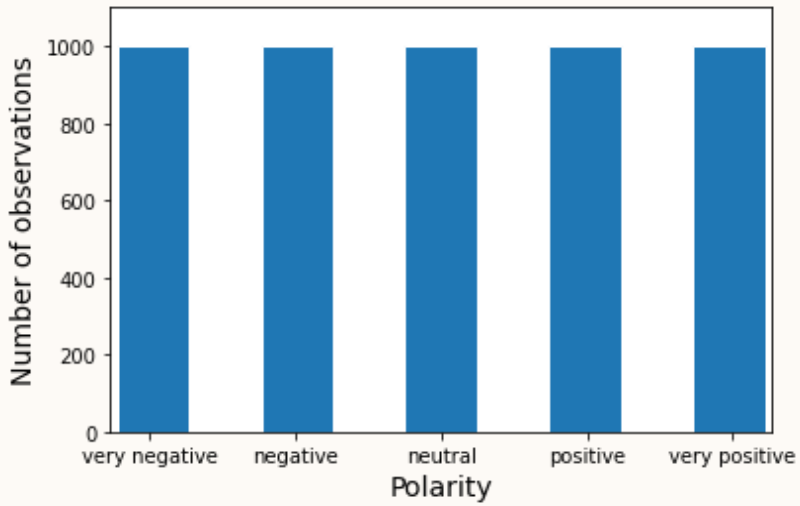
Testing the obtained results

AMAZON REVIEWS (ELECTRONICS)



Word Cloud

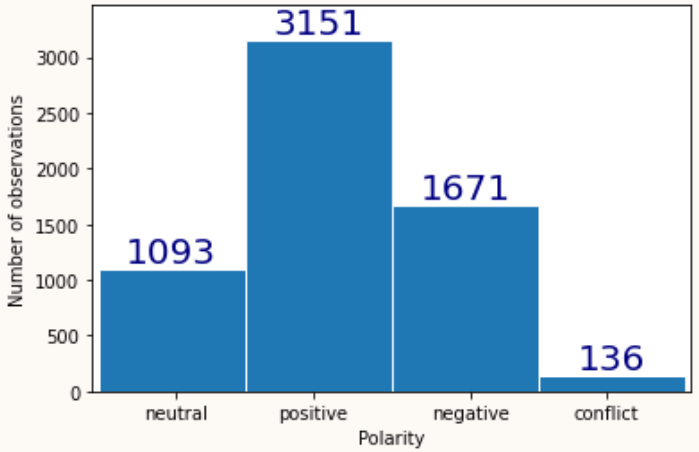
4K longest reviews from over 20M in the Electronics dataset



Distribution of reviews by polarity

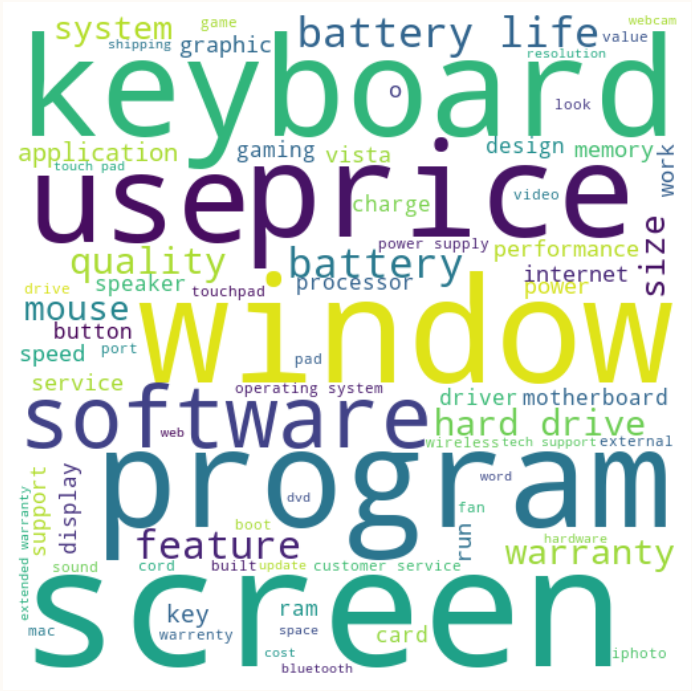
SEMEVAL-2014
(LAPTOPS & RESTAURANTS)

Distribution of all reviews by polarity



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

Distribution of reviews and aspects by data set

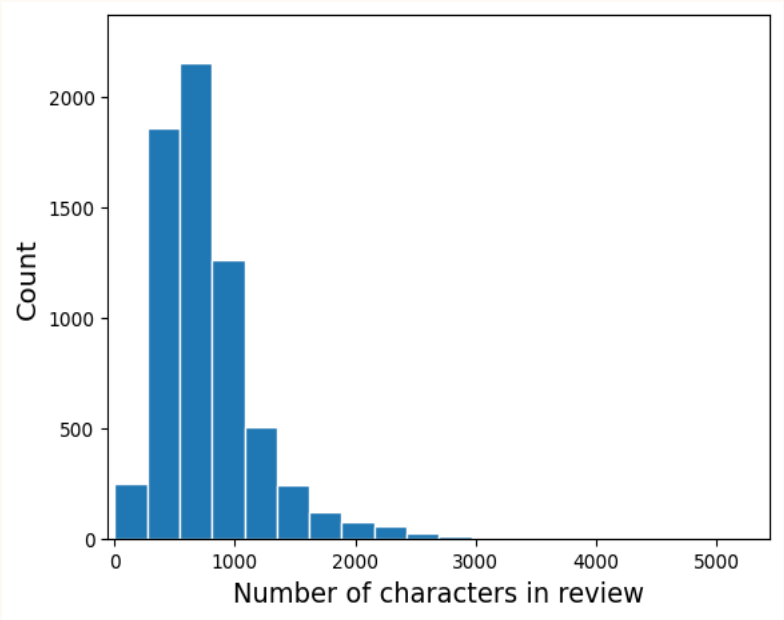


Word Cloud for laptops subset

Word Cloud for restaurants subset

POLEMO 2.0

consumer reviews from 4 domains: medicine, hotels, products and school



Distribution of lengths of the reviews

Word Cloud

STATE OF THE ART

What to use?

CHAT GPT

flair

spaCy

SENTISTRENGTH

PYABSA

Google BERT

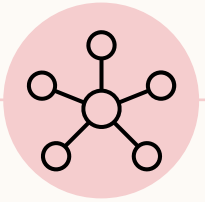
EXPERIMENTS

Make a use of the SOTA tools

TYPES OF EXPERIMENTS

OVERALL SENTIMENT

- Chat GPT
- SentiStrength
- Flair



ASPECT BASED SENTIMENT

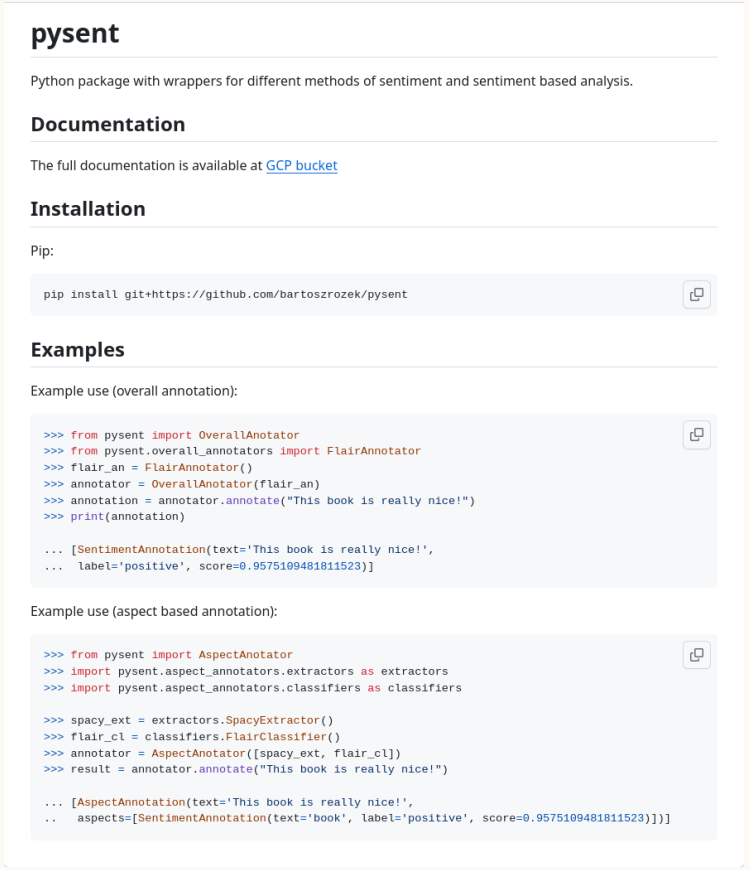
One-step approach:

- Chat GPT
- PyABSA

Two-step approach:

- Chat GPT + Flair/SentiStrength
- PyABSA + Flair/SentiStrength
- Spacy + Flair/SentiStrength

ADDITIONAL PROJECT OUTPUT



pysent

Python package with wrappers for different methods of sentiment and sentiment based analysis.

Documentation

The full documentation is available at GCP bucket

Installation

Pip:

```
pip install git+https://github.com/bartoszrozek/pysent
```

Examples

Example use (overall annotation):

```
>>> from pysent import OverallAnotator
>>> from pysent.overall_annotators import FlairAnnotator
>>> flair_an = FlairAnnotator()
>>> annotator = OverallAnotator(flair_an)
>>> annotation = annotator.annotate("This book is really nice!")
>>> print(annotation)

... [SentimentAnnotation(text='This book is really nice!',
...  label='positive', score=0.9575109481811523)]
```

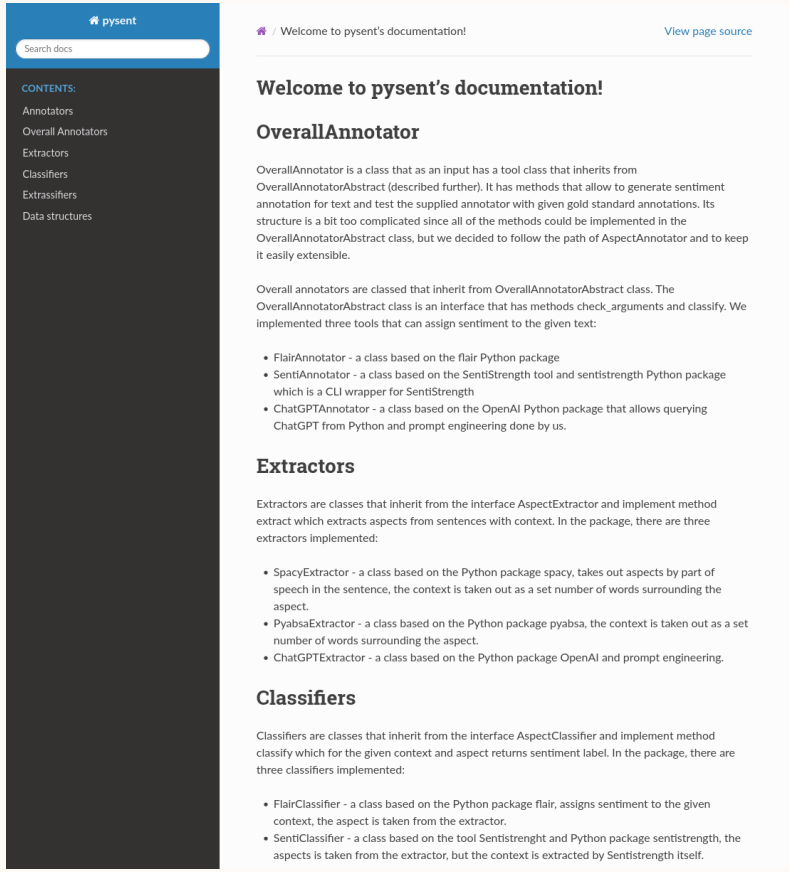
Example use (aspect based annotation):

```
>>> from pysent import AspectAnotator
>>> import pysent.aspect_annotators.extractors as extractors
>>> import pysent.aspect_annotators.classifiers as classifiers

>>> spacy_ext = extractors.SpacyExtractor()
>>> flair_cl = classifiers.FlairClassifier()
>>> annotator = AspectAnotator([spacy_ext, flair_cl])
>>> result = annotator.annotate("This book is really nice!")

... [AspectAnnotation(text='This book is really nice!',
..   aspects=[SentimentAnnotation(text='book', label='positive', score=0.9575109481811523)])]
```

https://github.com/bartoszrozek/pysent



pysent

Search docs

CONTENTS:

Annotators
Overall Annotators
Extractors
Classifiers
Extrassifiers
Data structures

Welcome to pysent's documentation! View page source

Welcome to pysent's documentation!

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.

Extractors

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.

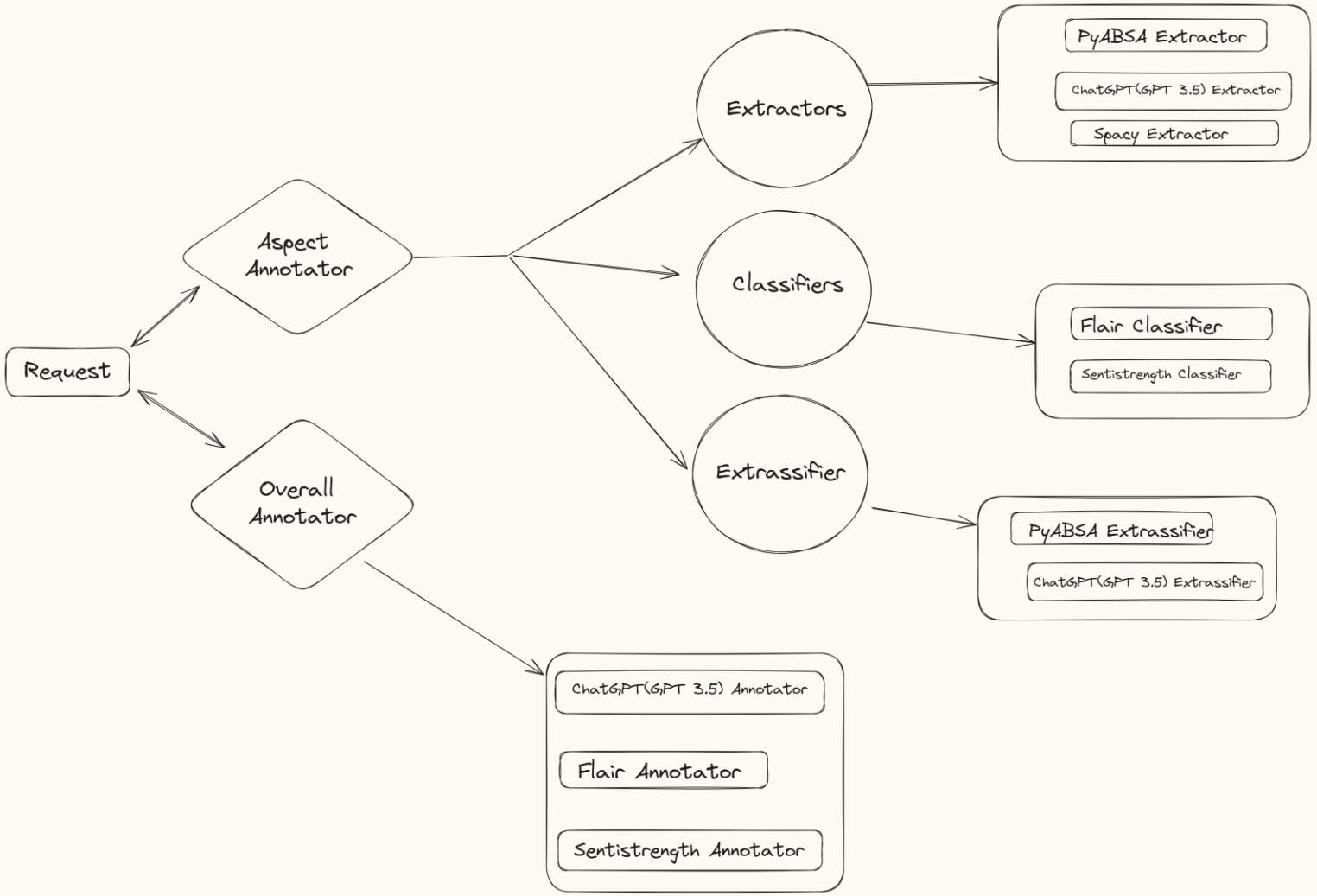
Classifiers

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 Sentistrenght and Python package sentistrength, the aspects is taken from the extractor, but the context is extracted by Sentistrength itself.

https://storage.googleapis.com/nlp_bucket420/html/index.html?fbclid=IwAR0AyYC11tef7D4MWJXs5mKOetqCiekaooKSrjSgS2Egslzwie6L3SsRb9w#

Architecture and Flow



https://github.com/bartoszrozek/pysent

EXAMPLE USAGE

Example use (overall annotation):

```
>>> from pysent import OverallAnotator
>>> from pysent.overall_annotators import FlairAnnotator
>>> flair_an = FlairAnnotator()
>>> annotator = OverallAnotator(flair_an)
>>> annotation = annotator.annotate("This book is really nice!")
>>> print(annotation)

... [SentimentAnnotation(text='This book is really nice!',
...  label='positive', score=0.9575109481811523)]
```

Example use (aspect based annotation):

```
>>> from pysent import AspectAnotator
>>> import pysent.aspect_annotators.extractors as extractors
>>> import pysent.aspect_annotators.classifiers as classifiers

>>> spacy_ext = extractors.SpacyExtractor()
>>> flair_cl = classifiers.FlairClassifier()
>>> annotator = AspectAnotator([spacy_ext, flair_cl])
>>> result = annotator.annotate("This book is really nice!")

... [AspectAnnotation(text='This book is really nice!',
..   aspects=[SentimentAnnotation(text='book', label='positive', score=0.9575109481811523)])]
```

https://github.com/bartoszrozek/pysent

METRICS

MEASURE SENTIMENT MODELS PERFORMANCE

METRICS

FOR SENTIMENT ANALYSIS

- Accuracy/Global accuracy
- Precision
- Recall
- F1-score

Micro approach

All results are aggregated on the level of every element of the confusion matrix (analogously for precision, recall, F1-score). For example, for precision:

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

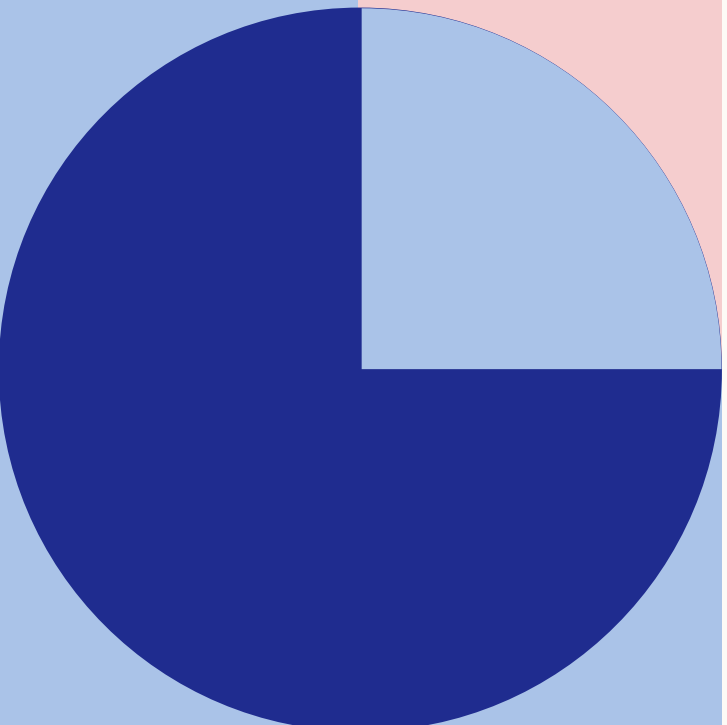
Macro approach

All results are aggregated as weighted mean on statistics calculated for each class (for all, precision, recall, F1-score). For example, for precision:

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

"CONFUSION MATRIX"

FOR ASPECT BASED SENTIMENT ANALYSIS



- Correct (COR): if the observation and its label is the same as the gold-standard annotation
- Incorrect (INC): if the observation is the same as the gold-standard annotation, but has incorrect label
- Partial (PAR): if the observation partially overlaps the gold-standard annotation and has correct label
- Missing (MIS): if a gold-standard annotation does not occur in result dataset
- Spurius (SPU): if the observation does not occur in the gold-standard annotation
- This further translates to:
- Possible (POS): POS = COR + INC + PAR + MIS = TP + FN
- Actual (ACT): ACT = COR + INC + PAR + SPU = TP + FP

METRICS

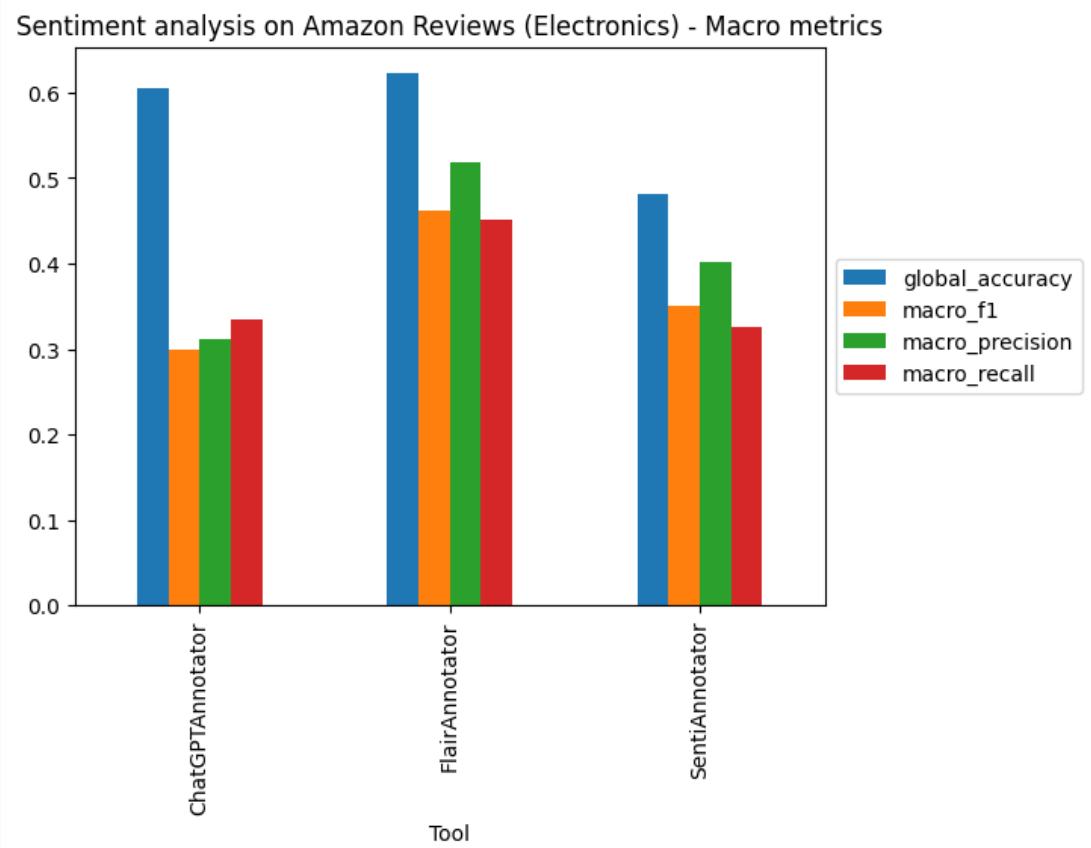
FOR ASPECT-BASED SENTIMENT ANALYSIS

- Precision: $$Precision = \frac{COR}{ACT} = \frac{TP}{TP + FP}$$
- Recall: $$Recall = \frac{COR}{POS} = \frac{TP}{TP + FN}$$
- F1-score: $$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

RESULTS

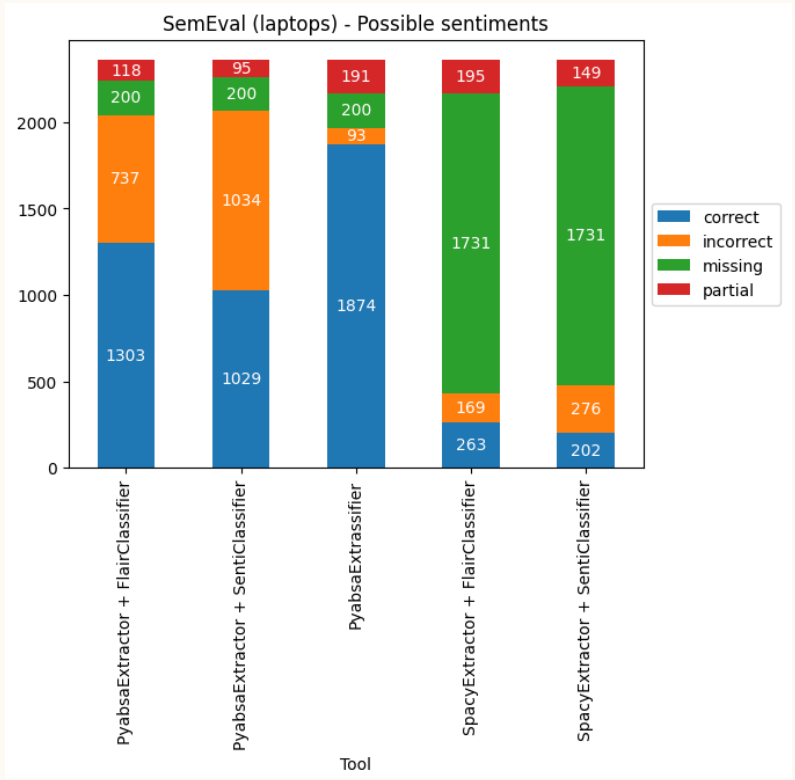
OVERALL SENTIMENT

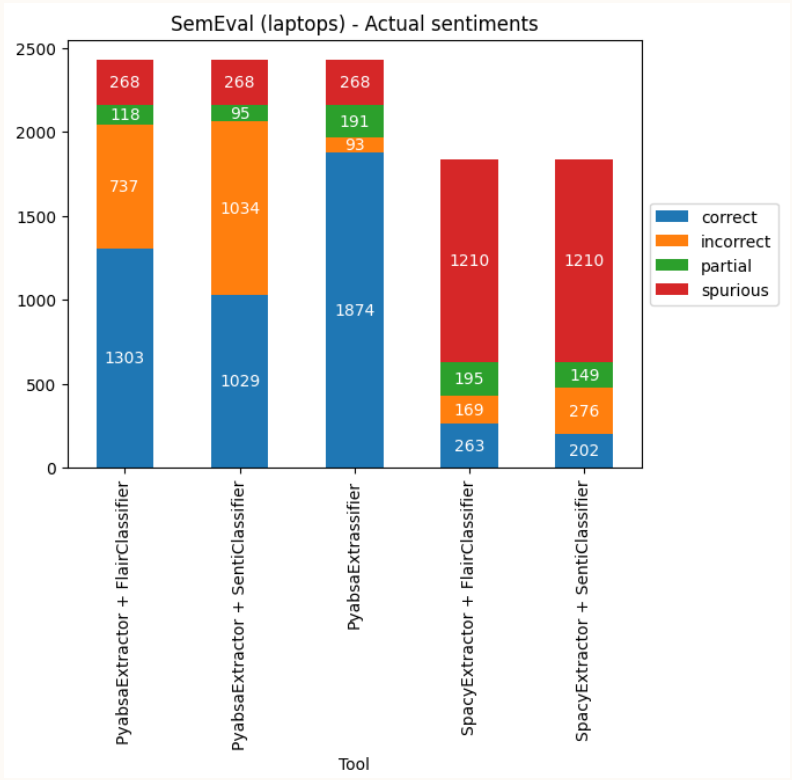
AGGREGATED



ABSA RESULTS

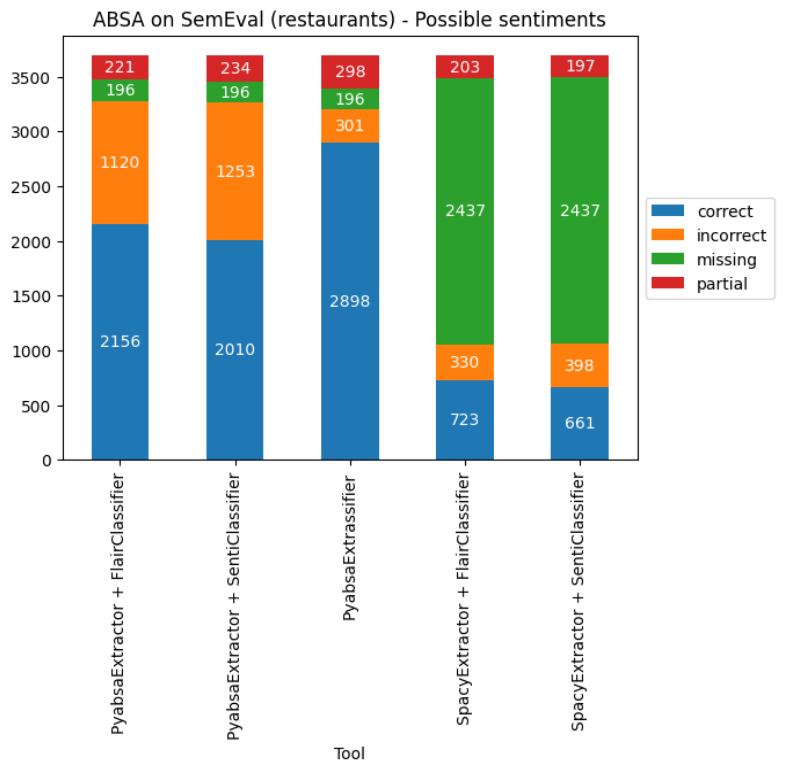
LOW LEVEL

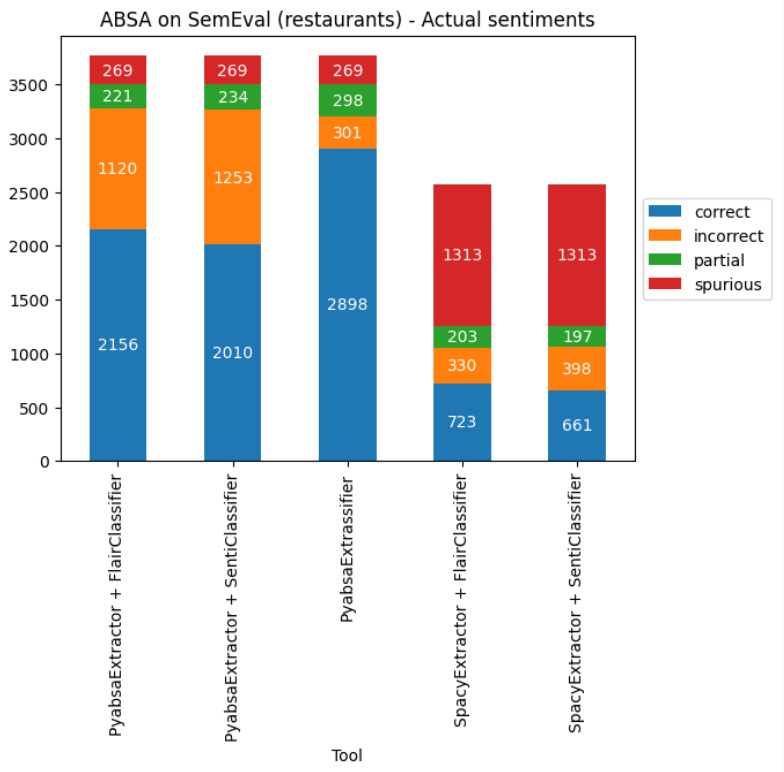




ABSA RESULTS

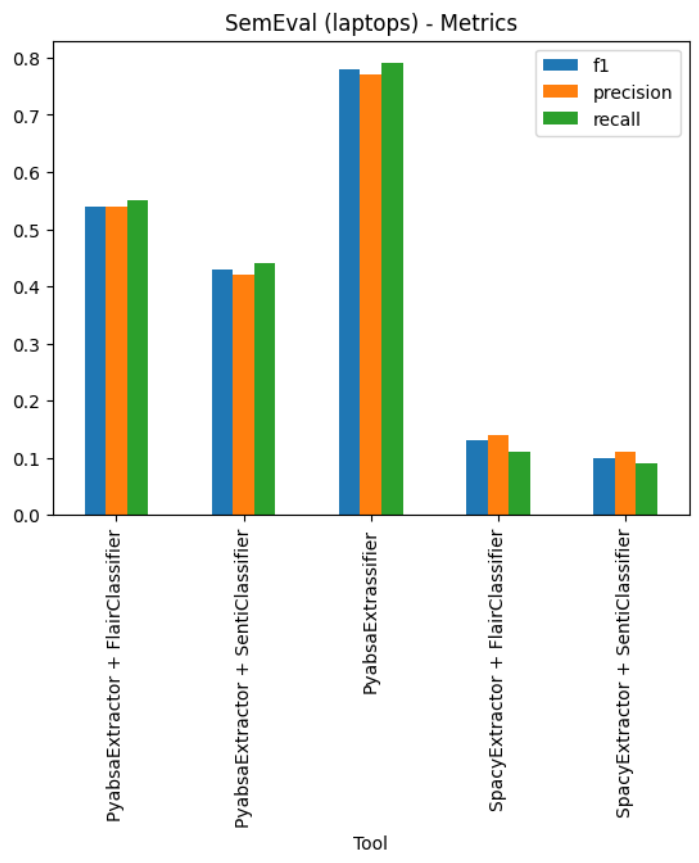
LOW LEVEL

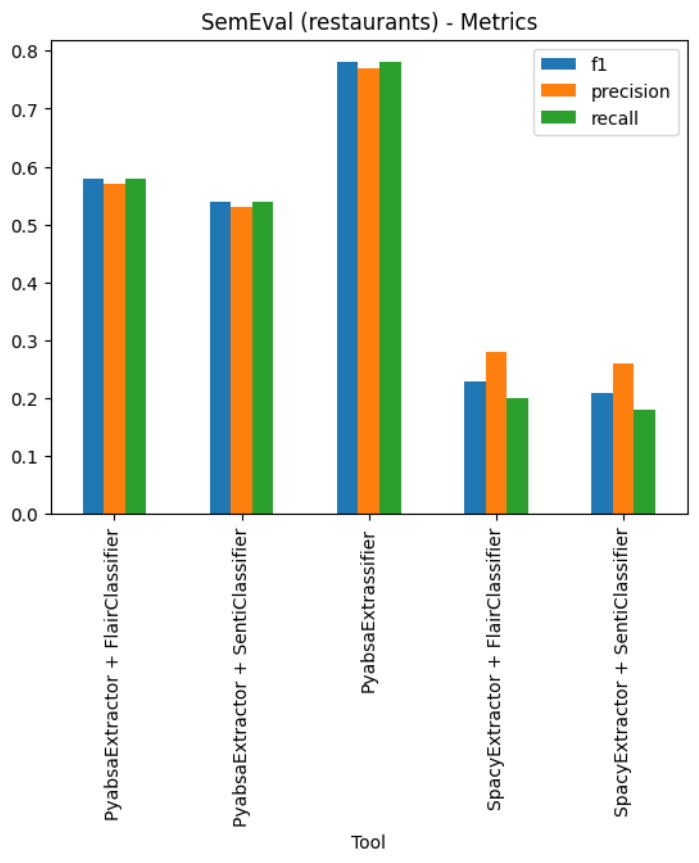




ABSA RESULTS

AGGREGATED





POLISH DATASET

992 reviews -> 2489 aspects

Review	Aspect	Label
Hotel bardzo dobrze położony . Pokój przestronny , ładnie urządzony . Posiłki dobre , ale bez rewelacji . **WiFi** właściwie nie działa . Duży aquapark to na pewno atrakcja . Dopłata za korzystanie z lodówki w pokoju , brak chusteczek higienicznych w łazience . Zepsutą suszarkę w łazience wymieniono dopiero następnego dnia . Ogólnie ok , ale szału nie ma .	WiFi	negative
Zdecydowanie odradzam . **Pokoje** w rzeczywistosci nie mają nic wspólnego z tym co widzimy na stronie hotelu . Ja trafilam na malą ciemna klitke typu hotel pracowniczy * * .	Pokoje	negative
Wybraliśmy z mężem pokój dwuosobowy z podwójnym łóżkiem , oprócz tego że podwójne łóżko polegało na połączeniu dwóch pojedynczych łóżek to więcej nie mamy uwag . W recepcji mile **Panie** , hotel nie jest nowy , ale ma swój klimacik . Jeśli ktoś nie wymaga nowego wykończenia wnętrz to polecam :)	Panie	neutral
Sam **zamek** SUPER , nedza i rozpacz dotyczy obsługi klienta w , , PSELDO " recepcji hotelowej , brak profesjonalizmu obsługi gościa hotelowego . Oferta niezgodna z rezerwacją , Pani DYREKTOR chyba zwraca zbyt malą uwagę na jakość usług . Nie zawsze ilość jest najwazniejsza . Polecam HOTEL ANDERS w STARYCH JABLONKACH , tej samej Grupy , tam zawsze wszystko jest OK .	zamek	positive
Sam zamek SUPER , nedza i rozpacz dotyczy **obsługi klienta** w , , PSELDO " recepcji hotelowej , brak profesjonalizmu obsługi gościa hotelowego . Oferta niezgodna z rezerwacją , Pani DYREKTOR chyba zwraca zbyt malą uwagę na jakość usług . Nie zawsze ilość jest najwazniejsza . Polecam HOTEL ANDERS w STARYCH JABLONKACH , tej samej Grupy , tam zawsze wszystko jest OK .	obsługi klienta	negative

IDEAS FOR PROJECT 2

- creating tool that allows non-it people to correct annotations -> lower costs to create gold standard data set

- using different tools

- additional training for pyabsa

- speed up by parallel computing

LITERATURE

- Massively Multilingual Corpus of Sentiment Datasets and Multi-faceted Sentiment Classification Benchmark (Ł. Augustyniak, Sz. Woźniak, M. Gruza, P. Gramacki, K. Rajda, M. Morzy, T. Kajdanowicz)
- Erick Cambria works on sentiment analysis and affective computing (https://scholar.google.com/citations?hl=en&user=ilSYpW0AAAAJ&view_op=list_works&sortby=pubdate)
- SentiStrength (http://sentistrength.wlv.ac.uk/)
- NLP Progress (http://nlpprogress.com/english/sentiment_analysis.html)
- Effective Seed-Guided Topic Discovery by Integrating Multiple Types of Contexts (Y. Zhang, Y. Zhang, M. Michalski, Y. Jiang, Y. Meng, J. Han)
- Amazon reviews (https://arxiv.org/abs/2212.06002)
- Izabela Telejko, BSc thesis
- Generating Explainable Product Comparisons for Online Shopping. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (N. Vedula, M. Collins, E. Agichtein, O. Rokhlenko)

THANK YOU

Q&A