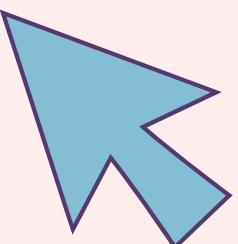
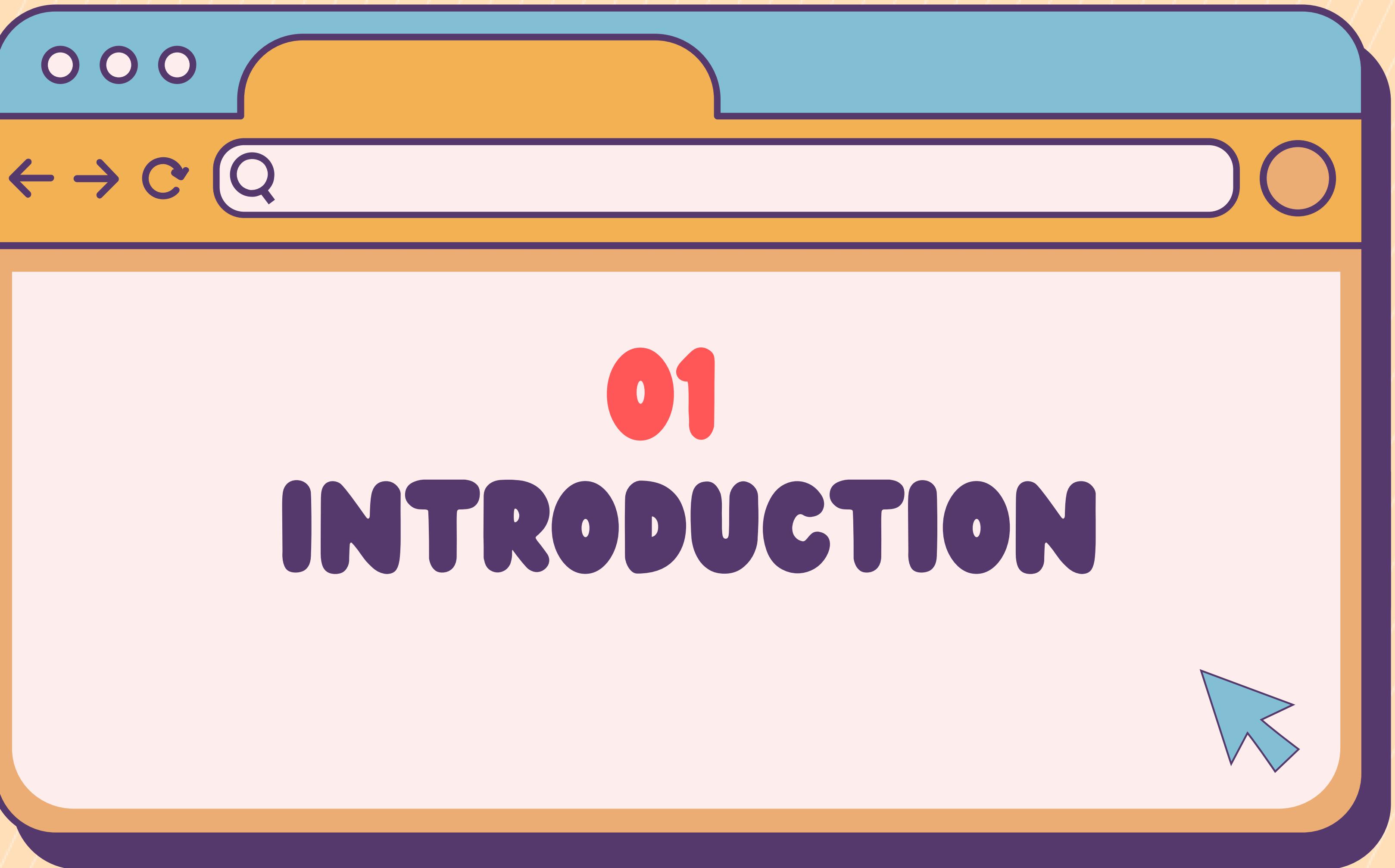


BIDA ANG SAYA! :

**IMPLEMENTING A LEXICON-BASED
TAGLISH SENTIMENT ANALYSIS
THROUGH FINITE STATE MACHINES**

Presented by:
Ang, Bolipata, Zialcita
CSCI115 O - Final Report





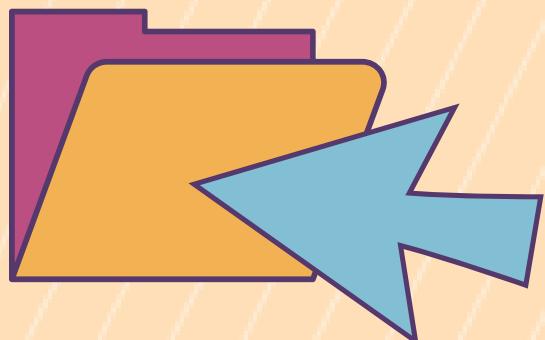
TAGALOG SENTIMENT ANALYSIS



- Filipino culture of sharing sentiments on online platforms
- Use of Tagalog SMS Language to convey sentiments
- A local sentiment analysis model that will analyze texts written by Filipinos



BRIEF DEFINITION



SENTIMENT ANALYSIS



- The process of computationally identifying and categorizing opinions as positive or negative
- Can process **high volumes of raw text** online when automated through a computer model

SIGNIFICANCE

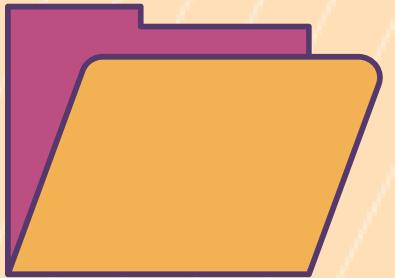
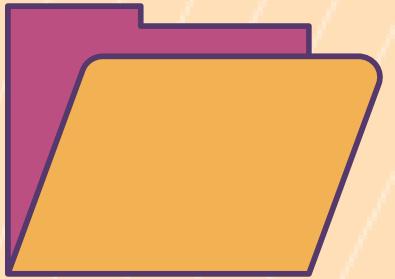


CURRENT MODELS



- Current studies largely focus on English sentiment analysis
- Existing models for Tagalog sentiment analysis are simplified; they do not account for slang or informal language in text
- Existing models use machine learning algorithms

SIGNIFICANCE



OUR PROJECT

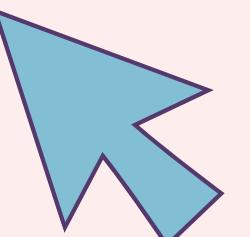


- Our model will use a **finite state automata**, and recognize Tagalog SMS language and Taglish and categorize sentiments accordingly
- Create an efficient system that will analyze the massive influx of data found all over the Filipino mediascape
 - Analyze sentiments of Filipinos communicated in a casual and common language

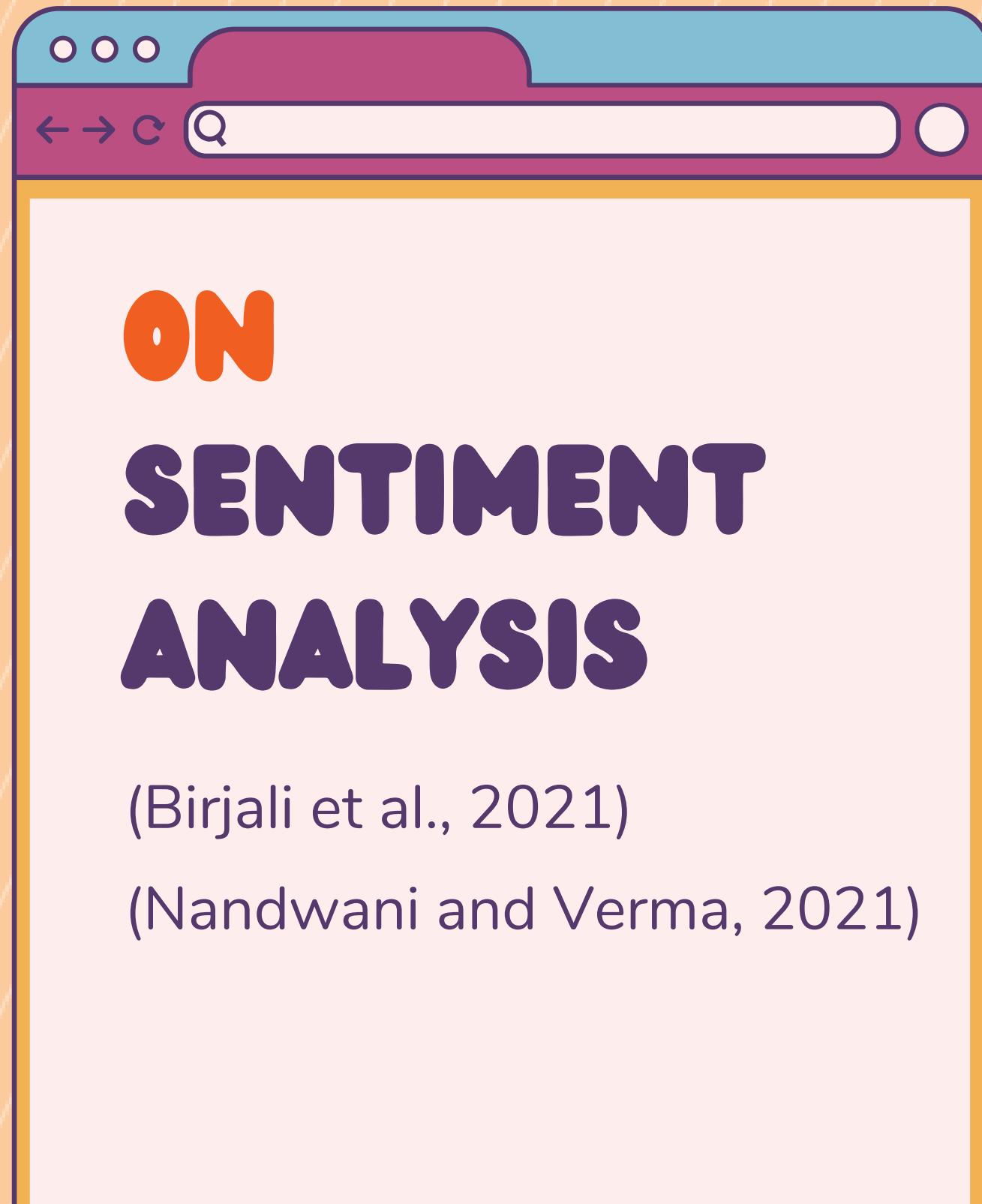


02

REVIEW OF RELATED LITERATURE



TYPES OF SENTIMENT ANALYSIS



Sentence level

- Text is broken down into **sentences**. Polarity of each sentence is identified

Document level

- Sentiment is detected from the entire document. Relies on **context** of the word

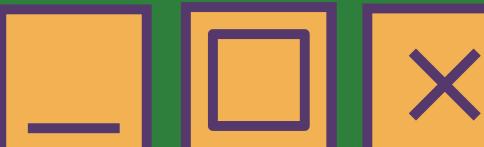
Aspect level

- Identify the **aspects** of a service in the text.
Assign sentiment accordingly.
- Ex. food <- yummy (+). price <- expensive (-)



Assigning of sentiment can be through:

- a value from -1 to +1
- fixed **categorization** (positive/negative)
- ranking sets (strong positive, positive, neutral etc.)



Categories of Lexicons

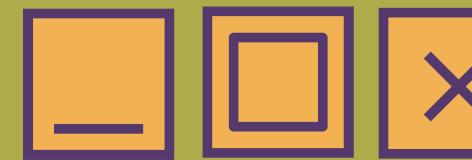
1. **Sentiment:** beautiful, ugly
2. **Intensifier:** very, highly
3. **Negation:** not, doesn't
4. **Adjective Sense Disambiguation:** cheap, mura
5. **Phrase and Idiom:** does its job
6. **Syntax Pattern:** can't recommend it enough

ON LEXICON AND SENTIMENT VALUES

(Eng et al., 2021)

(Kaity and Balakrishnan, 2020)

(Andrei, 2014)



Defining of Overall Text Sentiment:

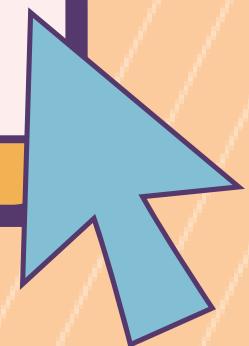
- **Positive**
 - Only neutral and positive sentiments
- **Negative**
 - Only neutral and negative sentiments
- **Conflicting**
 - Existence of positive and negative sentiments
- **Neutral**
 - No positive or negative sentiment

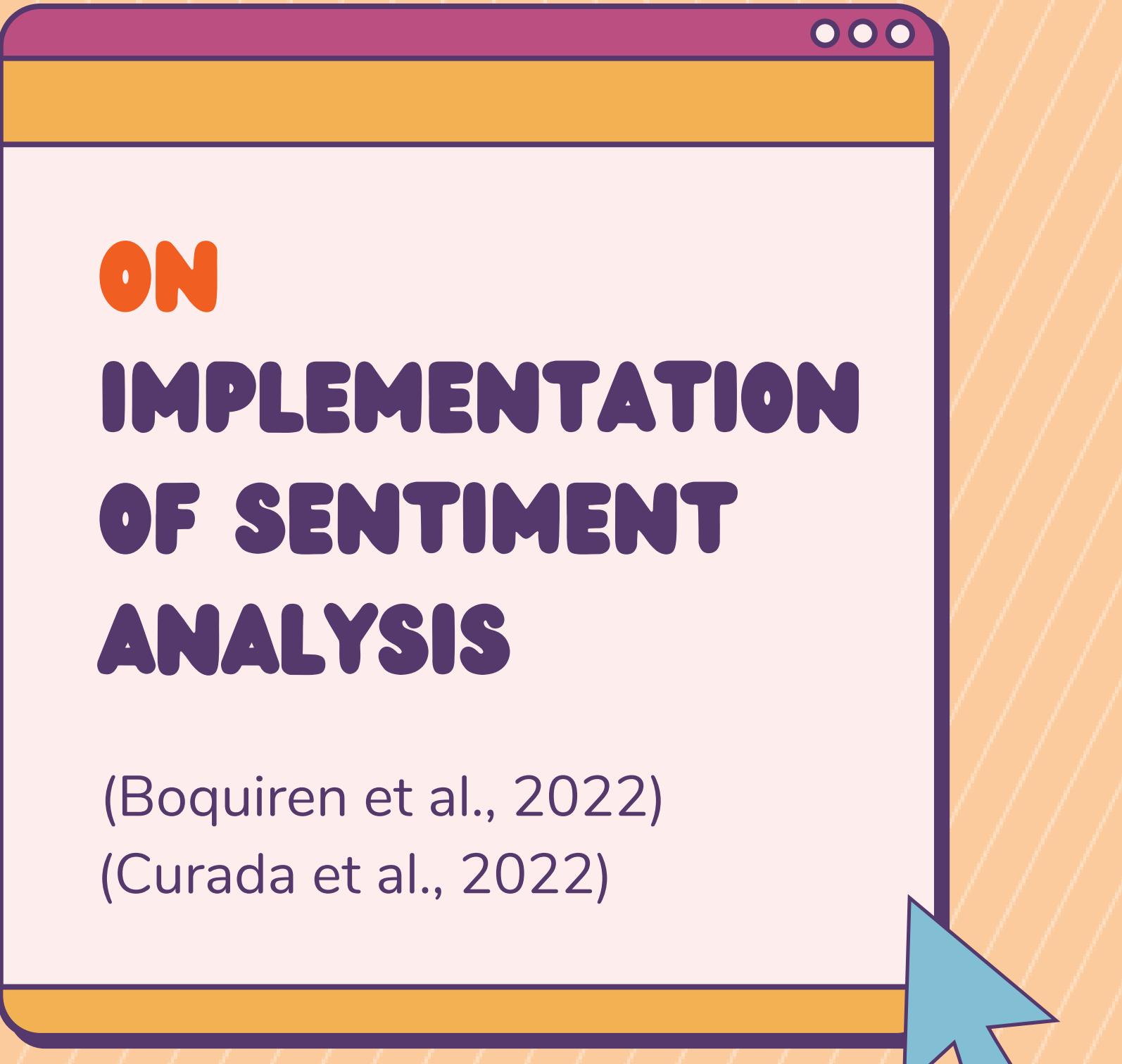
ON EXISTING LEXICON-BASED APPROACHES

(Pelosi et al., 2019)

(Contreras et al., 2018)

(Castaneda et al., 2018)





ON IMPLEMENTATION OF SENTIMENT ANALYSIS

(Boquiren et al., 2022)

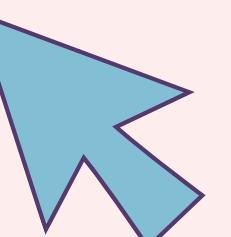
(Curada et al., 2022)

- 
- 1. Input Module**
 - a. Importing of text and lexicons
 - 2. Preprocessing Module**
 - a. Converting uppercase to lowercase
 - b. Removal of punctuations
 - 3. Processing Module**
 - a. Machine learning algorithm/similar procedures
 - 4. Output Module**
 - a. Analysis, testing

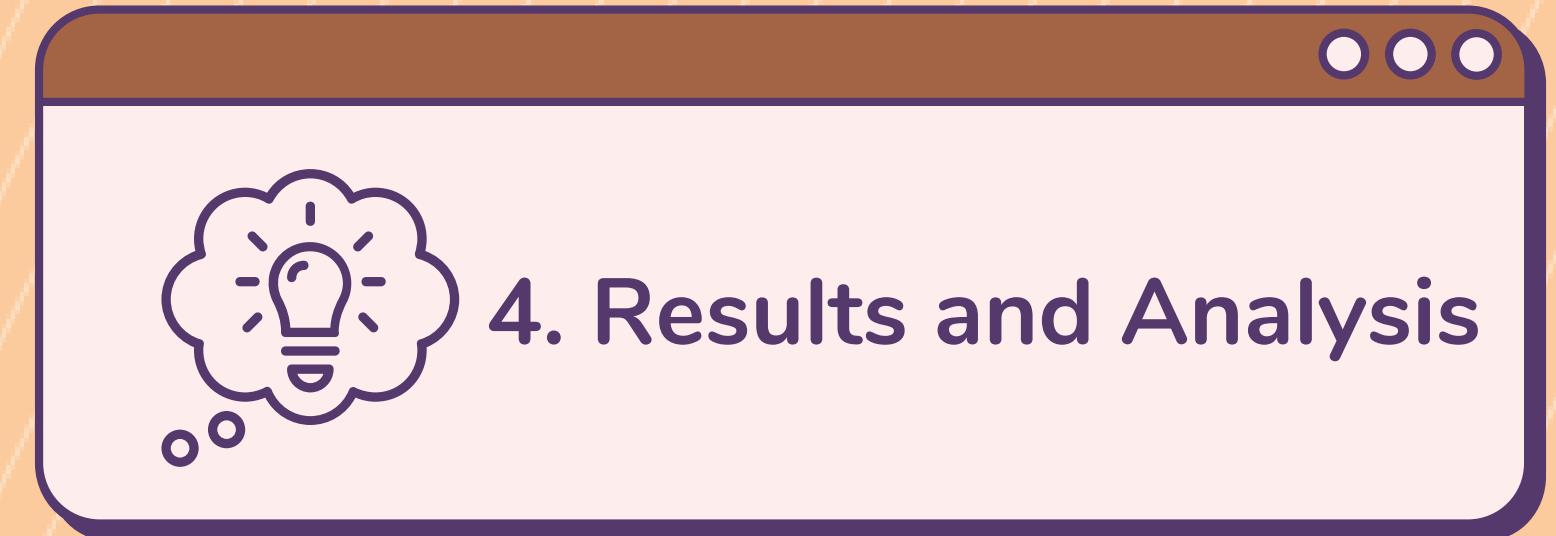
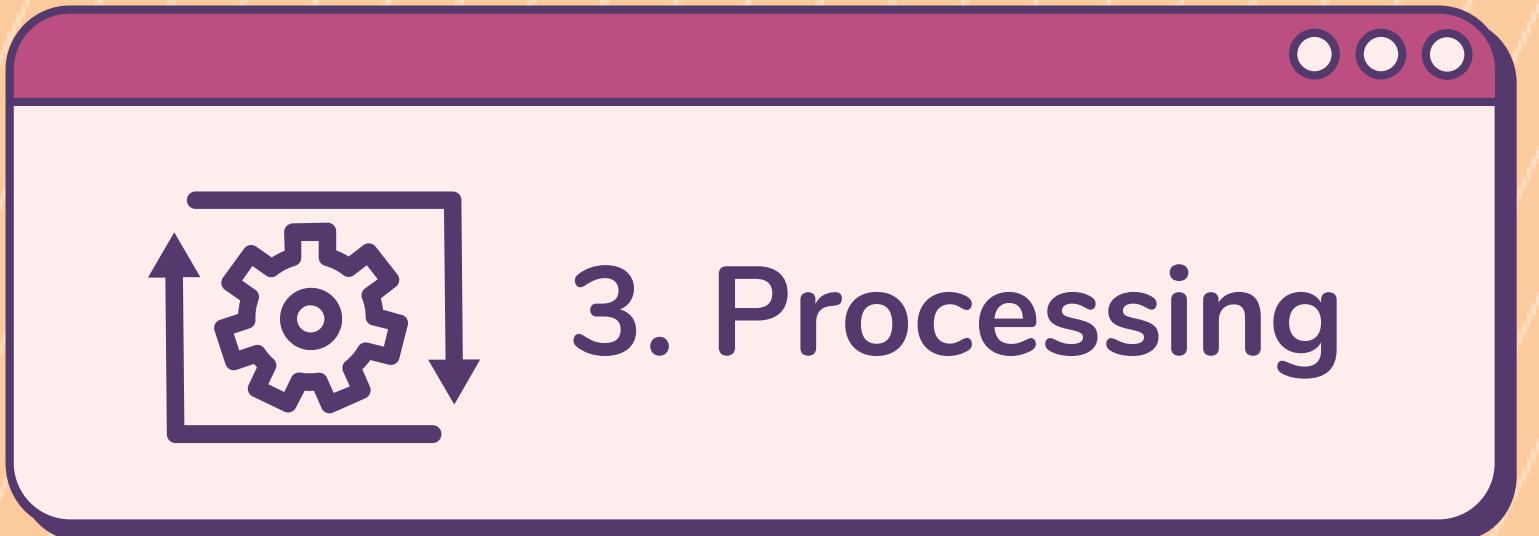
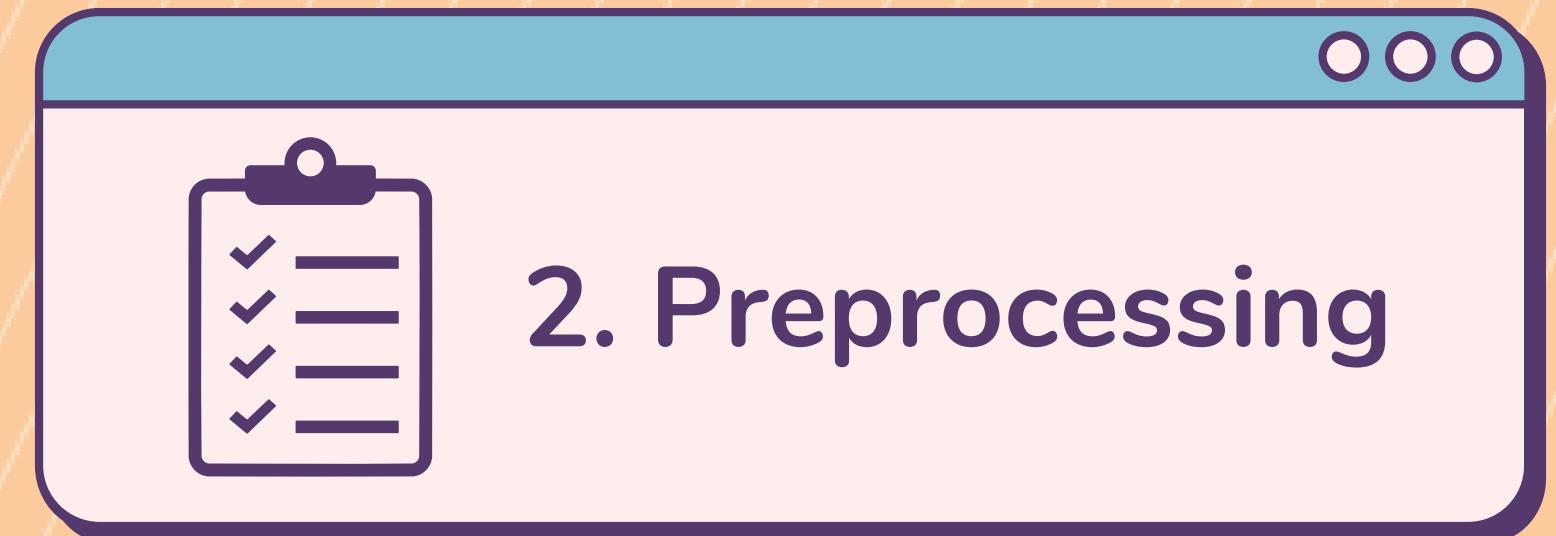


03

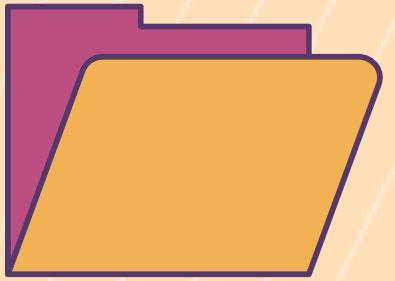
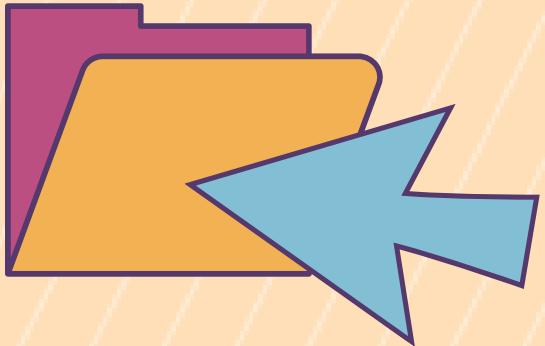
SIMULATION AND MODELLING



FOUR-STAGE METHODOLOGY



STAGE 1: INPUT



RAW DATA SETS



Existing sentiment lexicons mapping words to sentiments

- **AFINN** wordset for English
- **Kaggle** wordset for Tagalog

Used two review datasets to determine the accuracy of our model

- **Amazon** review dataset
- **Lazada** review dataset

The files are either .csv or .txt files

STAGE 2: PREPROCESSING



SAMPLE RAW LEXICONS



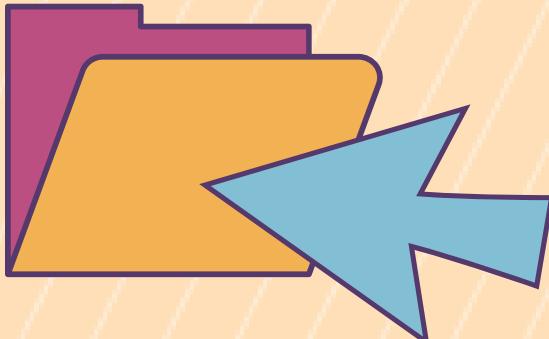
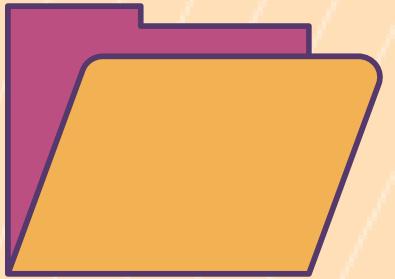
ENGLISH

term	value
regret	-2
indifferent	-2

TAGLISH

word	definition	dialect	sentiment
masira	masira (nasisira, tagalog		negative
maganda	maganda adj. be tagalog		positive
pangit	adj. 1. ugly (to lo tagalog		negative

STAGE 2: PREPROCESSING



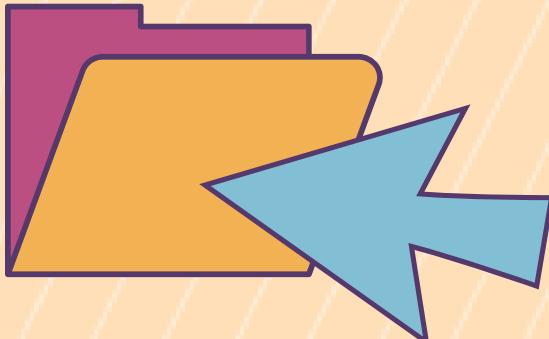
SORTING AND FILTERING



- English lexicons were categorized based on their sentiment values.
- Taglish lexicons were filtered directly and categorized into:

POSITIVE NEGATIVE NEUTRAL

STAGE 2: PREPROCESSING

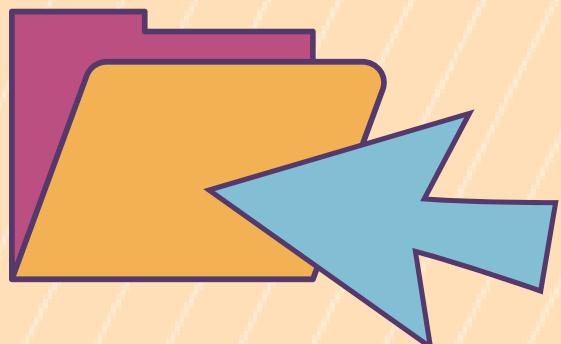


SORTING AND FILTERING



- The project also identified the following **lexicons**, and were **stored in their respective lists**:
 - negation words: not
 - positive phrase lexicons: does its job
 - negative phrase lexicons: no excuse
 - neutral phrase lexicons: ok lang
 - dual words: cheap, mura
- This was done to address the issue on **Phrase Lexicons**, and **Adjective Sense Disambiguation**

STAGE 2: PREPROCESSING



SAMPLE REVIEWS



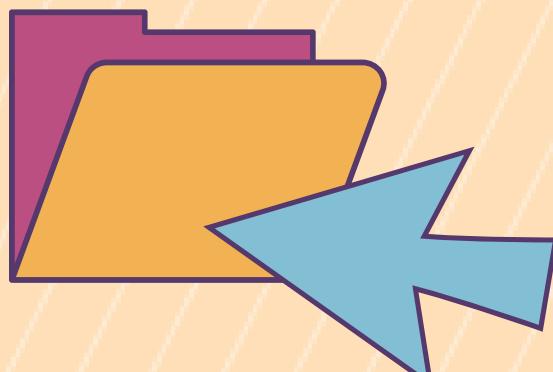
ENGLISH

Works just fine in my Galaxy S4. No problem s. Unbelievable how much storage you can get in such a small space and for such a low price.

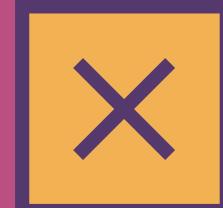
TAGLISH

Good for beginners, its not that dark halos kakumay lang siya ng skin care ko. I'm actually expecting it to be darker pero keri na.

STAGE 3: PROCESSING



SIMULATION OVERVIEW



ENGLISH

250 reviews

Used the English Lexicons only

TAGLISH

250 reviews

Used the English and Tagalog Lexicons together

The lexicons were used as the alphabet for the FSA

STAGE 3: PROCESSING



THE FINITE STATE AUTOMATON



Q : the current model has 7 states

q_0 : neutral

q_1 : positive

q_2 : negative

q_3 : conflicting

q_4 : negation

q_5 : negation preceded by positive

q_6 : negation preceded by negative

Σ : 4 entries in the alphabet

$+$: POSITIVE LEXICONS
(GOOD, MAGANDA)

$-$: NEGATION LEXICONS
(NOT, HINDI)

n : NEGATIVE LEXICONS
(BAD, PANGIT)

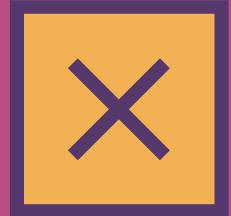
0 : NEUTRAL LEXICONS
(JOLLIBEE, ADOBO)

NOTE THAT THE PHRASE LEXICONS ARE INCLUDED IN THE POS/NEG/NEUT ALPHABET

STAGE 3: PROCESSING



THE FINITE STATE AUTOMATON



q_0 initial state (neutral)

F

q_0	q_1
q_2	q_3

final states
(positive, negative,
neutral, conflicting)

STAGE 3: PROCESSING

```
def sentiAutomata(autoList, autoString):
    states = [0]
    phraseLexicons = posPhraseLexicons + negPhraseLexicons + neutPhraseLexicons

    autoWords = processText(autoString)
```

- The sentiment automata accept 2 arguments:
 - (1) the **list of states of the FSA**, and (2) the **string**
- The string is **processed and transformed into a list of words/phrases** for the FSA through **processText()**

STAGE 3: PROCESSING

processText()

```
def processText(text):
    wordlist = text.translate(
        str.maketrans("!\"#$%&()*+, -./:;=>?@[\\]^_`{|}~",
                      '*'*len("!\"#$%&()*+, -./:;=>?@[\\]^_`{|}~"))).lower().split()
    wordlist = phrasify(wordlist)
    return wordlist
```

- The text is removed of any punctuation **except for the single apostrophe ('')**. This is to consider contractions of words such as **don't**
- The text is then made into a list of words, and the list is checked for phrase lexicons through **phrasify()**

phrasify()

```
def phrasify(list):
    wordlist = []
    phrase = []

    for word in list:
        if word not in lexicons:
            wordlist.append(word)
        elif word in lexicons:
            if len(wordlist) > 0:
                phrase.append(' '.join(wordlist))
            phrase.append(word)
            wordlist = []
        phrase.append(' '.join(wordlist))
    if phrase[-1] == '':
        phrase = phrase[0:-1]
    return phrase
```

- The text is broken down into **phrases** and **words** using the sentiment lexicons.

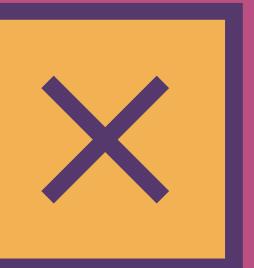
SAMPLE INPUT:

```
processText("The product does its JOB  
WELL, however i really don't like the  
color given to me. I want a refund")
```

SAMPLE OUTPUT:

```
['the product does it does its job',  
'well',  
'however i really',  
"don't",  
'like',  
'the color given to me i',  
'want',  
'a',  
'refund']
```

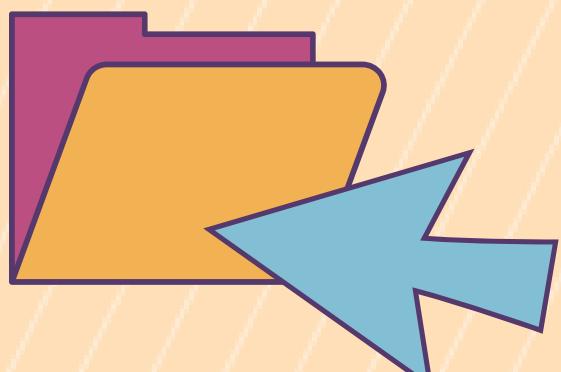
HOW DOES THE FSA WORK?



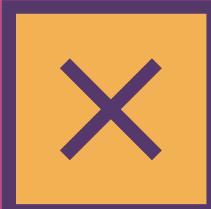
```
#checks if autoWord is a phrase
if len(autoWord.split(' ')) > 1:
    for phrase in phraseLexicons: #checks if the phrase has a predetermined sentiment
        if phrase in autoWord:
            autoWord = phrase
            autoCurrent = autoList[autoCurrent].getNewState(autoWord)
            states.append(autoCurrent)
    else: #if no, the state remains the same
        autoCurrent = autoCurrent
```

- The FSA first checks if the current element is a **phrase**
 - If **yes**, check if it has predetermined sentiment.
 - If **yes**, go to the next state accordingly
 - If **no**, there is no change in state.
 - If **no**, proceed

STAGE 3: PROCESSING

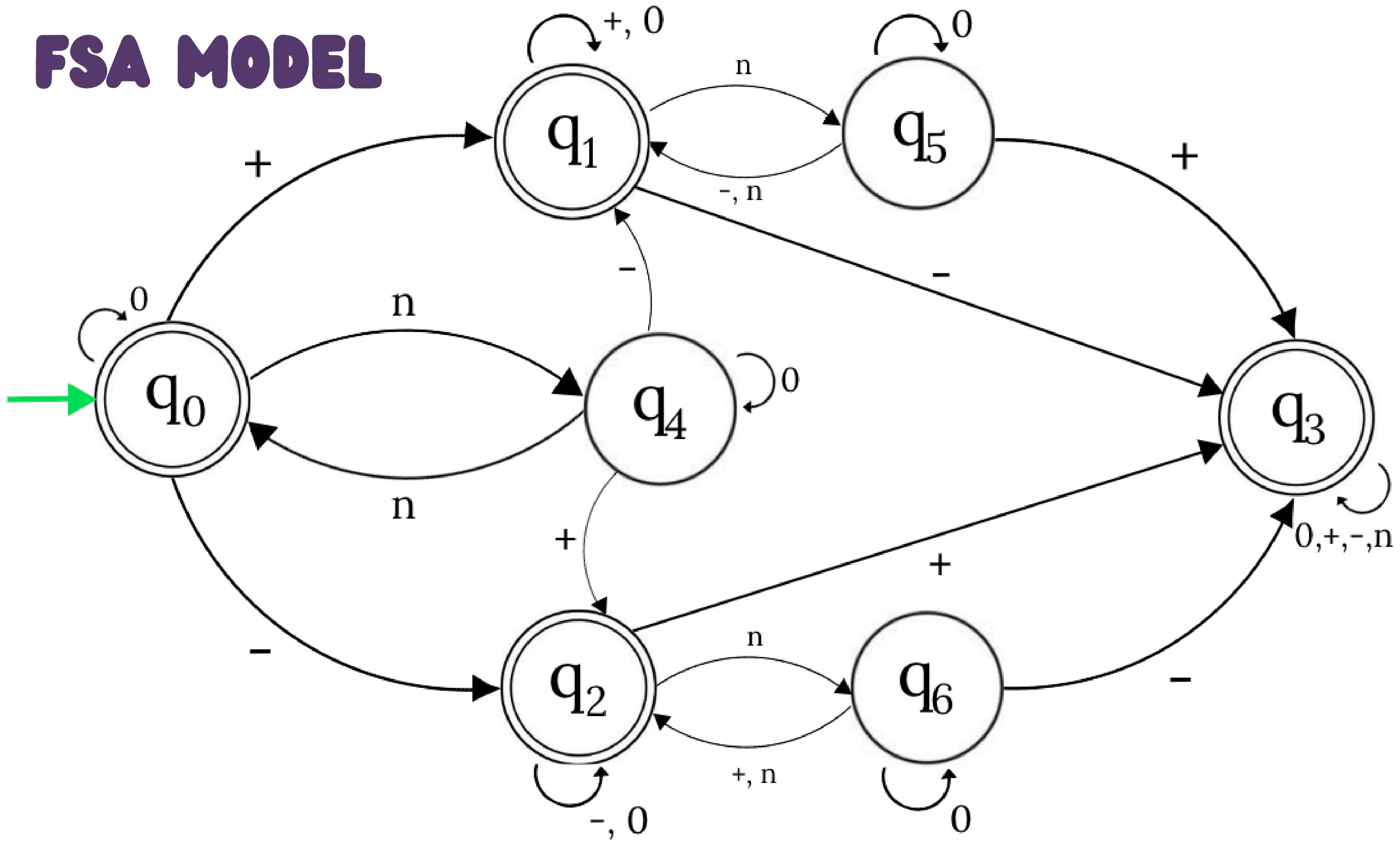


HOW DOES THE FSA WORK?



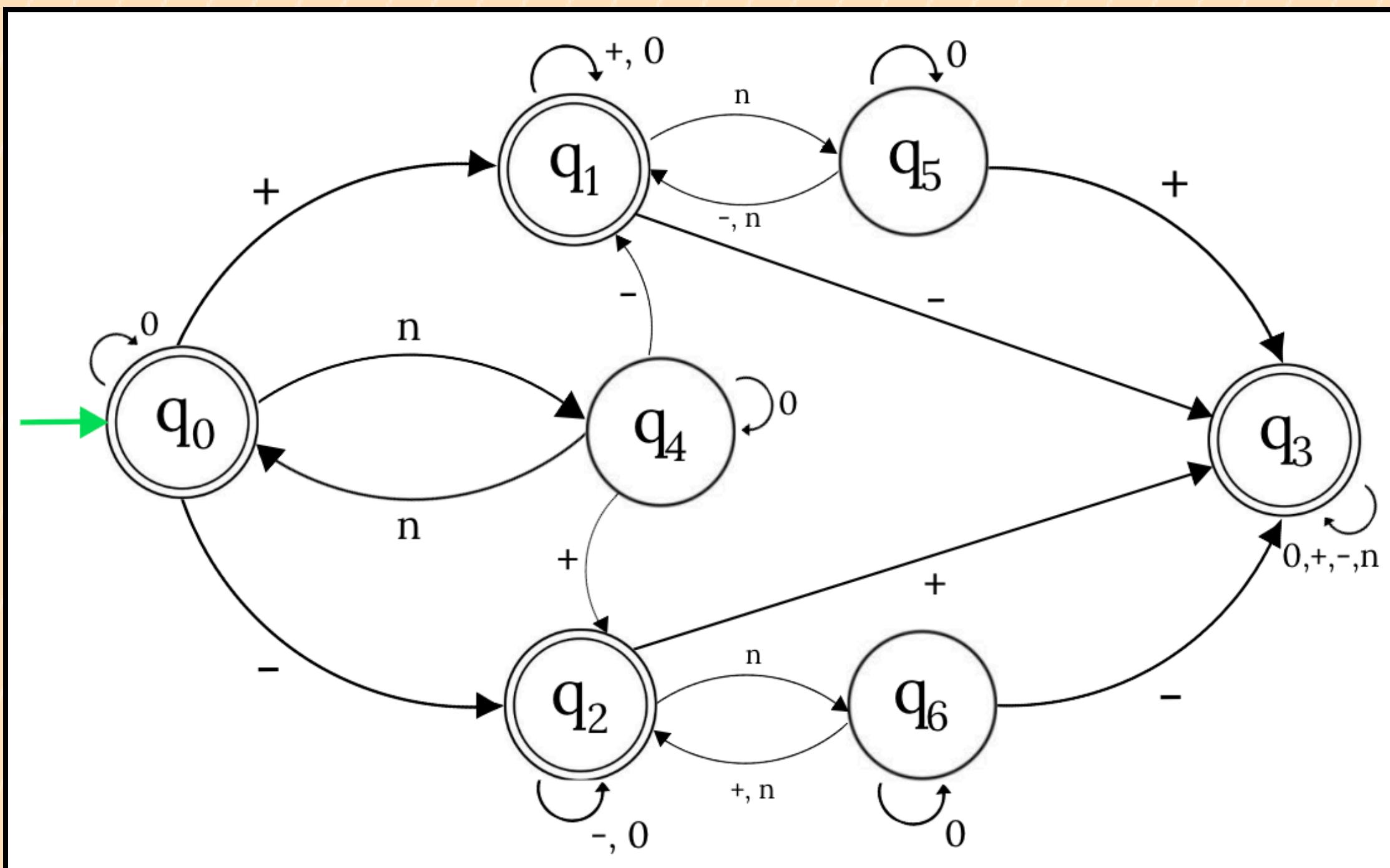
- The FSA then checks if the word is a **positive, negative, or negation lexicon**
 - If **no**, there is no change in state
 - If **yes**, proceed
- Since the word is a **defined lexicon**, transition to the next state accordingly.

FSA MODEL



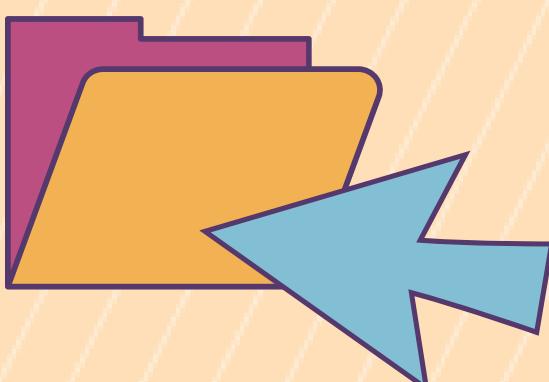
EXAMPLE

Lasted about 8 months of daily use in my Android phone. But yesterday it stopped performing writes and then failed altogether. While it was working, reads were incredibly fast and writes were pretty good as well.



stopped (state 2)
failed (state 2)
working (state 3)
fast (state 3)
good (state 3)
good (state 3)
well (state 3)
Conflicting

STAGE 4: ANALYSIS AND RESULTS



EVALUATION METRICS



- The effectiveness was evaluated based on two metrics:
 - **Accuracy** - total correctness of all the classification
 - **Precision** - accuracy of the classification on each class or category

STAGE 4: ANALYSIS AND RESULTS

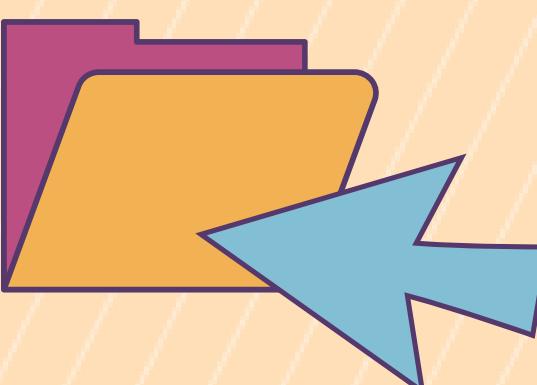
EVALUATION METRICS



$$\text{Accuracy} = \frac{\text{TotalTrueInstances}}{\text{TotalInstances}}$$

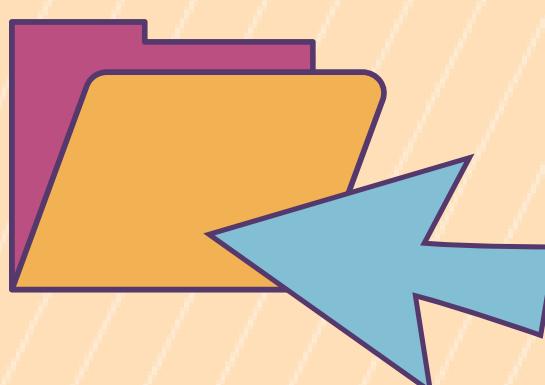
$$\text{Precision} = \frac{\text{TruePositiveInstances}}{\text{TotalPositiveInstances}}$$

(For Positive Sentiments)



Percent (%) Range	Verbal Interpretation
81.00%-100.00%	Highly Effective/ Highly Accurate/ Highly Precise
61.00%-80.00%	Effective/ Accurate/ Precise
41.00%-60.00%	Moderately Effective/ Moderately Accurate/ Moderately Precise
21.00%-40.00%	Slightly Effective/ Slightly Accurate/ Slightly Precise
00.00%-20.00%	Not Effective/ Not Accurate/ Not Precise

STAGE 4: ANALYSIS AND RESULTS

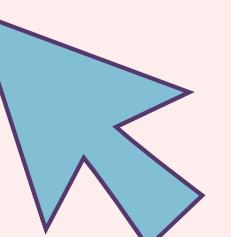


EVALUATION METRICS



Percent (%) Range	Verbal Interpretation
81.00%-100.00%	Highly Effective/ Highly Accurate/ Highly Precise
61.00%-80.00%	Effective/ Accurate/ Precise
41.00%-60.00%	Moderately Effective/ Moderately Accurate/ Moderately Precise
21.00%-40.00%	Slightly Effective/ Slightly Accurate/ Slightly Precise
00.00%-20.00%	Not Effective/ Not Accurate/ Not Precise





IMPLEMENTED ADJUSTMENTS

To increase accuracy and precision



1. Lexicons changes

- a. Adding words to the lexicons
- b. Adding phrase lexicons (positive and negative)
- c. Considering words with dual meanings from the lexicons

2. Function to separate and evaluate sentences

3. Implementing all effective adjustments

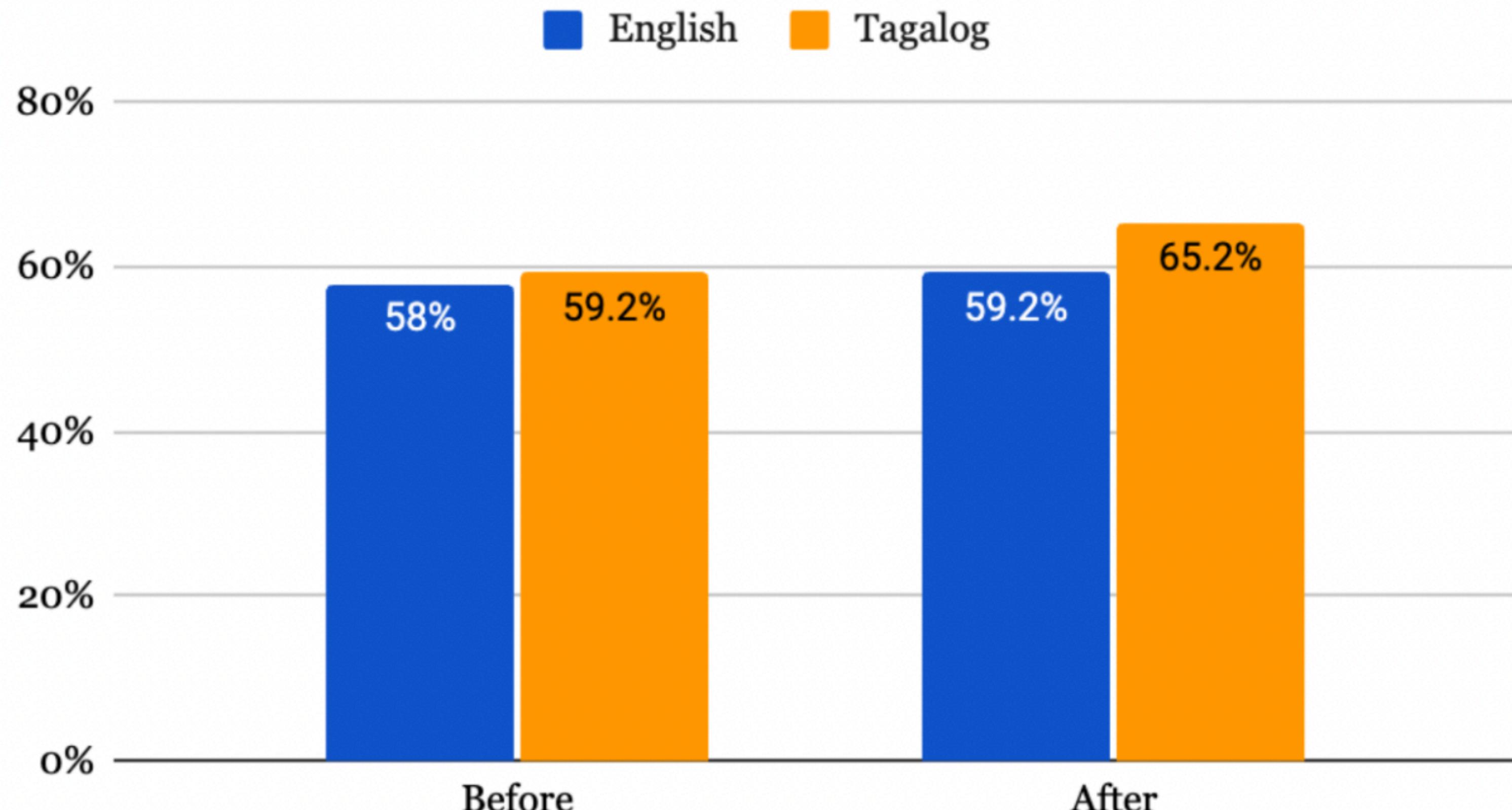
(1) LEXICON CHANGES

Adding words to the lexicons

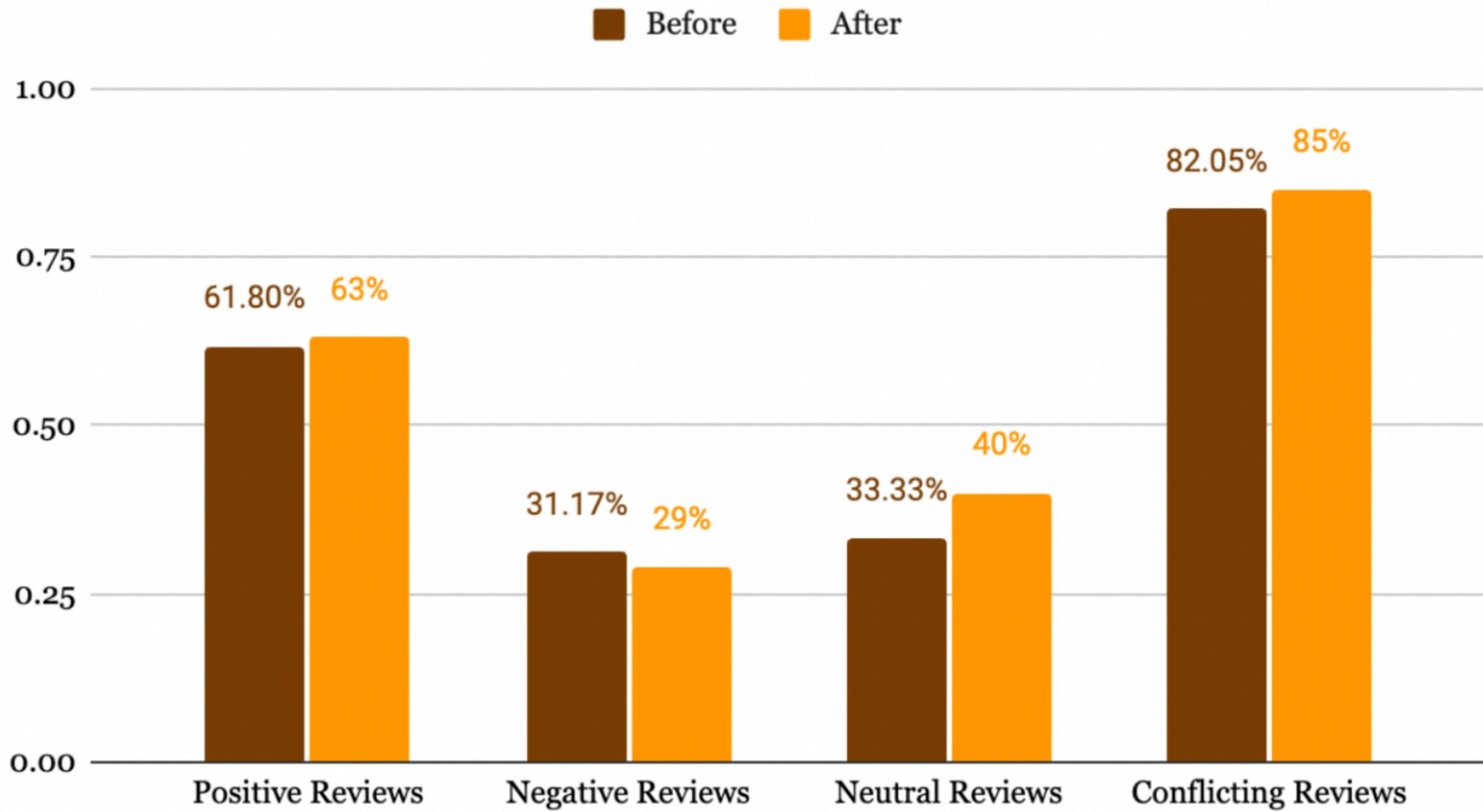


- Adding more positive and negative words to their respective lexicons
- Reclassifying wrongly assigned lexicons from the database
- Was done for English and Tagalog lexicons

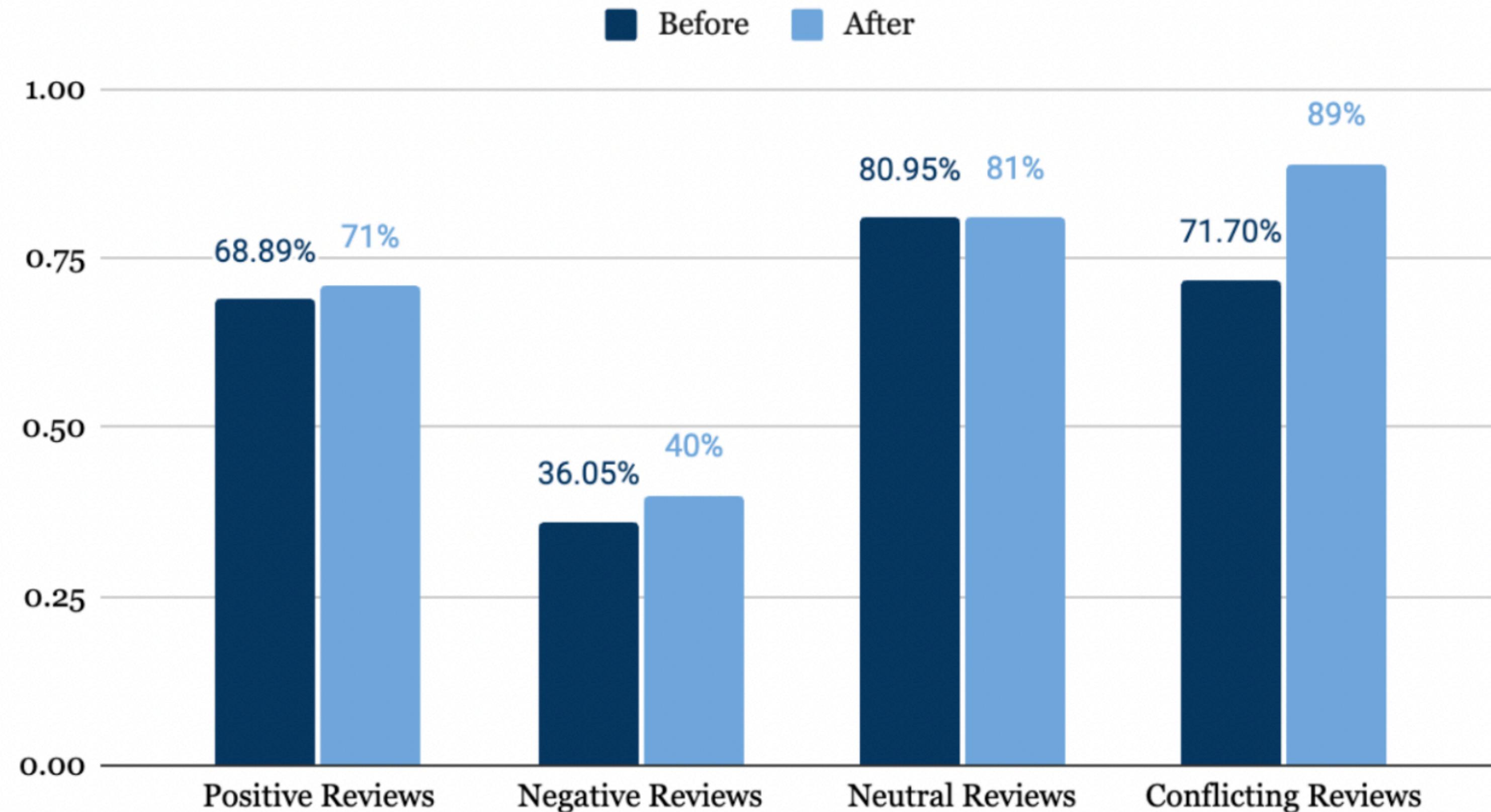
Adding words to the lexicons: Accuracy before and after



Adding words to the lexicons: Precision for English reviews before and after

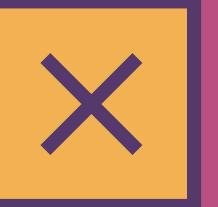


Adding words to the lexicons: Precision for Tagalog reviews before and after



(1) LEXICON CHANGES

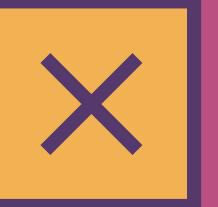
Adding words to the lexicons



- Increased precision for all English reviews but negative reviews
 - Negative reviews tend to use positive lexicons as well, in for example, comparing a bad product to a good product,
 - As such, the reviews are categorized as conflicting.

(1) LEXICON CHANGES

Adding words to the lexicons



Why it works?

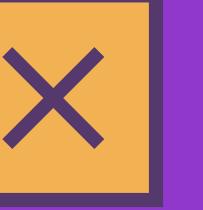
More words are recognized as positive or negative
→ fewer words categorized as neutral
→ sentiments are less likely to be misidentified as neutral or conflicting

(1) LEXICON CHANGES

Creating positive and negative phrase lexicons 

- Phrases with positive or negative meanings that comprise words with no positive or negative connotation
- Sample phrases
 - Positive phrases: “Does its job”, “Five stars”, “Served its purpose”
 - Negative phrases: “Not for me”, “Let me down”, “Hit or miss”

(1) LEXICON CHANGES

Creating positive and negative phrase lexicons 

Process

- Text is split into sentiment words and phrases made up of neutral words using the **phrasify** function
- Each phrase is checked against positive, negative, and neutral phrase lexicons

phrasify()

```
def phrasify(list):
    wordlist = []
    phrase = []

    for word in list:
        if word not in lexicons:
            wordlist.append(word)
        elif word in lexicons:
            if len(wordlist) > 0:
                phrase.append(' '.join(wordlist))
            phrase.append(word)
            wordlist = []
        phrase.append(' '.join(wordlist))
    if phrase[-1] == '':
        phrase = phrase[0:-1]
    return phrase
```

- The text is broken down into **phrases** and **words** using the sentiment lexicons.

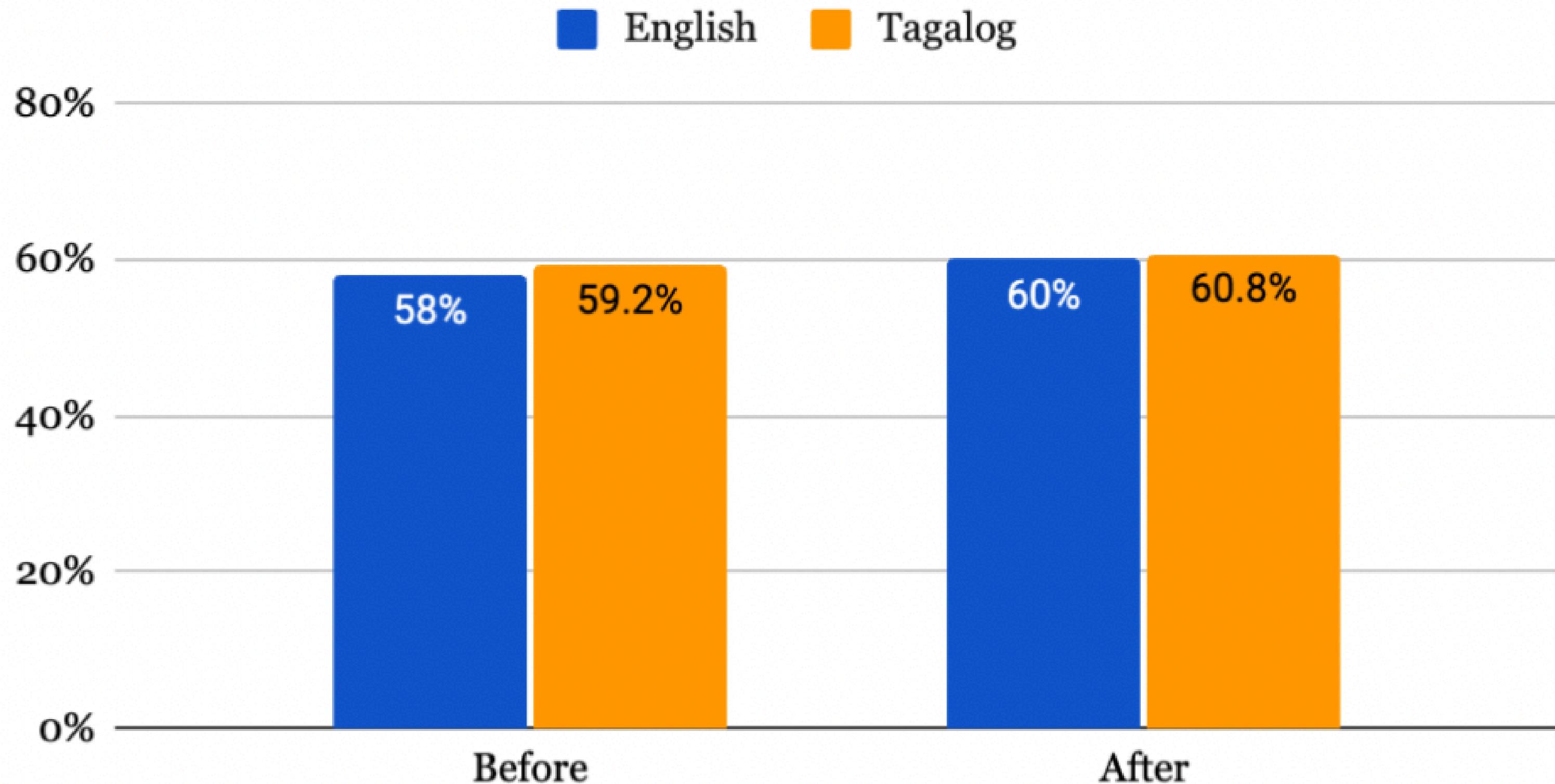
SAMPLE INPUT:

```
processText("The product does its JOB  
WELL, however i really don't like the  
color given to me. I want a refund")
```

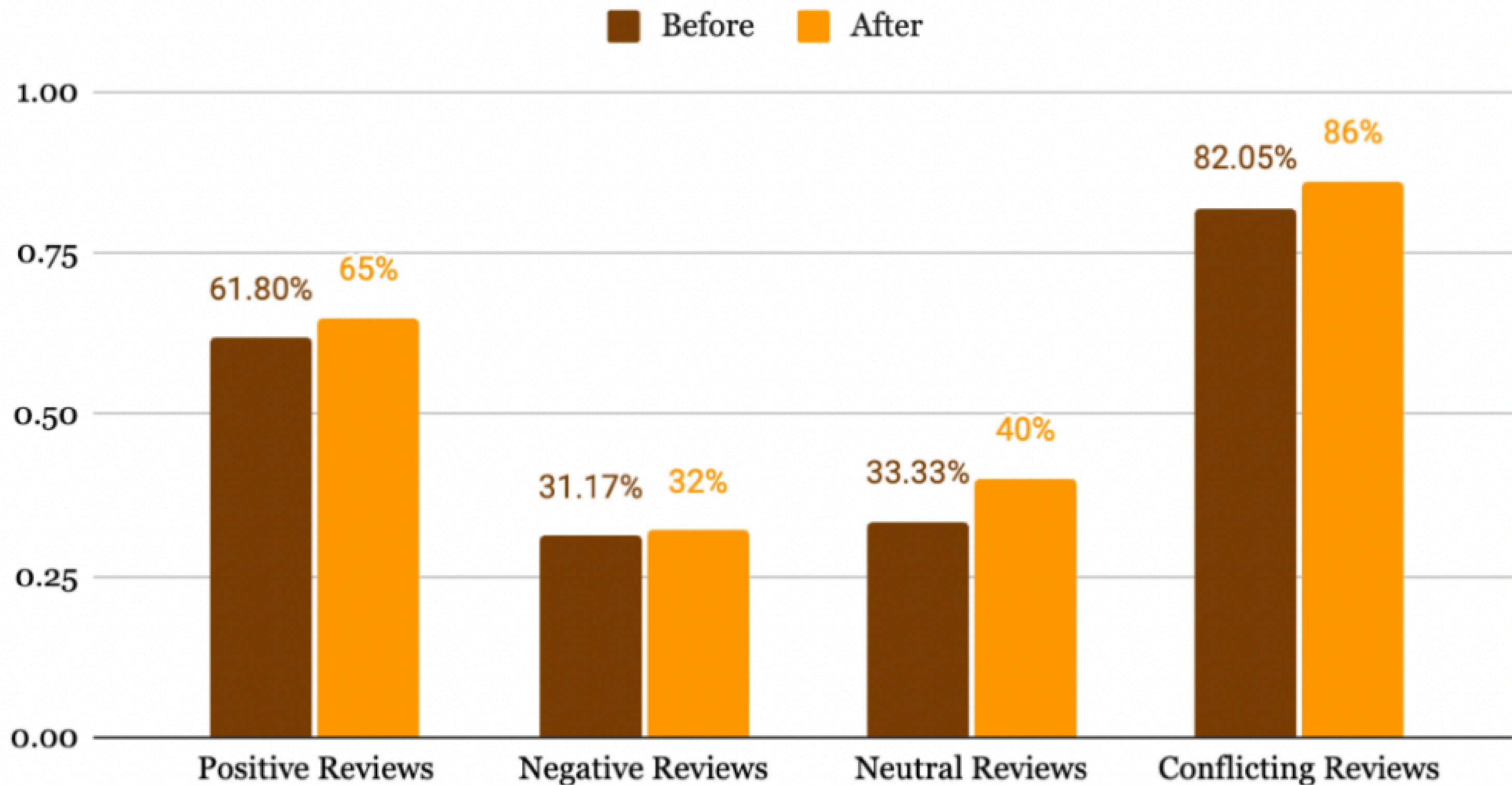
SAMPLE OUTPUT:

```
['the product does it does its job',  
'well',  
'however i really',  
"don't",  
'like',  
'the color given to me i',  
'want',  
'a',  
'refund']
```

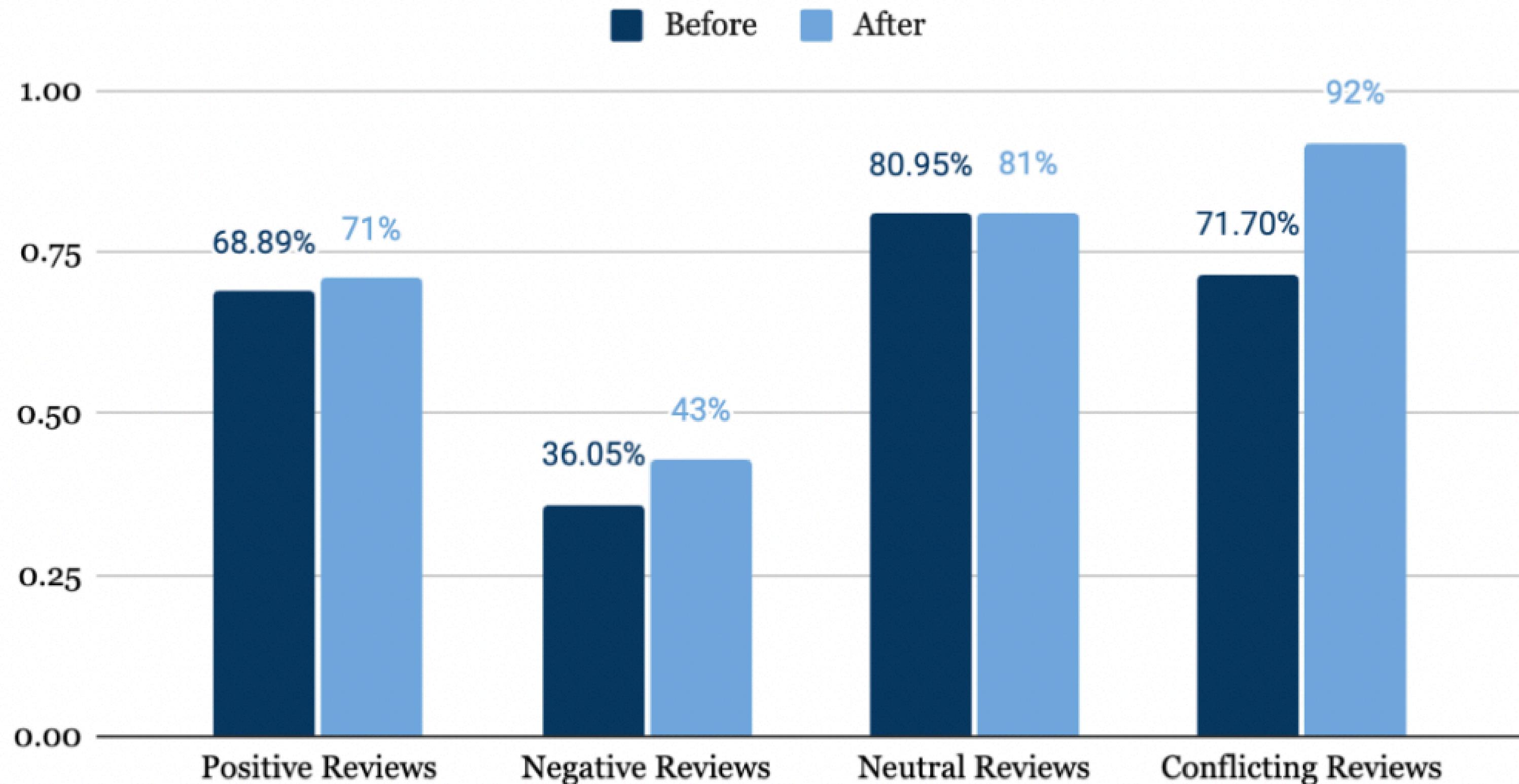
Creating lexicons for positive and negative phrases: Accuracy before and after



Creating positive and negative phrase lexicons: Precision for English reviews before and after



Creating positive and negative phrase lexicons: Precision for Tagalog reviews before and after



(1) LEXICON CHANGES

Creating positive and negative phrase lexicons 

Why did it work?

- Sentiments can now be identified as positive or negative even if they are not conveyed with words that are in the positive or negative lexicons

(1) LEXICON CHANGES

Identifying the polarity of words with dual meanings



Sample words

- “pretty”, “cheap”, “bait”, “sama”

Process

- The string is checked if it contains dual meaning lexicons, and assigns a sentiment through the `dualMeaningChecker()`

dualMeaningChecker()

- The function **removes the dual meaning lexicons** of a list, and gets the **non-neutral lexicons** of that list
- The **sentiment value** of the remaining lexicons is **computed**, and the corresponding **polarity is assigned** to the dual meaning lexicon
 - the lexicon is replaced with “**good**” to denote positive sentiment
 - or replaced with “**bad**” to denote negative sentiment

dualMeaningChecker() SAMPLE RESULTS

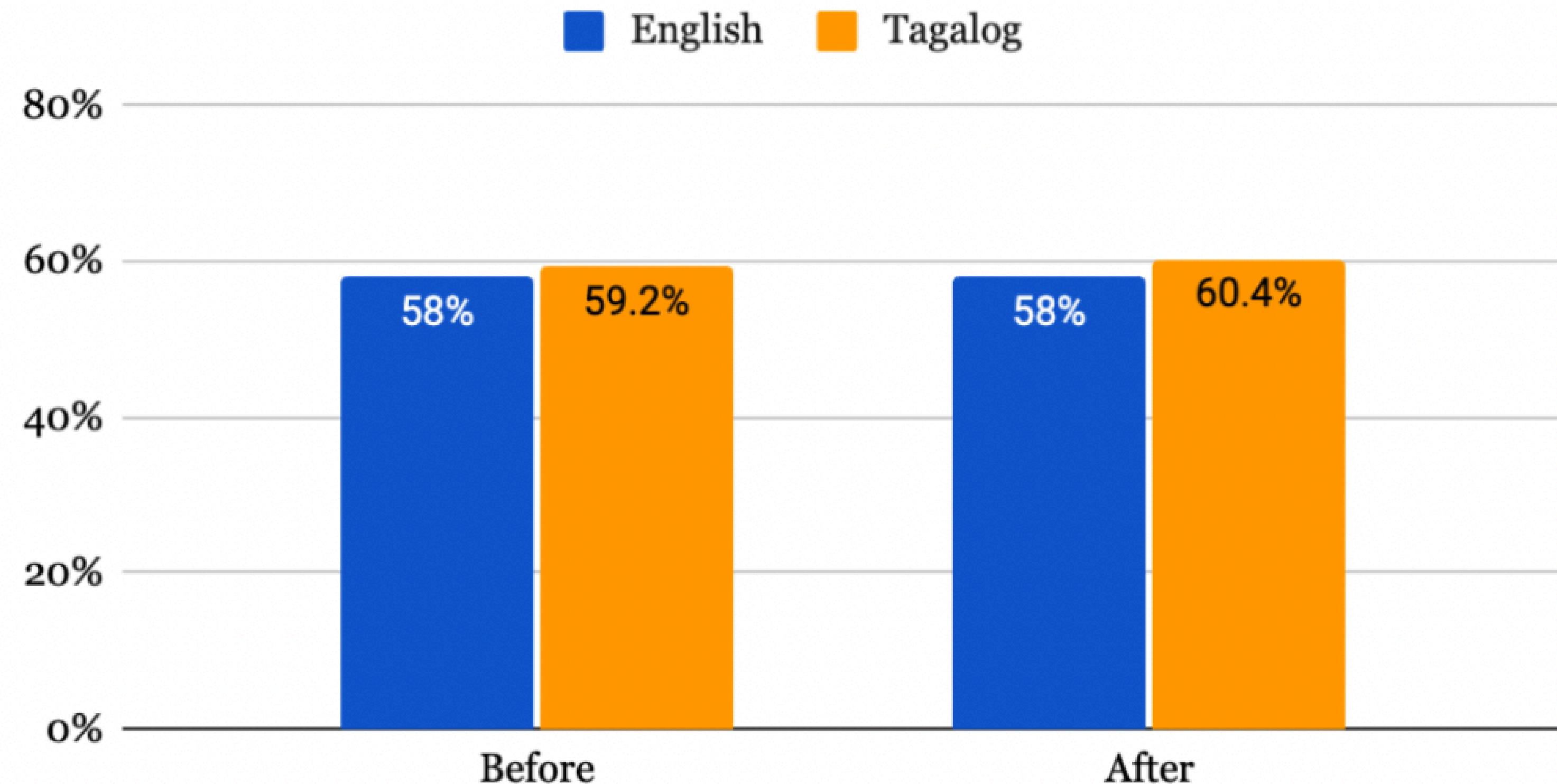
```
dualMeaningChecker(processText("The  
shirt is so ugly, its easily broken, the  
product advertisement was a bait."))
```

ugly (state 2)
broken (state 2)
bad (state 2)
String:
'Negative'

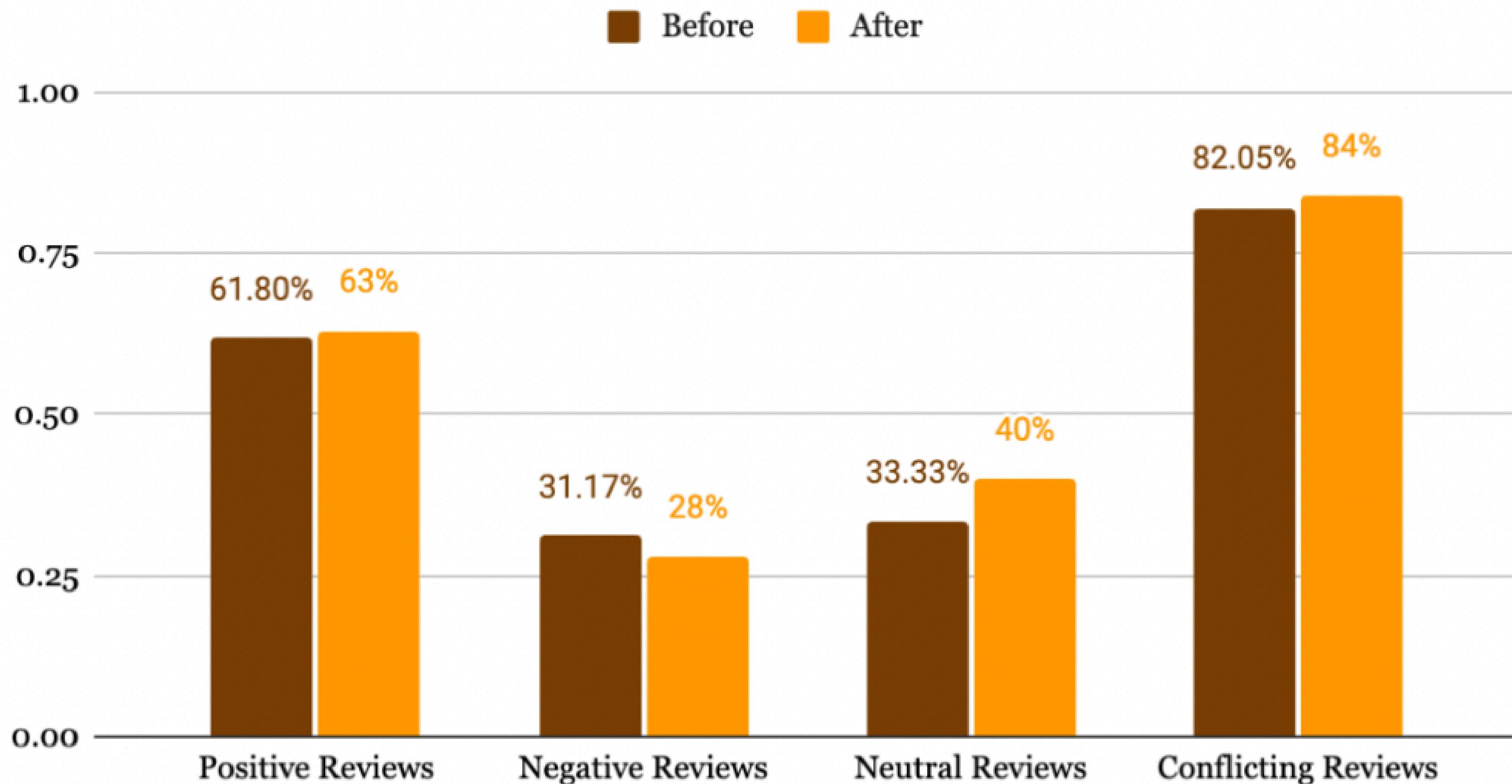
```
dualMeaningChecker(processText("Ang  
ganda ganda nung shirt, tapos ang bait  
ni seller. Thank you po"))
```

ganda (state 1)
ganda (state 1)
good (state 1)
thank (state 1)
'Positive'

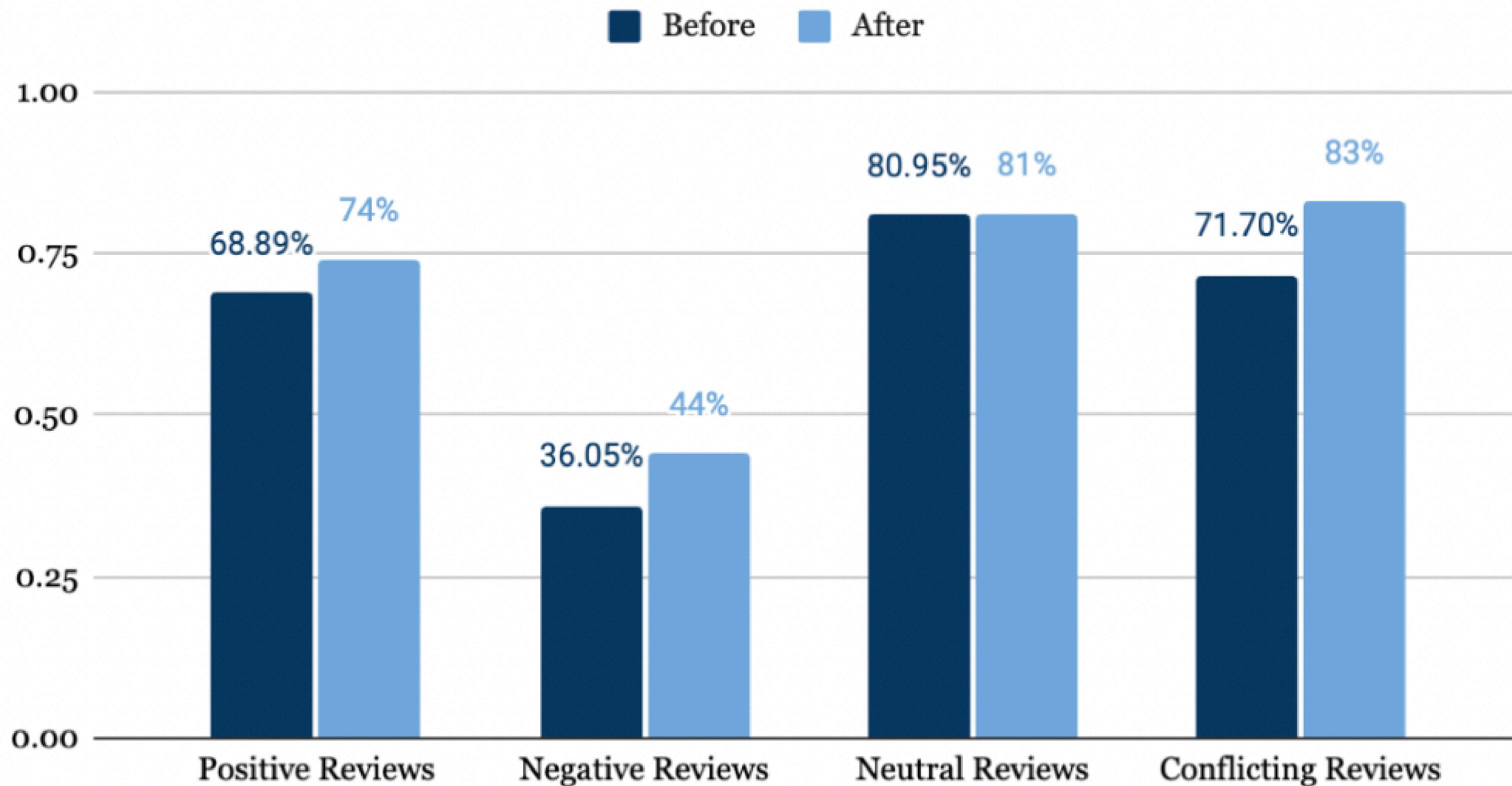
Identifying the polarity of words with dual meanings: Accuracy before and after



Identifying the polarity of words with dual meanings: Precision for English reviews before and after



Identifying the polarity of words with dual meanings: Precision for Tagalog reviews before and after



(1) LEXICON CHANGES

Identifying the polarity of words with dual meanings



Why did it work?

- Avoids misclassification of the dual-meaning words by holding onto assigning any non-neutral polarity to the word
- Dual-meaning words can now be identified contextually, relative to the sentiment of the other lexicons

(2) CREATING A FUNCTION TO SEPARATE AND EVALUATE SENTENCES



Process

- Splits a paragraph review into a list of sentences
- Passes each sentence into the FSA individually

paragraphSentiment()

- Splits paragraph into a list of sentences by period (.)
- Evaluates paragraphs with the same logic as sentence-level algorithm:
 - If there is a **positive and negative sentence**, or there is a **conflicting sentence** --> **conflicting** paragraph
 - **Positive sentences only** --> **positive** paragraph
 - **Negative sentences only** --> **negative** paragraph
 - **Neither positive nor negative sentences** --> **neutral** paragraph

ganda niyatry lang oorder ulit. kala ko malapad Hindi pala. pero ang ganda Saka madikit

Using the FSA alone:

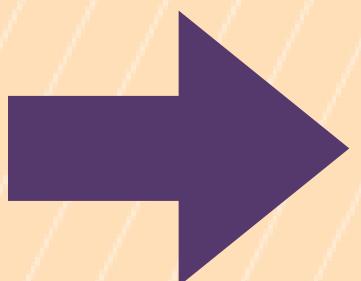
- Result gives us conflicting. However, the statement should be positive

ganda (state 1)
ulit (state 1)
hindi (state 5)
ganda (state 3)
saka (state 3)
Conflicting

Using the FSA with the paragraphSentiment() function

- Each sentence's polarity is determined

```
['Positive', 'Neutral', 'Positive']
```



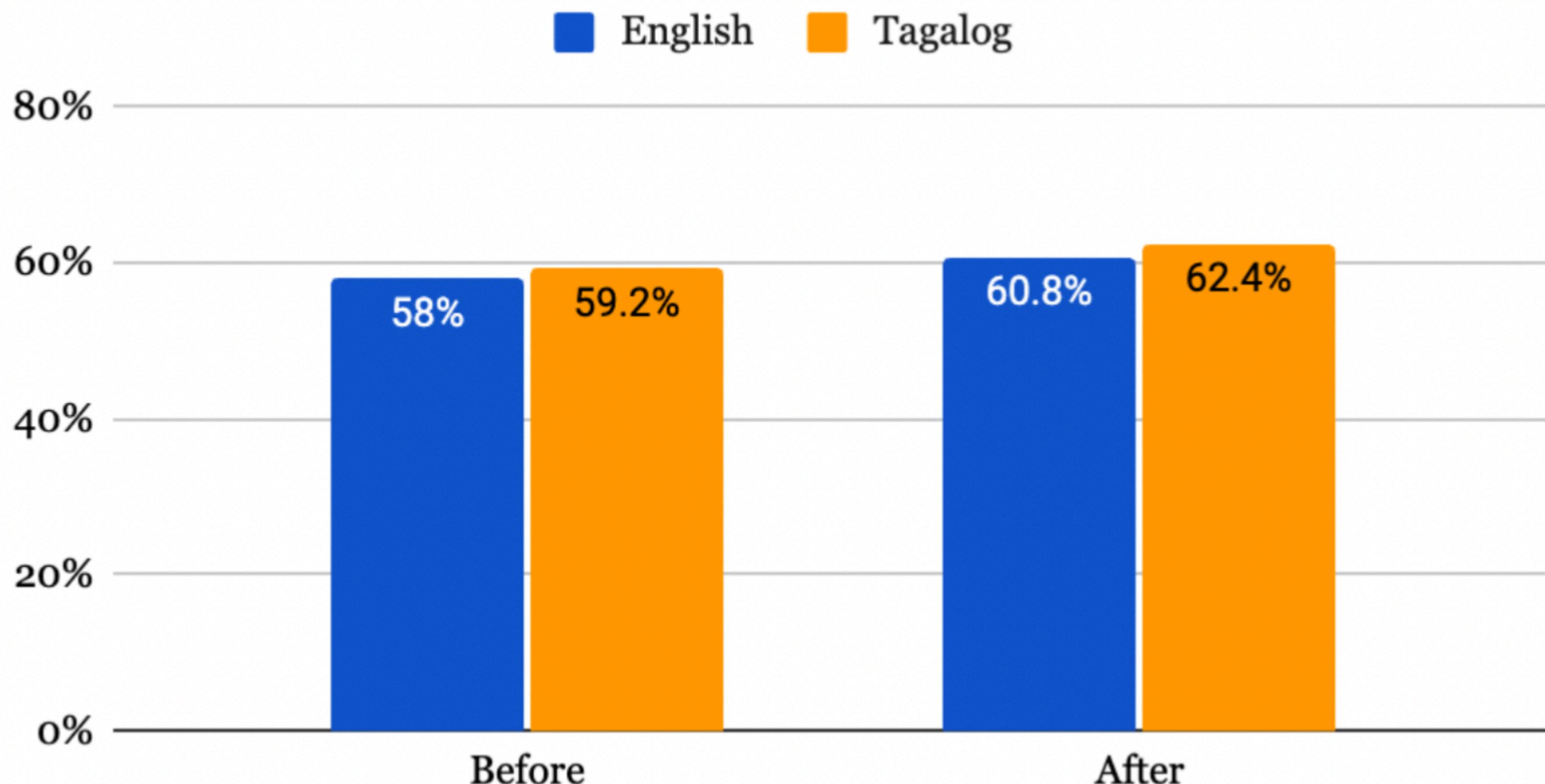
Positive
Sentiment

paragraphSentiment()

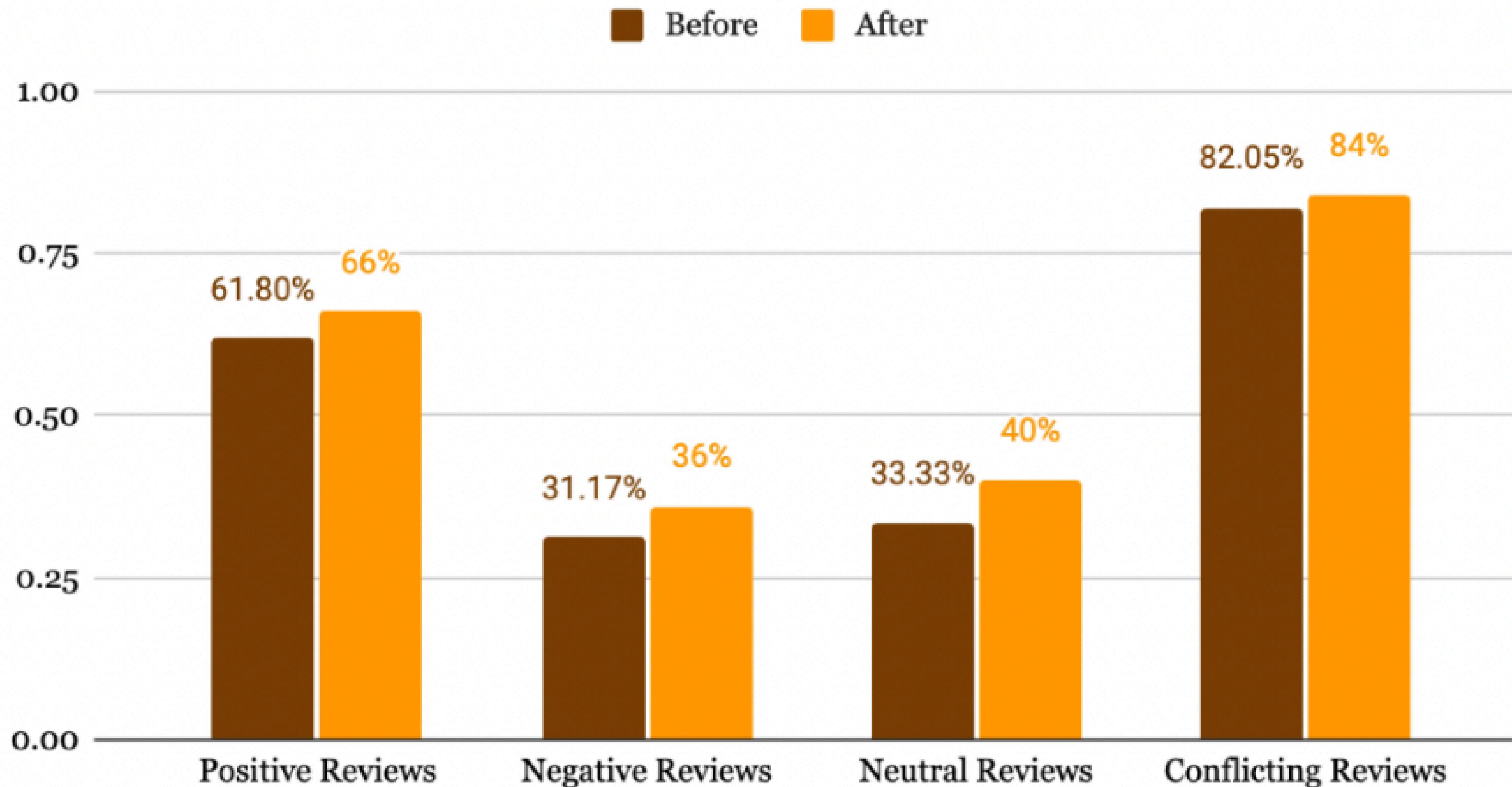
```
for i in englishReviews[0:250]:  
    result = sentiAutomata(Senti_FSA, i[0])  
    if result == "Conflicting" and paragraphSentiment(i[0]) == "Positive":  
        result = "Positive"  
    elif result == "Conflicting" and paragraphSentiment(i[0]) == "Negative":  
        result = "Negative"  
    if result == i[1]:
```

- This works because **if the result under sentiment automata is conflicting**, this implies that a **positive and negative sentiment is identified by the model**.
- However, **if the paragraph sentiment of the review is either positive or negative**, then it means that there is **no conflicting sentiments in each sentence**. Thus, there **must be a negation** being carried over to the next sentence.

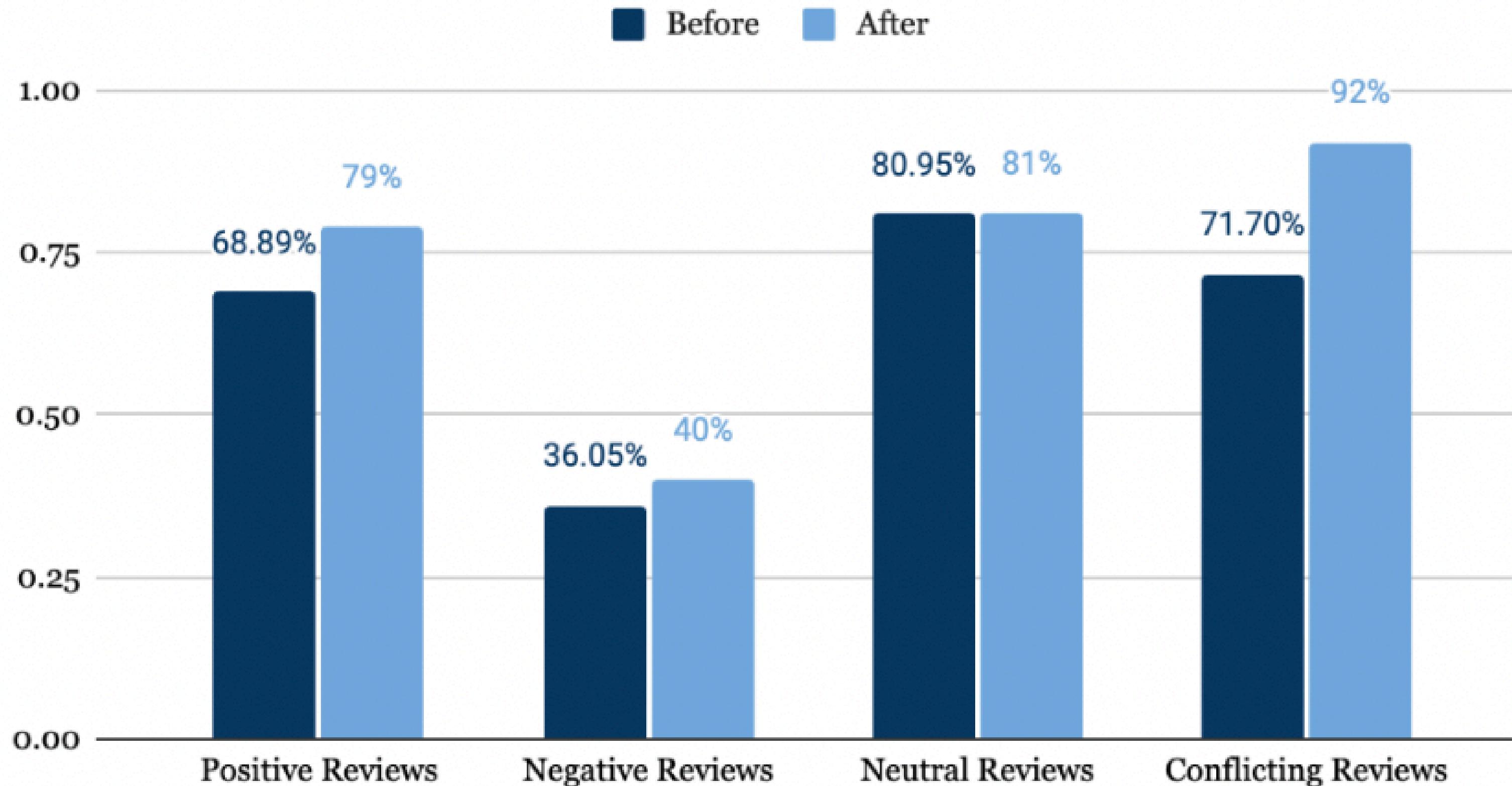
Creating a function to separate and evaluate sentences: Accuracy before and after



Creating a function to separate and evaluate sentences: Precision for English reviews before and after



Creating a function to separate and evaluate sentences: Precision for Tagalog reviews before and after



(2) CREATING A FUNCTION TO SEPARATE AND EVALUATE SENTENCES

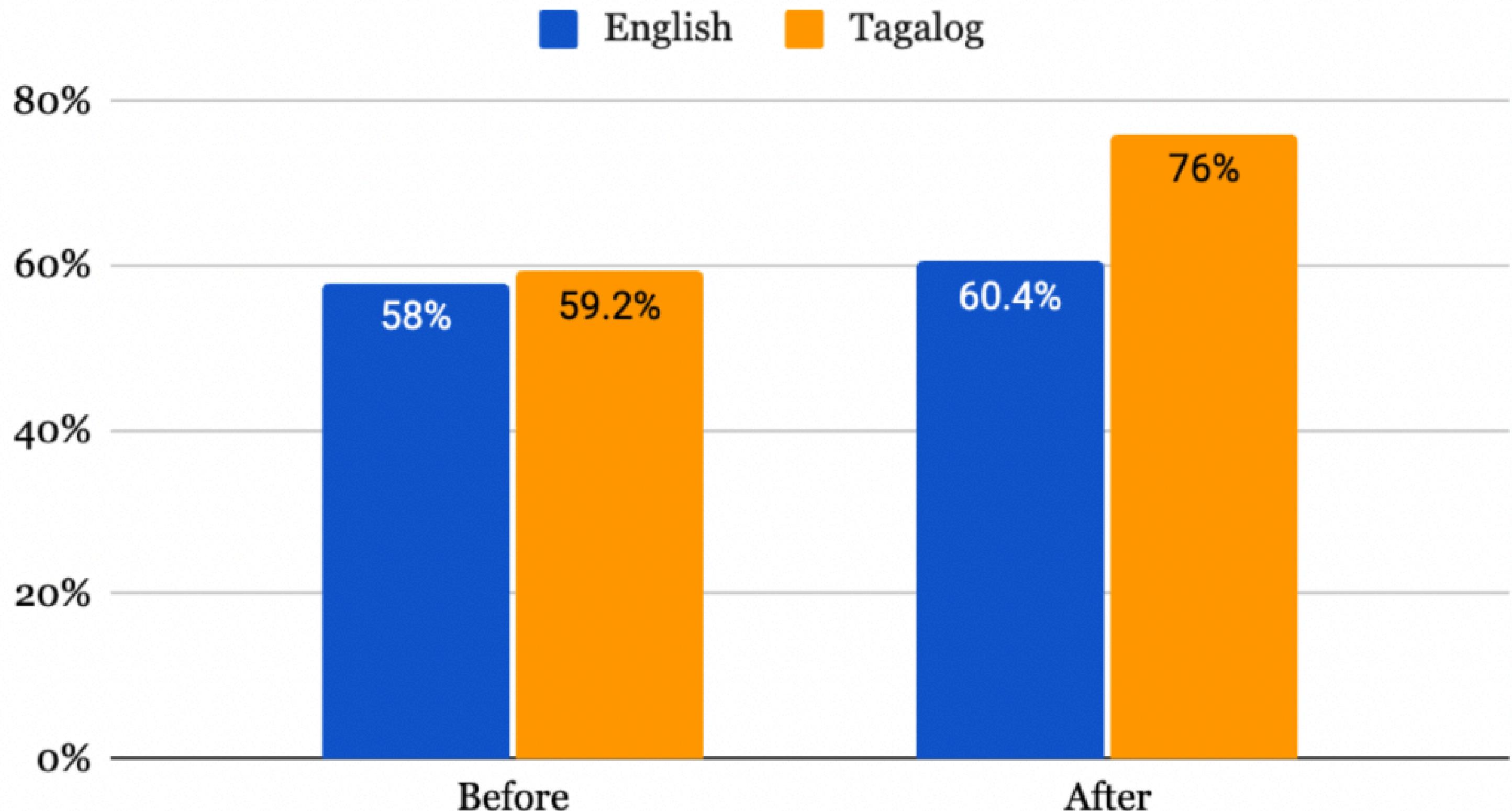


Why does it work?

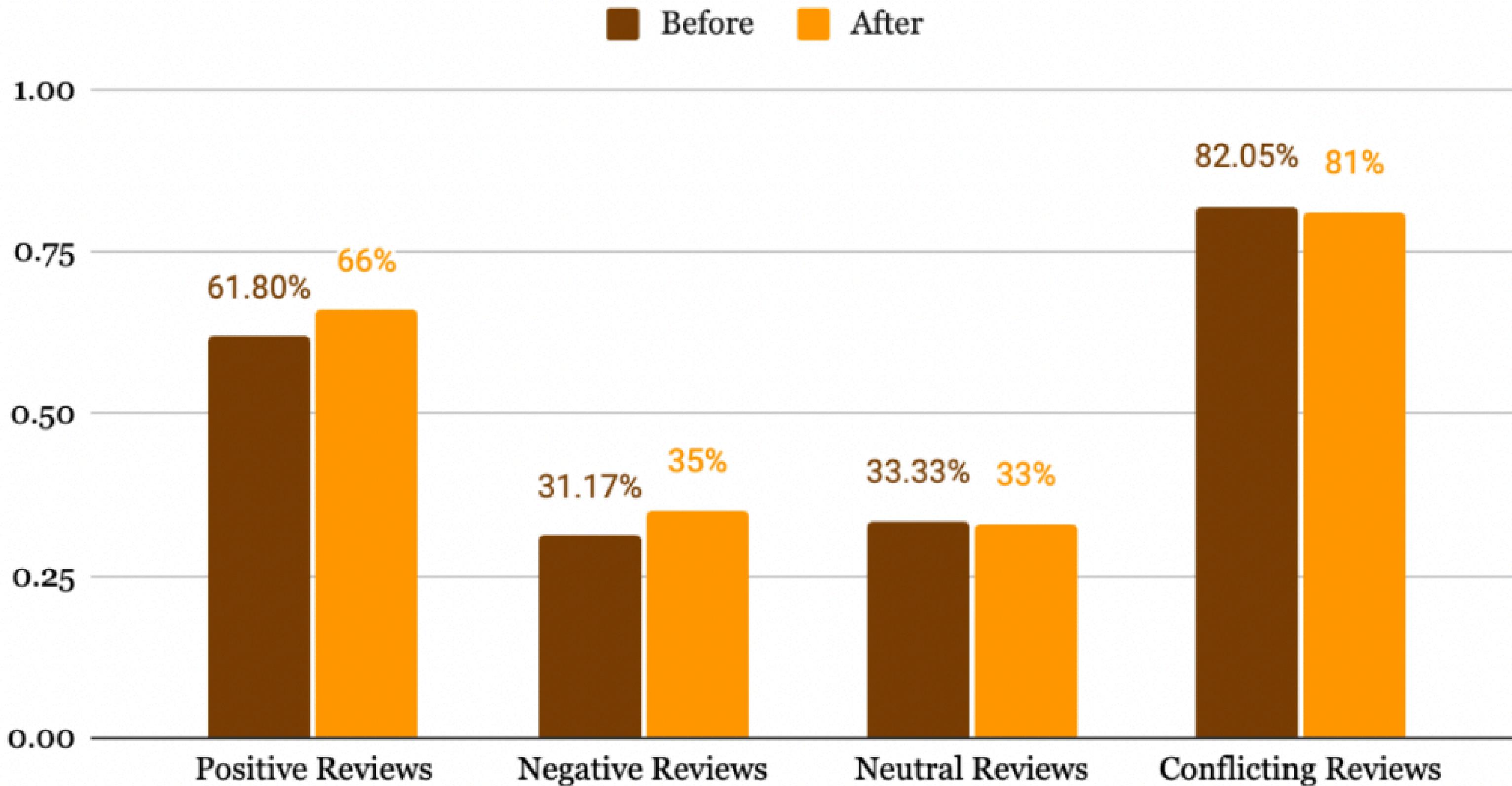
- This avoids cases in which the negation from the previous sentence carries over to the next sentence, unintentionally
- Creates a double-stage sentiment checking for positive or negative reviews

(3) COMBINING ALL EFFECTIVE ADJUSTMENTS

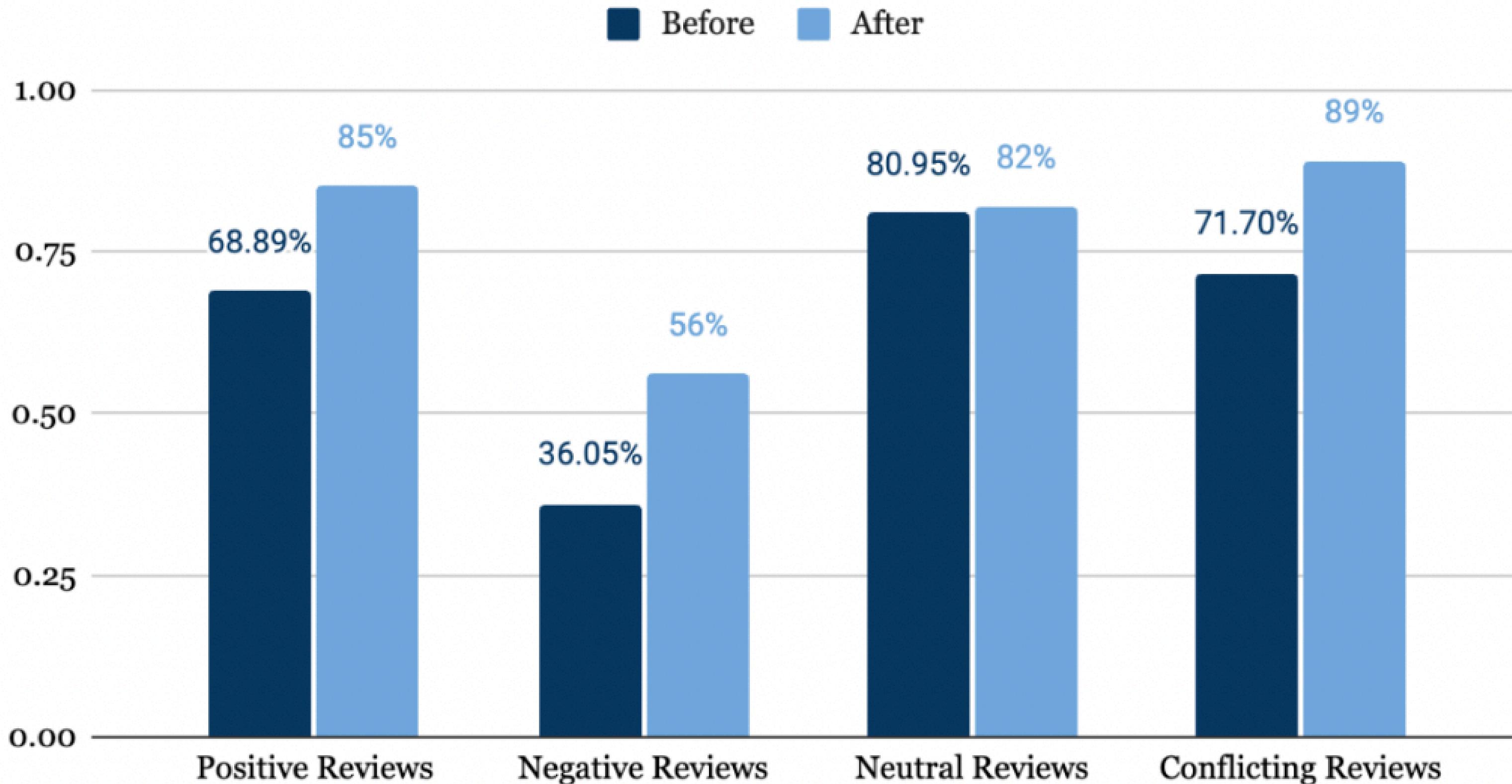
Combining all effective adjustments: Accuracy before and after



Combining all effective adjustments: Precision for English reviews before and after



Combining all effective adjustments: Precision for Tagalog reviews before and after



COMBINING EFFECTIVE ADJUSTMENTS



Discussion

- Decreased precision for neutral and conflicting English reviews
 - Combination of effective adjustments may not have been compatible
- Significant increase in precision for negative reviews
 - Can be further increased by adding more negative words to the Tagalog lexicon

COMBINING EFFECTIVE ADJUSTMENTS

LOW ACCURACY FOR ENGLISH REVIEWS

I have this baby in my samsung note2 and there is no
t much to say except it seems to be working without
a flaw

SanDisk ultra 32 GB MicroSDHC Class 10 UHS-1 Memory Card with Adapter, exactly
what was expected and what was described. Could not ask for anything more.

- English reviews tend to have a lot of phrase lexicons.
- Phrase lexicons need to be comprehensive enough to correctly identify the sentiment

INSIGHTS

LOW PRECISION FOR NEGATIVE REVIEWS

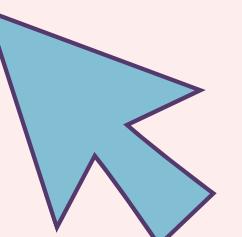
- Negative reviews tend to be longer
- Lots of negative reviews were aspect-specific. Subjects of sentiments were varying and the model could not distinguish.

“grabe **sobrang maganda yung product** at tsaka pagkabalot, what you see is what you get talaga. madali syang iadjust at yung para sa back ng cellphone ay designed sya para hindi madaling mahulog. I've always wanted this **masakit kasi yung neck ko kakatingin** sa ibaba. Thank you!”



05

CONCLUSION AND RECOMMENDATIONS



CONCLUSIONS



SUMMARY OF RESULTS



- The finite state automata is effective in analyzing positive and conflicting reviews. However, it has difficulties in correctly categorizing neutral and negative reviews.
- The FSA is highly dependent on the quality of the lexicon dictionaries. As such, constant reviewing and updating of the dictionary is recommended.

CONCLUSIONS

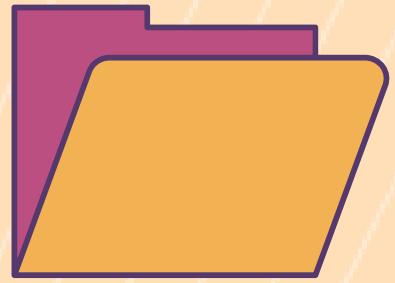


SUMMARY OF RESULTS



- Sentence-level sentiment analysis might not be as effective for negative reviews, so implementing an FSA aspect-level sentiment analysis is recommended
- Adding grammatical features increases the accuracy of the model.

CONCLUSIONS



SUMMARY OF RESULTS



- Using the finite state automata can be an effective approach to **sentence-level sentiment analysis**
 - Strengths:
 - Customizable lexicons (e.g. phrase and dual-meaning lexicons)
 - Weaknesses:
 - Tedium
 - Context and tone are not identified

RECOMMENDATIONS

FOR FUTURE STUDIES

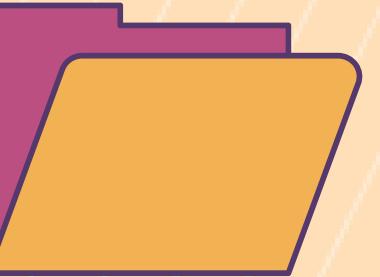


- Implement the Finite State Automata into **aspect-level sentiment analysis**
 - Divide text data based on its aspects
 - Categorize sentiments according to aspects
- This can avoid **spam messages** and **unrelated comments** from affecting the analysis

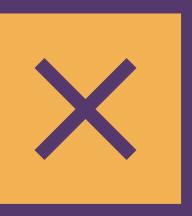


RECOMMENDATIONS

FOR FUTURE STUDIES



- Consider setting up multi-layer checking for sentiments, with the Sentiment Finite State Automata as its foundation.
- This can allow the consideration of more cases of linguistic intricacies
- Implement an effective spelling checker to identify wrongly spelled lexicons





REFERENCES

- [1] A. L. Andrei. "Development and evaluation of Tagalog Linguistic Inquiry and word count (LIWC) dictionaries for negative and positive emotion." *The MITRE Corporation*, 2014.
- [2] R. J. Bilog, "Application of naïve Bayes algorithm in sentiment analysis of Filipino, English and Taglish Facebook comments," *International Journal of Management and Humanities*, vol. 4, no. 5, pp. 73–77, Jan. 2020, doi: 10.35940/ijmh.e0524.014520.
- [3] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-based Systems*, vol. 226, p. 107134, Aug. 2021, doi: 10.1016/j.knosys.2021.107134.
- [4] A. J. V. Boquiren, R. A. Garcia, C. J. D. Hungria, and J. C. De Goma, "Tagalog sentiment analysis using deep learning approach with backward slang inclusion," in *Proceedings of the International Conference on Industrial Engineering and Operations Management* Nsukka. 2022. doi: 10.46254/af03.20220180.

REFERENCES

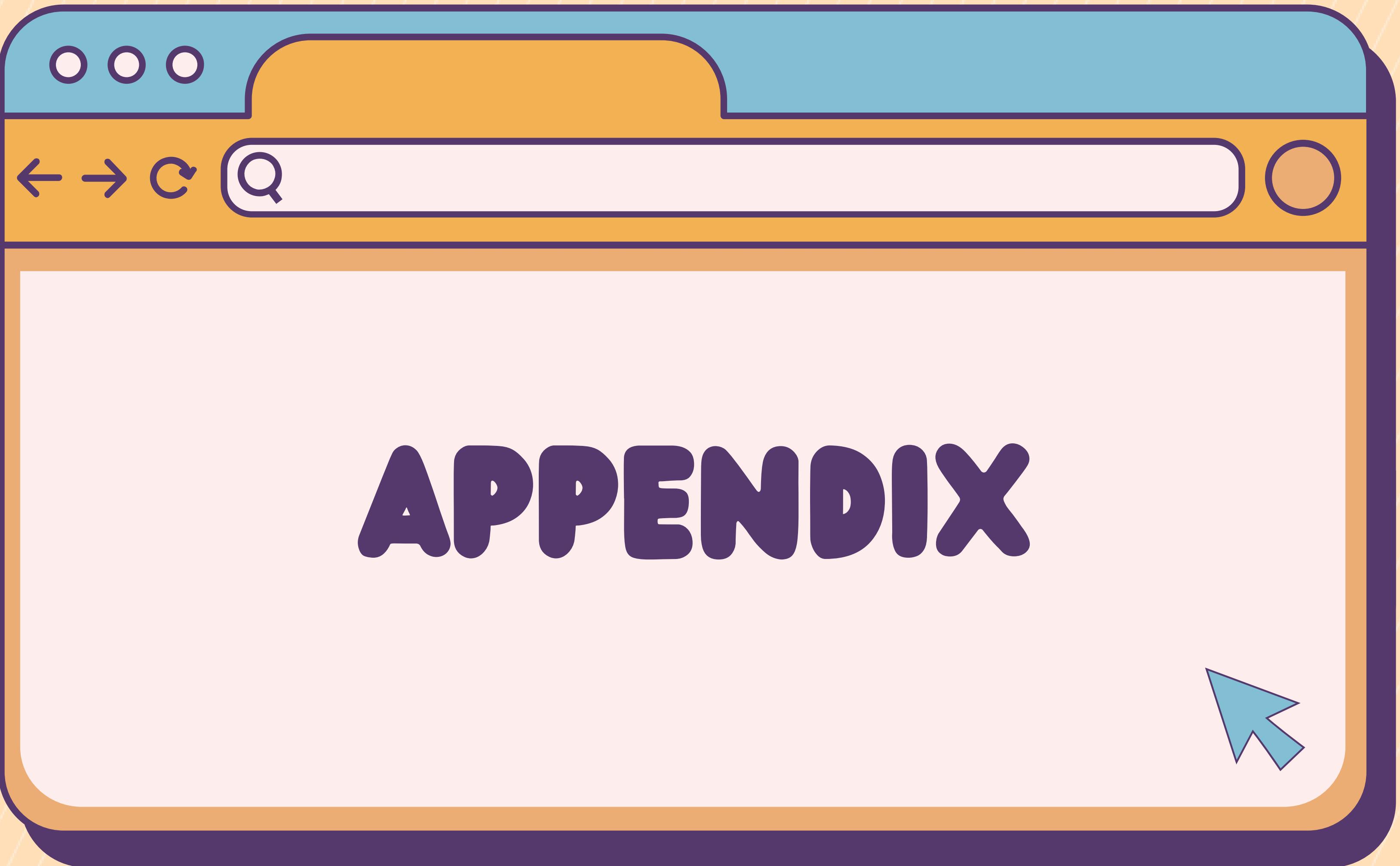
- [5] Y. Castañeda et al., “UMCC_DLSI: A Probabilistic Automata for Aspect Based Sentiment Analysis,” in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. 2014. doi: 10.3115/v1/s14-2129.
- [6] T. Eng, Md. R. I. Nawab, and K. M. Shahiduzzaman, “Improving Accuracy of The Sentence-Level Lexicon-Based Sentiment Analysis Using Machine Learning,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 57–69, doi: 10.32628/cseit21717.
- [7] T. Eng, Md. R. I. Nawab, and K. M. Shahiduzzaman, “Improving Accuracy of The Sentence-Level Lexicon-Based Sentiment Analysis Using Machine Learning,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 57–69, doi: 10.32628/cseit21717.
- [8] M. Kaity and V. Balakrishnan, “Sentiment lexicons and non-English languages: a survey,” *Knowledge and Information Systems*, vol. 62, no. 12, pp. 4445–4480, Jul. 2020, doi: 10.1007/s10115-020-01497-6.

REFERENCES

- [9] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Social Network Analysis and Mining*, vol. 11, no. 1, p. 81, 2021.
- [10] S. Pelosi, A. Maisto, L. Melillo, and A. Elia, "Sentiment Analysis through finite State Automata," in *Lecture notes in computer science*, 2023, pp. 182–197. doi: 10.1007/978-3-031-24340-0_14.
- [11] V. G. Curada, et al. "Lexicon-Based Sentiment Analysis of Professor Evaluation Students' Comments with Code Switching using Naive Bayes Algorithm and Support Vector Machines," 2022.

DATASETS

- A. Perez, 2022, "Filipino Word-Sentiment Dictionary", Kaggle [Online]. Available: <https://www.kaggle.com/datasets/abeperez/filipino-wordsentiment-dictionary>
- F. Nielsen, 2011, "AFINN", Technical University of Denmark [Online]. Available: <https://www2.imm.dtu.dk/pubdb/pubs/6010-full.html>





COMPREHENSIVE REVIEW OF RELATED LITERATURE

ON SENTIMENT ANALYSIS



GENERAL INFORMATION



- Useful for product or service improvement, especially when sentiment analysis models can be done at **aspect-levels**
- Sentiment analysis entails **data mining**. This involves sorting through data to **identify patterns and relationships**.
- This is particularly useful when reviews are posted on websites that are **not particularly dedicated for reviews**

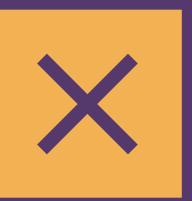
(Birjali et al., 2021)

(Nandwani and Verma, 2021)

ON SENTIMENT ANALYSIS



THREE LEVELS OF SENTIMENT ANALYSIS



- **Sentence level:** documents or **paragraphs are broken into sentences**. The polarity of each sentence is identified.
- **Document level:** sentiment is detected from the **entire document**. Usually used to extract global sentiment from long texts that contain redundancy.
- **Aspect level:** opinion about a **specific aspect or feature** of a subject is determined.
 - For example, “I don’t like the teacher’s assessments” describes dislike toward the assessments, not the teacher.

(Birjali et al., 2021)

(Nandwani and Verma, 2021)

ON LEXICON AND SENTIMENT VALUES



LEXICON CATEGORIES

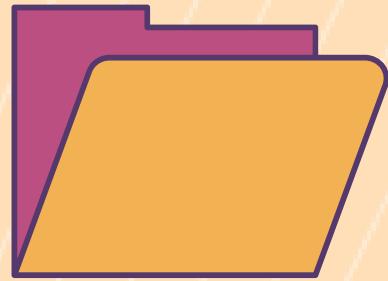


- Studies more or less consider **seven lexical categories** in the analysis of texts.
- **Sentiment Lexicon**
 - Sentiment words which are assigned sentiment values
 - Ex. Good, satisfactory, dependable
- **Intensifier Lexicon**
 - It can lower or higher the strength of the sentiment words
 - Ex. extremely, tremendously, etc.
- **Negation Lexicon**
 - Reverse the polarity or reduce the strength of the sentiment word
 - Ex. not, lack of, etc.
- **Adjective Sense Disambiguation**
 - Vocabularies that have a bipolar meaning when taken in different contexts
 - Ex. cheap, pretty, high

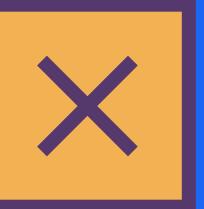
(Eng et al., 2021)

(Kaity and Balakrishnan, 2020)

ON LEXICON AND SENTIMENT VALUES



LEXICON CATEGORIES



- **Emoticon Lexicon**
 - Symbols that expresses emotions and sentiments
 - Ex. :-) or wink ;-) or frown :(or dog nose :O
- **Phrase and Idiom Lexicon**
 - Groups of words that trigger sentiments
 - Ex. stay away, let down, does its job
- **Syntax Pattern**
 - some syntax that looks like a negative sentence, but it is actually a positive sentence
 - Ex. I can't recommend it enough, I couldn't agree more

(Eng et al., 2021)

(Kaity and Balakrishnan, 2020)

ON LEXICON AND SENTIMENT VALUES



TAGALOG SENTIMENT LEXICONS



- A study developed a **Tagalog linguistic inquiry and word count dictionary for positive and negative emotion** to analyze mixed language Twitter data from the Philippines
- According to the study, if a dictionary is used for general texts, it is **recommended to start with translating from the English into the target language.**
- It was concluded that **negative emotions were much easier to detect** based on sentiment lexicons, **and that positive emotions were harder to detect.** This was because negative lexicons can turn into positive emotion when conjoined with other words in the same sentence.

ON IMPLEMENTATION OF SENTIMENT ANALYSIS



PREPROCESSING OF TEXTS



- Preprocessing practices
 - Removal of stop words,
 - Removal of non-alphabetic and special characters,
 - Changing uppercase to lowercase
 - Removal of duplicates,
 - Removing links,
 - Removing numbers,
 - Removing punctuations,
 - Reverting repeated words,
 - Replacing emoticons, and
 - Normalizing mixed-language texts

ON IMPLEMENTATION OF SENTIMENT ANALYSIS

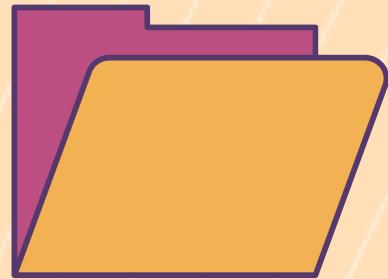


GENERAL METHODOLOGY



- Some studies divided their methodology into **4 modules**, namely the **input**, **preprocessing**, **process**, and **output** modules. The process module is further divided into the training module, which contains the lexicon module, and the testing module.
 - a. **Input Module** -- importing of text and lexicons
 - b. **Preprocessing Module** -- converting uppercase to lowercase, removal of punctuations, etc.
 - c. **Processing Module** -- Machine learning algorithm/similar procedures
 - d. **Output Module** -- Analysis, testing

ON LEXICON-BASED APPROACHES



LEXICON-BASED APPROACH



- Existing literature on sentiment analysis mainly focuses on two approaches: **machine learning-based** approaches, those that employ training datasets and annotated corpus, and **lexicon-based** approaches.
- Although, **lexicon-based approaches** are more time-consuming, due to the need of manually generating sentiment lexicons, they have been shown to be **more accurate than other methods**

(Birjali et al., 2021)

(Nandwani and Verma, 2021)

ON LEXICON-BASED APPROACHES



(Pelosi et al., 2019)

(Contreras et al., 2018)

(Castañeda et al., 2014).

- This study presents a method to exploit finite-state automata with the purpose of building high-performance tools for sentiment analysis
- The study computed the polarity of more than 15,000 Italian sentiment words
- Italian lexical databases are mostly created by translating and adapting the English ones
- Defining of **Overall Text Sentiment**:
 - Positive Only neutral and positive sentiments
 - Negative Only neutral and negative sentiments
 - Conflicting Existence of positive and negative sentiments
 - Neutral No positive or negative sentiment



