# EXPLAINABLE SENTENCE-LEVEL SENTIMENT ANALYSIS

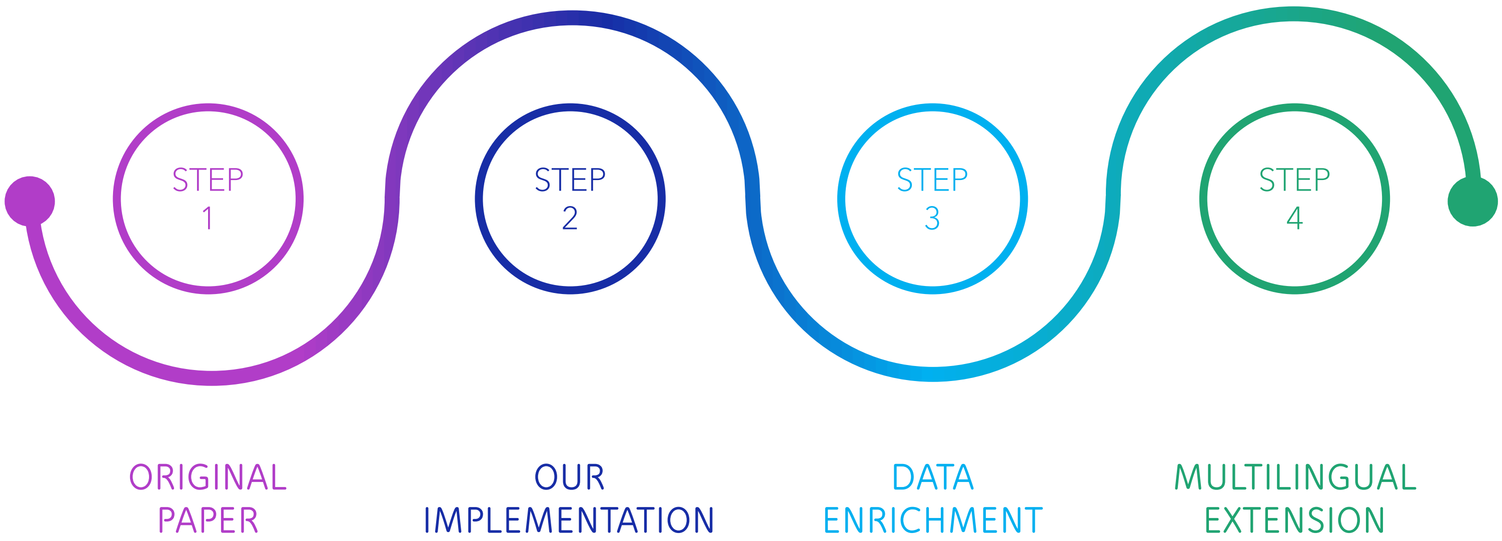
Salvatore Stefano Furnari
Politecnico di Torino
Torino, Italy
s290057@studenti.polito.it

Giuseppe Gallipoli
Politecnico di Torino
Torino, Italy
s291086@studenti.polito.it

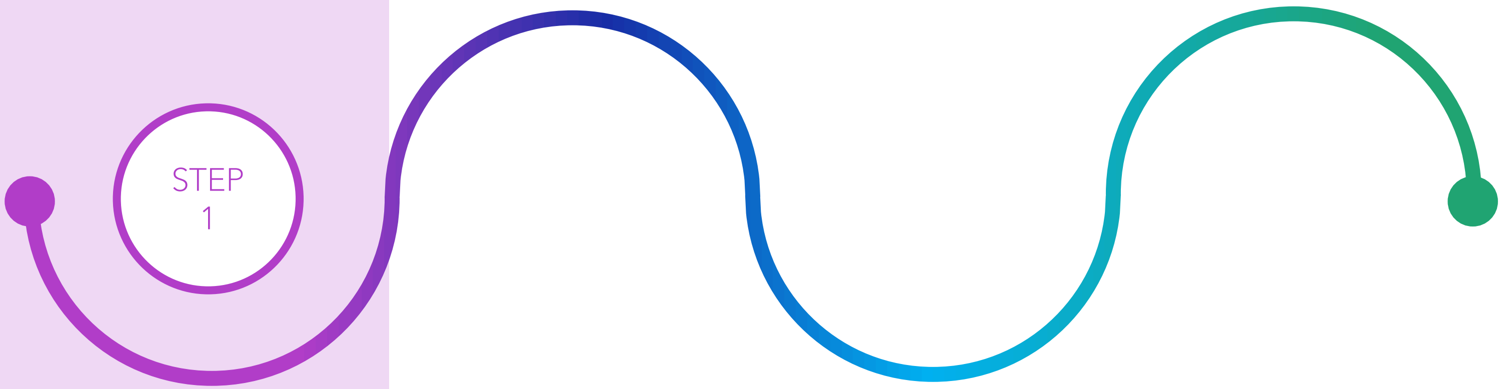
Marco Tasca
Politecnico di Torino
Torino, Italy
s285174@studenti.polito.it

Deep Natural Language Processing
Politecnico di Torino

# WORKFLOW


STEP 1
STEP 2
STEP 3
STEP 4
ORIGINAL PAPER
OUR IMPLEMENTATION
DATA ENRICHMENT
MULTILINGUAL EXTENSION

# WORKFLOW




ORIGINAL
PAPER

# ORIGINAL PAPER

## PROBLEM STATEMENT

### THE ISSUES

**High number** of reviews (non humanly readable).

**Inconsistency** between item descriptions and products.

**Black-box** models.

?

PROBLEMS

[1] Xuechun Li et al. Explainable Sentence-Level Sentiment Analysis for Amazon Product Reviews

# ORIGINAL PAPER

## PROBLEM STATEMENT

### THE ISSUES

**High number** of reviews (non humanly readable).

**Inconsistency** between item descriptions and products.

**Black-box** models.

### THE MODEL

A **sentiment analysis model**, with a module about **explainability**.

It automatically **labels reviews** as positive or negative, extracting the **latent sentiment score** of each review.

?

PROBLEMS

!

SOLUTIONS

[1] Xuechun Li et al. Explainable Sentence-Level Sentiment Analysis for Amazon Product Reviews

# ORIGINAL PAPER

## PROBLEM STATEMENT

### THE ISSUES

**High number** of reviews (non humanly readable).

**Inconsistency** between item descriptions and products.

**Black-box** models.

### THE MODEL

A **sentiment analysis model**, with a module about **explainability**.

It automatically **labels reviews** as positive or negative, extracting the **latent sentiment score** of each review.

### ACHIEVEMENT

Both sellers and customers benefit from this sentiment measure, as a **fundamental index** for commodities.

With the module about explainability, we can check **how the score is given**.

PROBLEMS

SOLUTIONS

RESULTS

[1] Xuechun Li et al. Explainable Sentence-Level Sentiment Analysis for Amazon Product Reviews

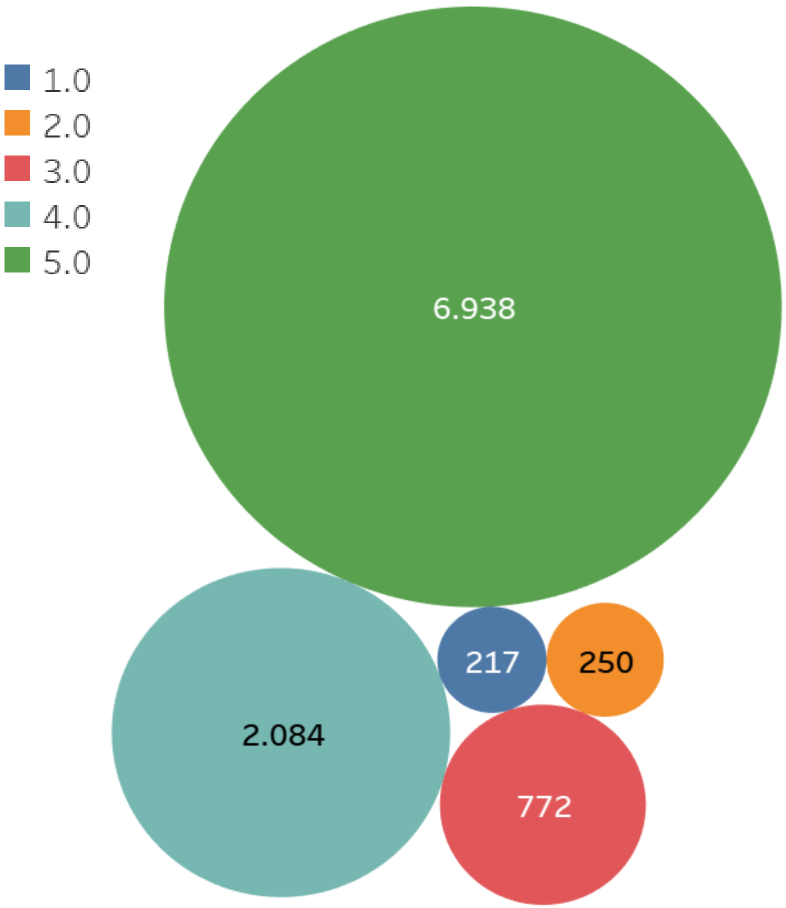
# ORIGINAL PAPER

## DATASET: Amazon Musical Instruments Reviews

### 10.261 SAMPLES

**Overall** score (from 1 to 5).

**Text** of the review (English).




number of reviews, based on their score

# ORIGINAL PAPER

## DATASET: Amazon Musical Instruments Reviews

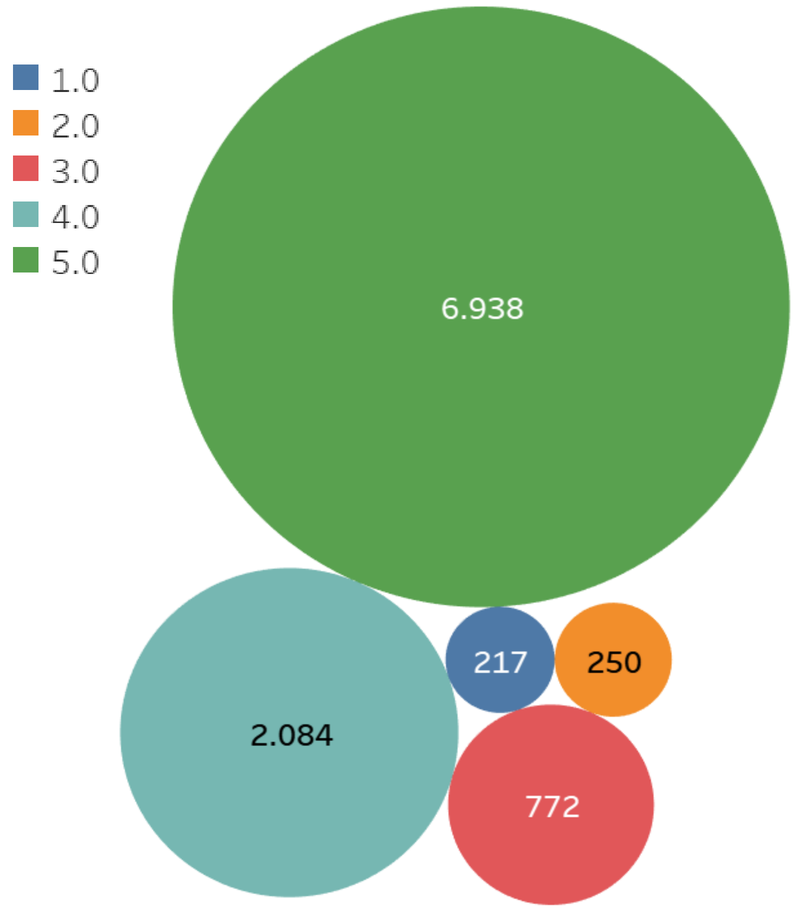
### 10.261 SAMPLES

**Overall** score (from 1 to 5).

**Text** of the review (English).

### HIGHLY UNBALANCED

**Impossible** to group into two sets creating a balanced distribution.




number of reviews, based on their score

# ORIGINAL PAPER

## DATASET: Amazon Musical Instruments Reviews

### 10.261 SAMPLES

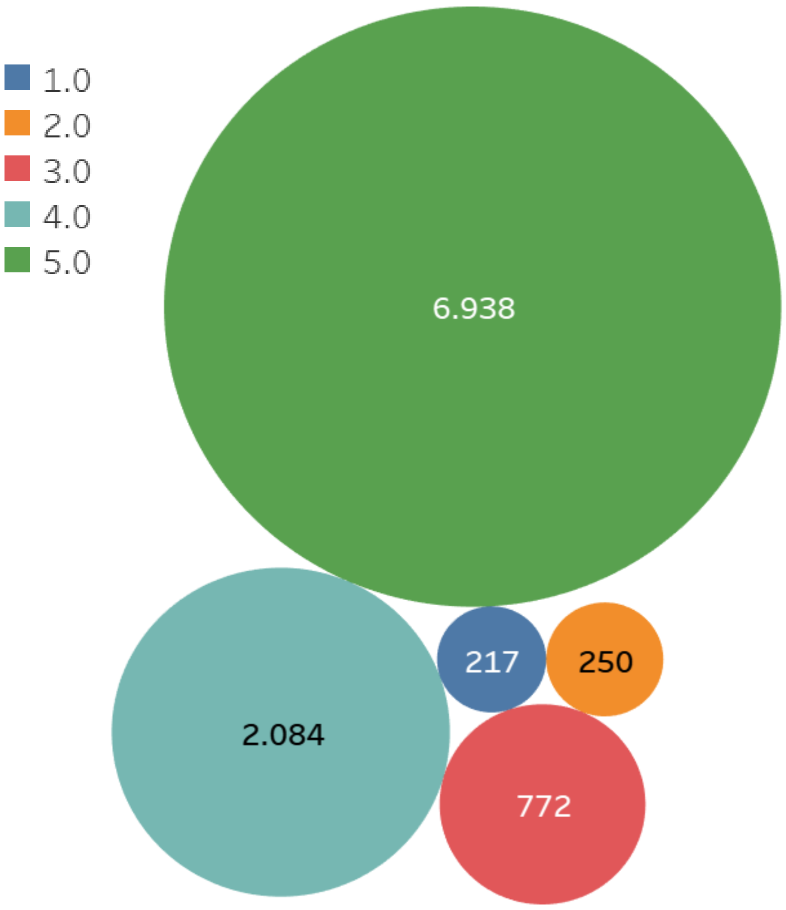
**Overall** score (from 1 to 5).

**Text** of the review (English).

### HIGHLY UNBALANCED

**Impossible** to group into two sets creating a balanced distribution.

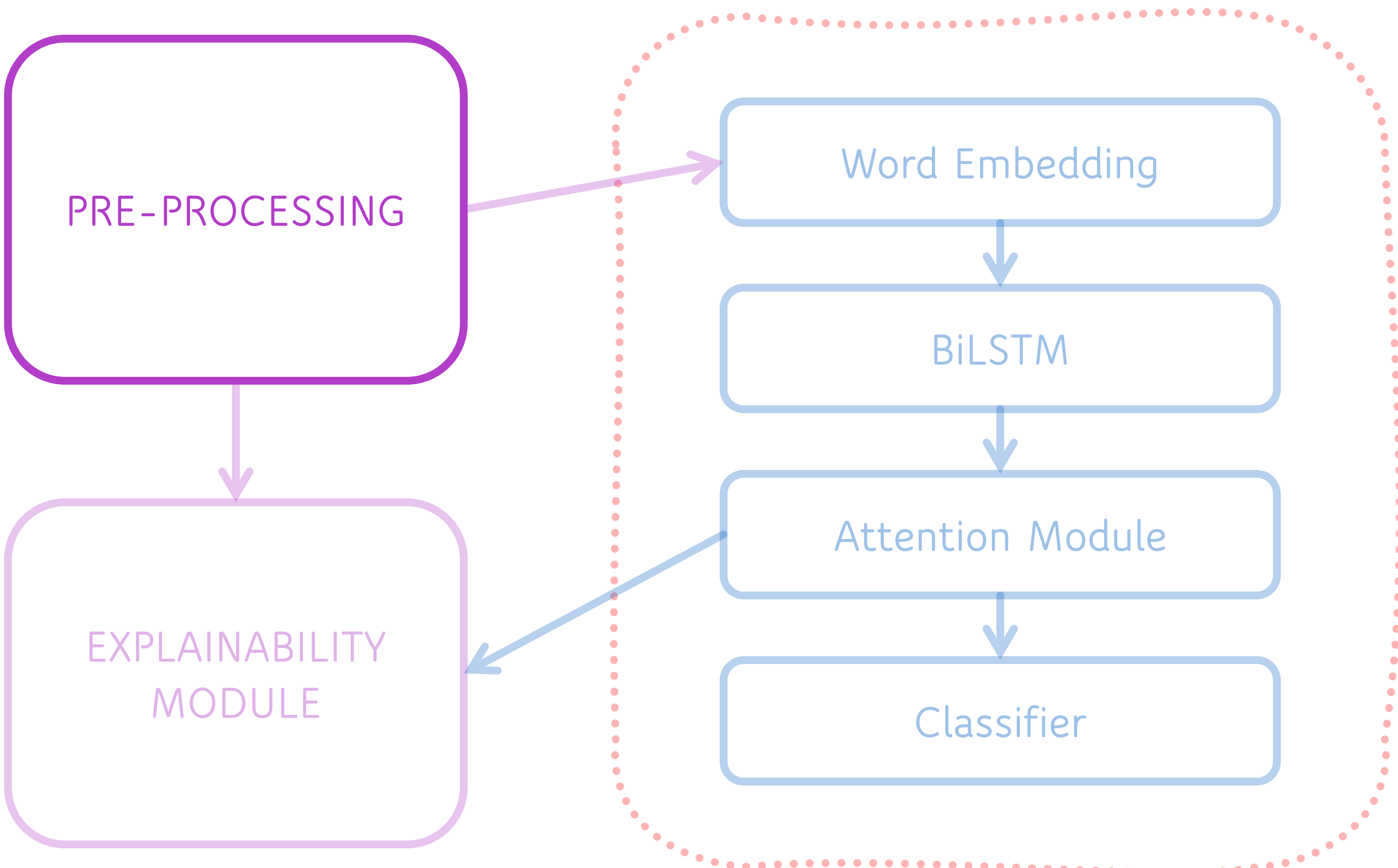
Not clear how Li et al. grouped them.




number of reviews, based on their score

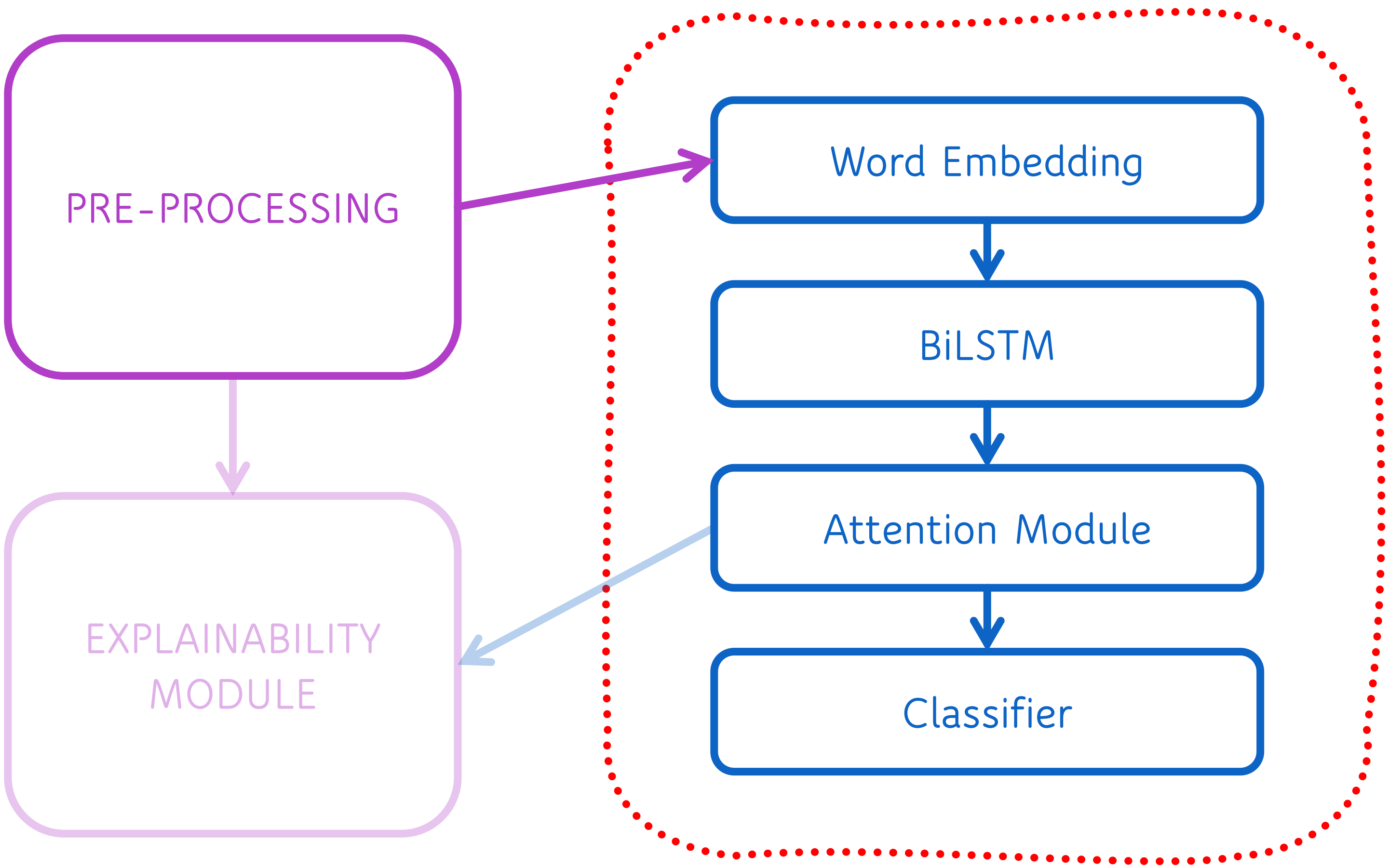
# ORIGINAL PAPER

## MODEL OVERVIEW



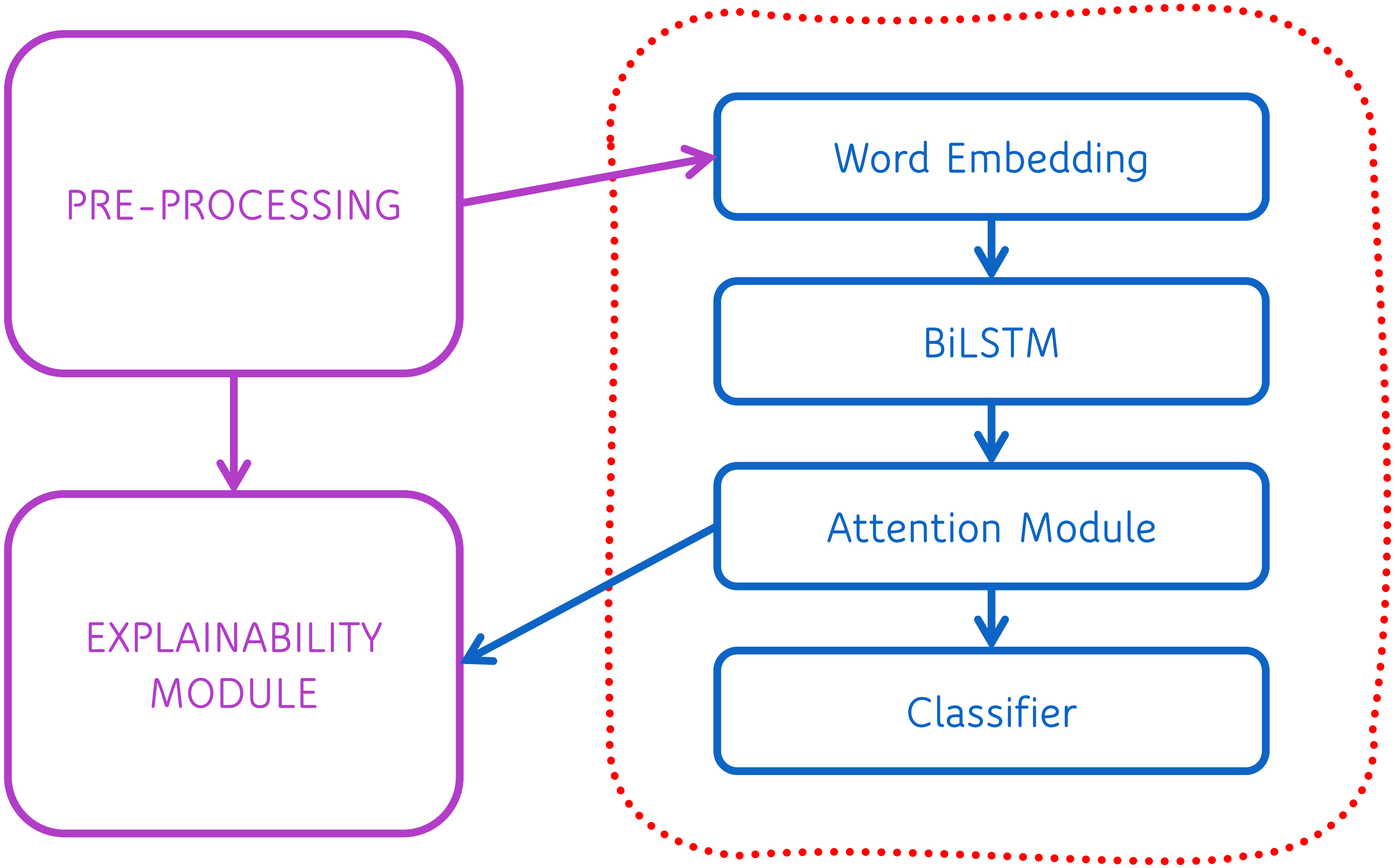
# ORIGINAL PAPER

## MODEL OVERVIEW

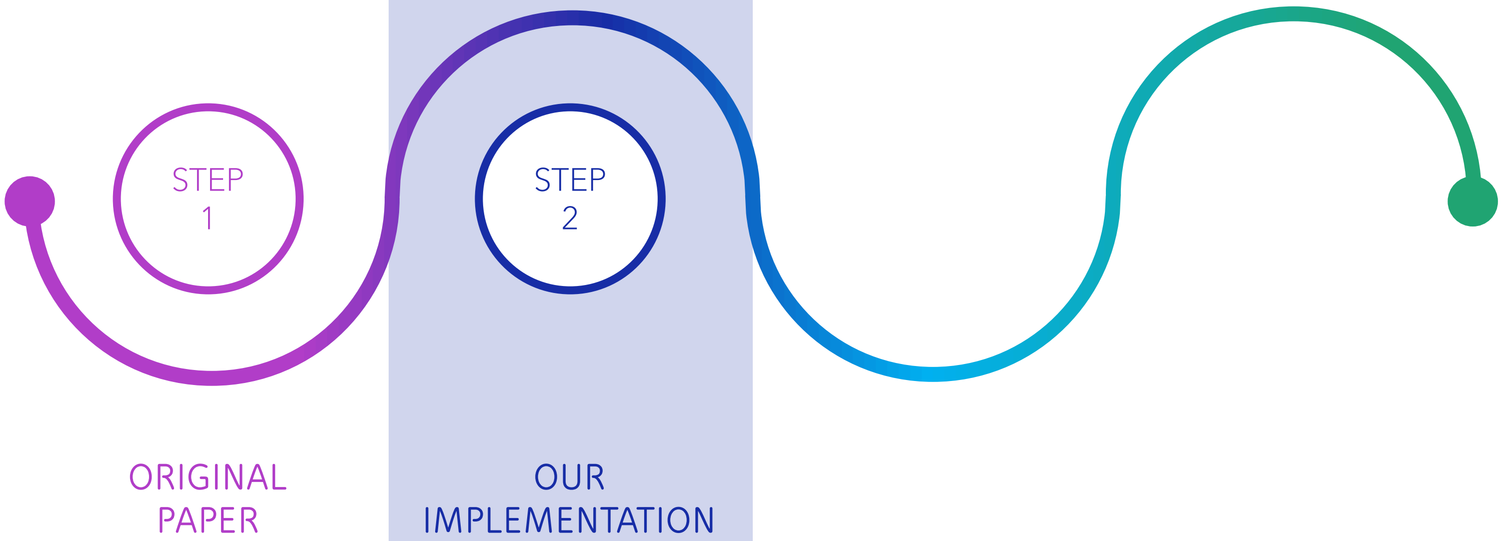


# ORIGINAL PAPER

## MODEL OVERVIEW



# WORKFLOW



# OUR IMPLEMENTATION

## PRE-PROCESSING

### TEXT CLEANING & BINARY LABELING

#### TEXT CLEANING

**reviewText**: we deleted analphabetic signs, lowercased, lemmatized and removed stopwords [NLTK].

#### BINARY LABELING

**Overall**: we grouped the 5 possible scores: {1,2,3} = 0 and {4,5} = 1.

| reviewerID | asin | reviewerName | helpful | reviewText | Overall summary | unixReviewTime | reviewTime |
|---|---|---|---|---|---|---|---|
| A2IBPI20UZIR0U | 13847192 | cassandra tu " Yeah, well, that's just like, u... | [0, 0] | Not much to write about here, … | 5good | 1393545600 | 02 28, 2014 |
| A14VAT5EAX3D9S | 138471932 | Jake | [13, 14] | The product does exactly as it should and is quite affordable... | 5Jake | 1363392000 | 03 16, 2013 |
| A195EZSQDW3E21 | 138471932 | Rick Bennette " Rick Bennette" | [1, 1] | The primary job of this device is to block… | 5It Does The Job Well | 1377648000 | 08 28, 2013 |
| | | | | Nice windscreen protects | 5GOOD WINDSCREEN | | |

Amazon Musical Instruments Reviews dataset

# OUR IMPLEMENTATION

## PRE-PROCESSING

### TEXT CLEANING & BINARY LABELING

### TEXT CLEANING

**reviewText**: we deleted analphabetic signs, lowercased, lemmatized and removed stopwords [NLTK].

### BINARY LABELING

**Overall**: we grouped the 5 possible scores: {1,2,3} = 0 and {4,5} = 1.

| reviewerID | asin | reviewerName | helpful | reviewText | Overall summary | unixReviewTime | reviewTime |
|---|---|---|---|---|---|---|---|
| A2IBPI20UZIR0U | 13847192 | cassandra tu " Yeah, well, that's just like, u... | [0, 0] | Not much to write about here, … | 5good | 1393545600 | 02 28, 2014 |
| A14VAT5EAX3D9S | 138471932 | Jake | [13, 14] | The product does exactly as it should and is quite affordable... | 5Jake | 1363392000 | 03 16, 2013 |
| A195EZSQDW3E21 | 138471932 | Rick Bennette " Rick Bennette" | [1, 1] | The primary job of this device is to block… | 5It Does The Job Well | 1377648000 | 08 28, 2013 |
| | | | | Nice windscreen protects | 5GOOD WINDSCREEN | | |

Amazon Musical Instruments Reviews dataset

# OUR IMPLEMENTATION

## PRE-PROCESSING

### TEXT CLEANING & BINARY LABELING

#### TEXT CLEANING

**reviewText**: we deleted analphabetic signs, lowercased, lemmatized and removed stopwords [NLTK].

#### BINARY LABELING

**Overall**: we grouped the 5 possible scores: {1,2,3} = 0 and {4,5} = 1.

| reviewerID | asin | reviewerName | helpful | reviewText | Overall summary | unixReviewTime | reviewTime |
|---|---|---|---|---|---|---|---|
| A2IBPI20UZIR0U | 13847192 | cassandra tu " Yeah, well, that's just like, u... | [0, 0] | Not much to write about here, … | 5good | 1393545600 | 02 28, 2014 |
| A14VAT5EAX3D9S | 138471932 | Jake | [13, 14] | The product does exactly as it should and is quite affordable... | 5Jake | 1363392000 | 03 16, 2013 |
| A195EZSQDW3E21 | 138471932 | Rick Bennette " Rick Bennette" | [1, 1] | The primary job of this device is to block… | 5It Does The Job Well | 1377648000 | 08 28, 2013 |
| | | | | Nice windscreen protects | 5GOOD WINDSCREEN | | |

Amazon Musical Instruments Reviews dataset

# OUR IMPLEMENTATION

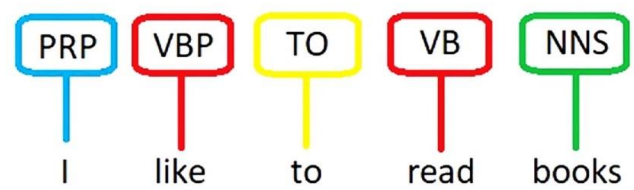
## PRE-PROCESSING

TEXT CLEANING & BINARY LABELING

POS TAGGING & TF-IDF

### POS TAGGING

Each word is POS-tagged context-aware, **[NLTK]**, to be divided into two subgroups: **aspect terms= nouns**, and **sentimental words=adj+adv**.




### TF-IDF

The whole corpus is vectorized word by word **[scikit-learn]** then, the 160 most relevant aspect terms, and the 160 most relevant sentimental words are selected by **max(w1, w2)**.

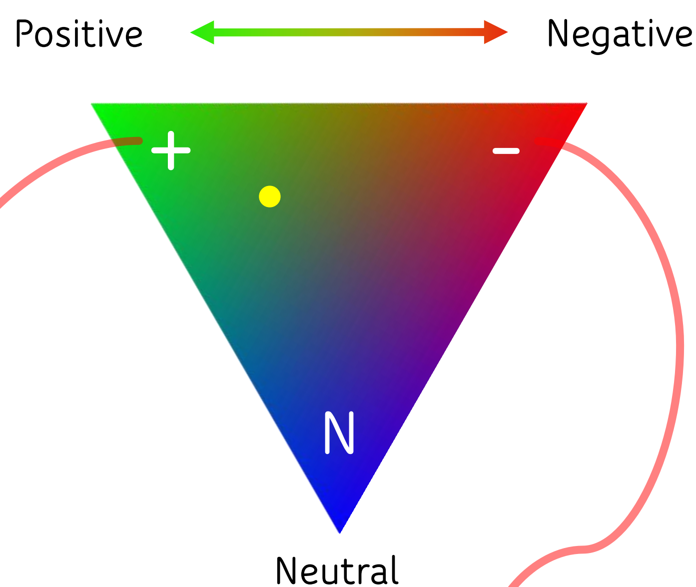
$$W_{i,j} = tf_i \times \left[\log \frac{1+N}{1+df_{i,j}} + 1\right]$$

# OUR IMPLEMENTATION

## PRE-PROCESSING

TEXT CLEANING & BINARY LABELING

SENTIMENT LEXICON




$$SW = 0.5 \times (pos - neg + 2)$$

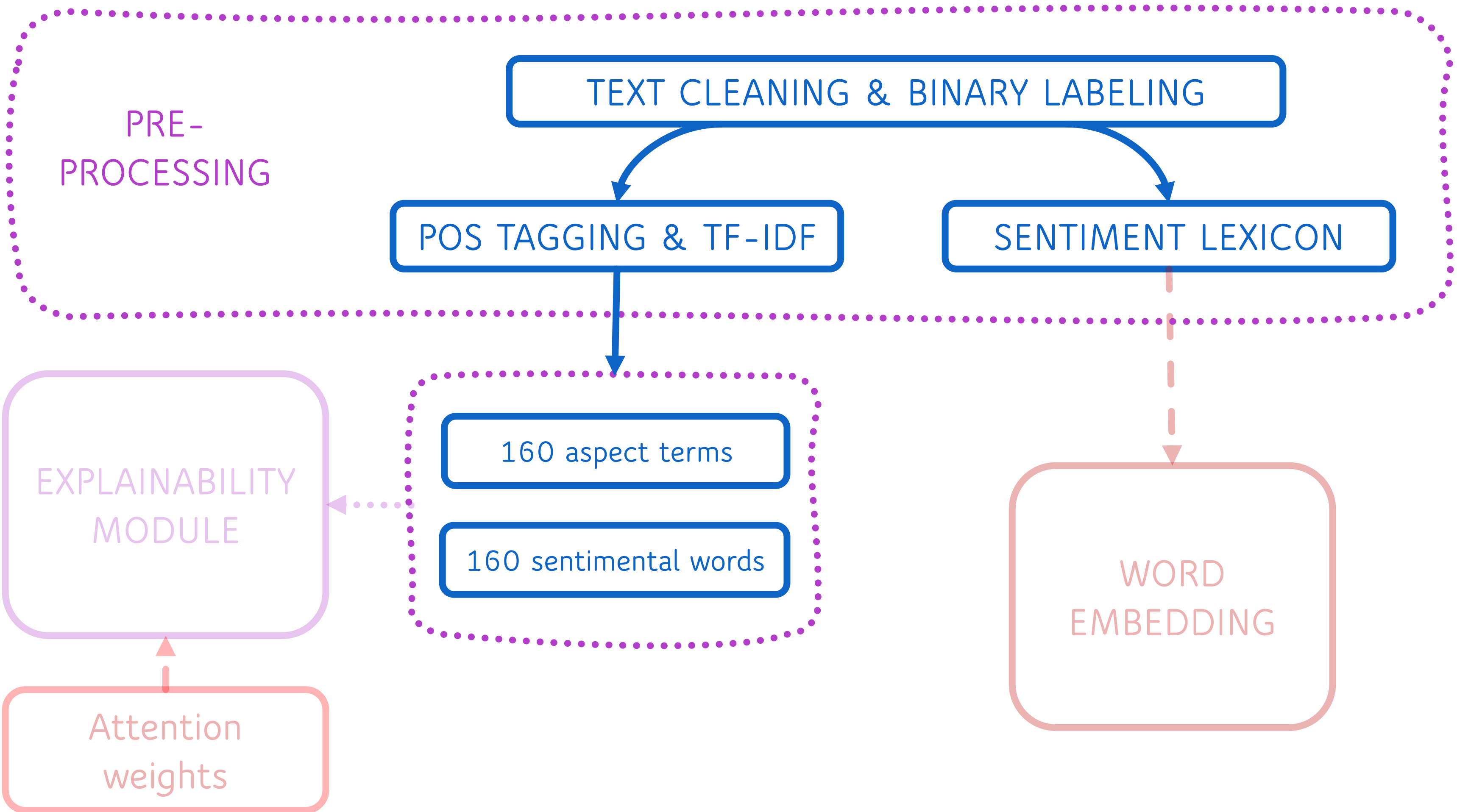
### SentiWordNet

Outputs **sentiment triplets={positive, negative, neutral}**, for each word, context-aware. We combine them, so that the resulting **sentiment weight** is in [0.5, 1.5].

We **assign a weight of 1** to words not present and we **average** the instances of weighted words.

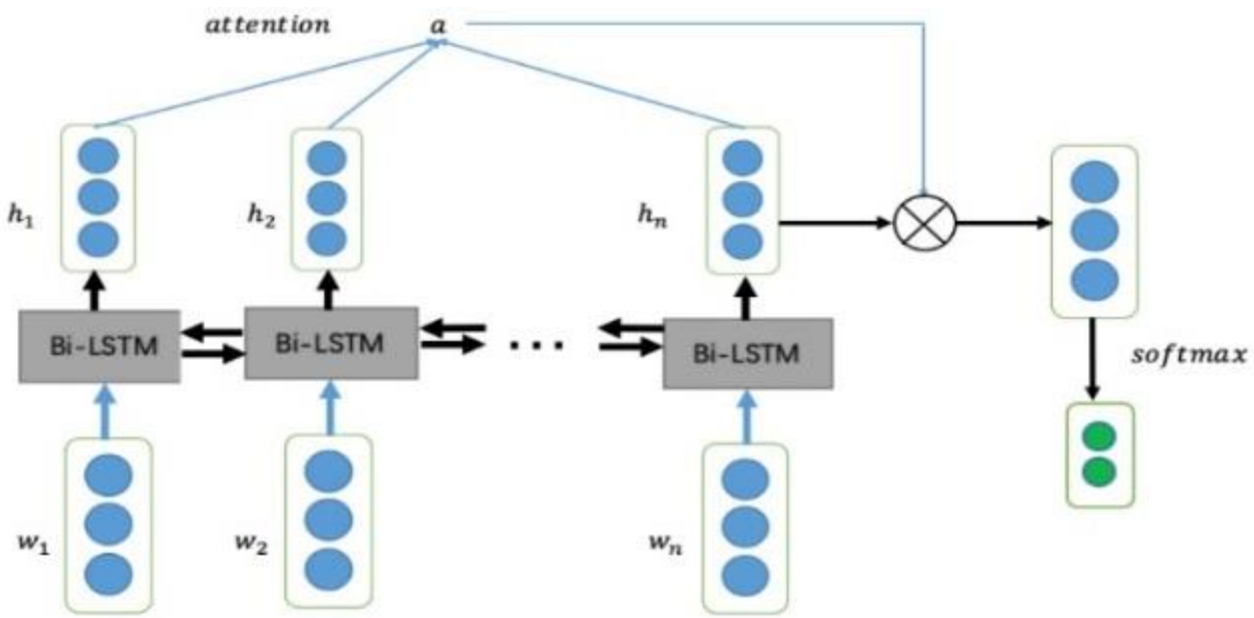
# OUR IMPLEMENTATION

## PRE-PROCESSING



# OUR IMPLEMENTATION

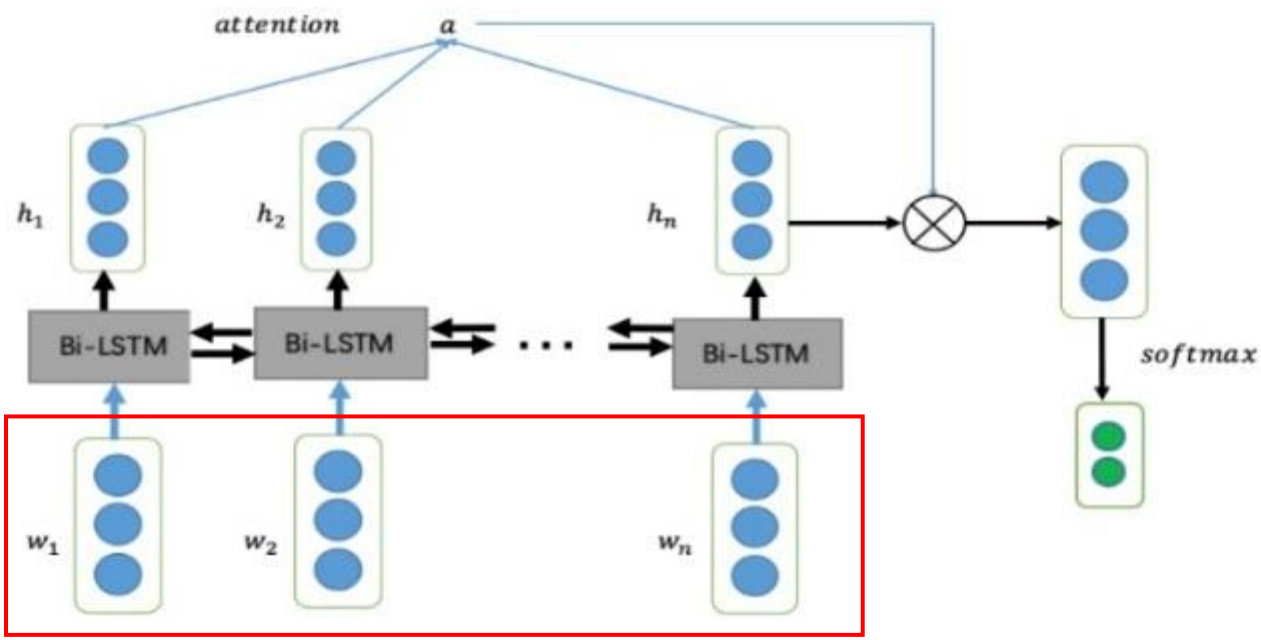
## MODEL ARCHITECTURE




- Word Embedding layer
- BiLSTM layer
- Attention layer
- Softmax classifier

# OUR IMPLEMENTATION

## MODEL ARCHITECTURE




- Word Embedding layer
- BiLSTM layer
- Attention layer
- Softmax classifier

BERT word embedding

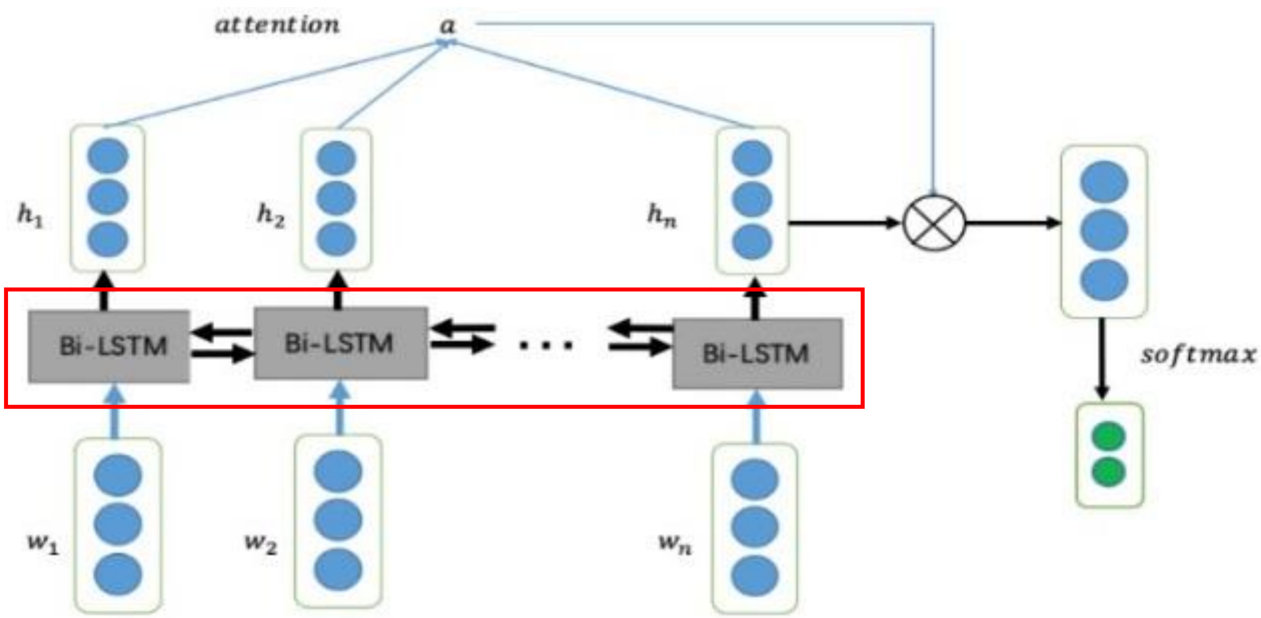
Encode input reviews into a suitable and **context-aware** representation



Furnari, Gallipoli, Tasca

# OUR IMPLEMENTATION

## MODEL ARCHITECTURE



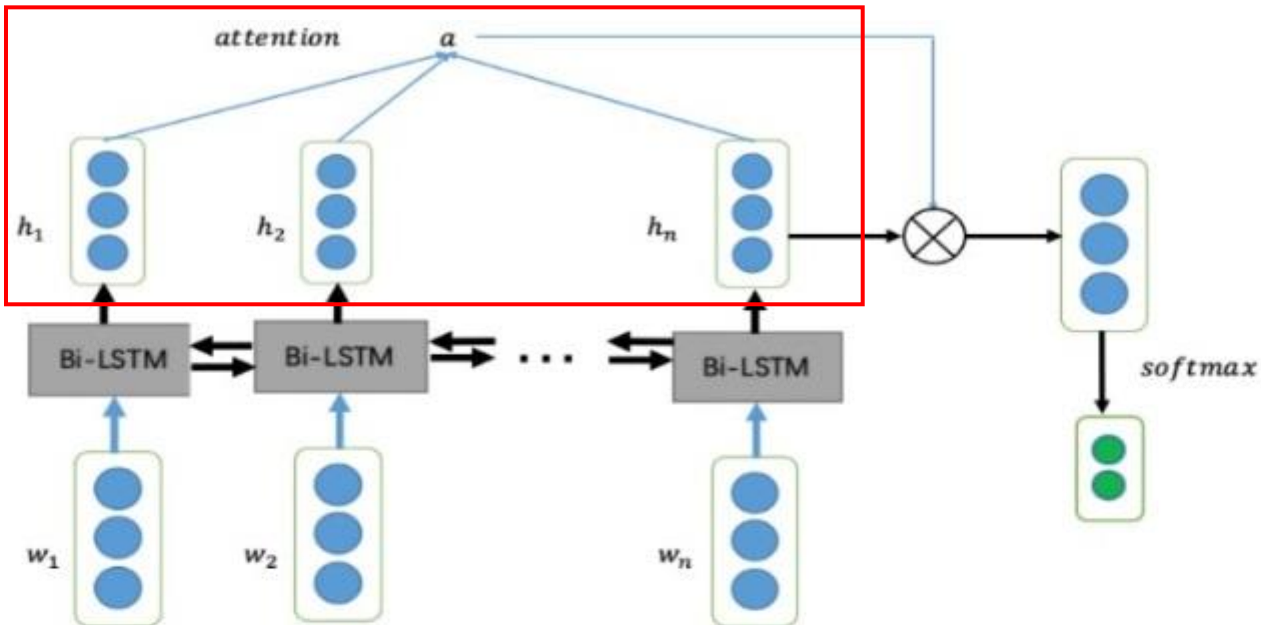

- Word Embedding layer
- BiLSTM layer
- Attention layer
- Softmax classifier

Process word embeddings across subsequent time steps in a **bidirectional** way

Take into account both the preceding and the subsequent word

# OUR IMPLEMENTATION

## MODEL ARCHITECTURE




- Word Embedding layer
- BiLSTM layer
- Attention layer
- Softmax classifier

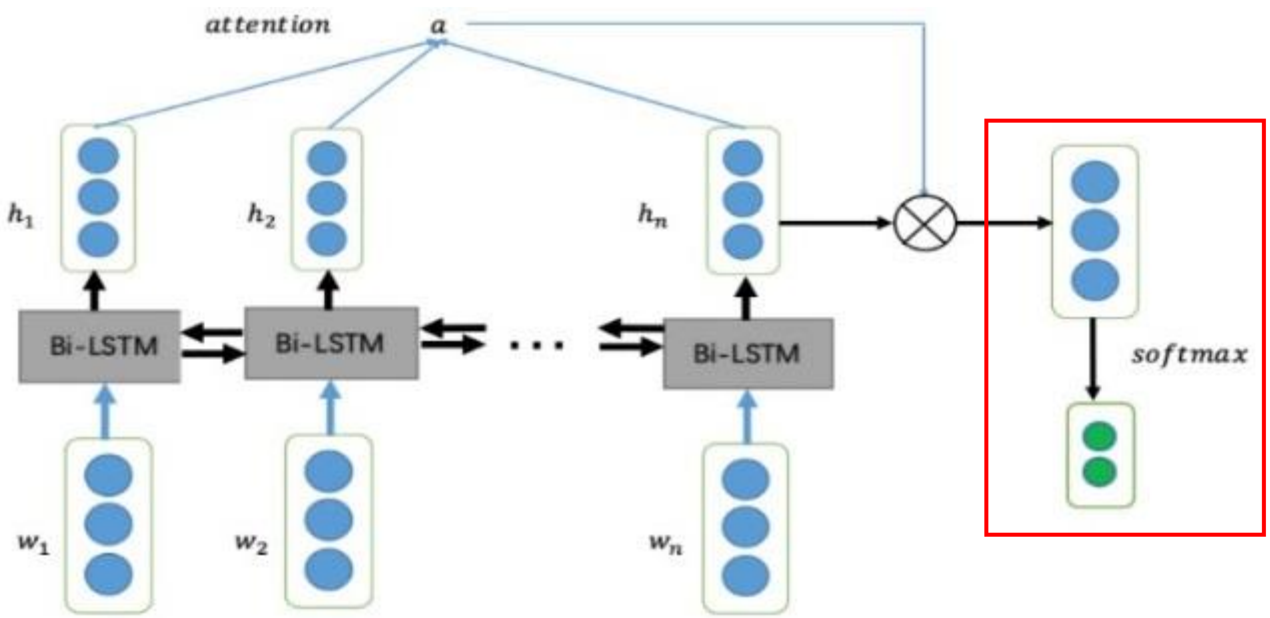
Self-attention mechanism

Identify the words on which the network "**focuses more**"

Important for **model interpretability**

# OUR IMPLEMENTATION

## MODEL ARCHITECTURE




- Word Embedding layer
- BiLSTM layer
- Attention layer
- Softmax classifier

Binary classification

Softmax classifier to produce class probabilities

# OUR IMPLEMENTATION

## MODEL TRAINING AND EVALUATION

Loss function

$$\mathcal{L} = \frac{1}{2}CSE + \frac{1}{2}MSE$$

combination of:

- Cross Entropy (CSE)
- Mean Square Error (MSE)

# OUR IMPLEMENTATION

## MODEL TRAINING AND EVALUATION

Loss function

$$\mathcal{L} = \frac{1}{2}CSE + \frac{1}{2}MSE$$

combination of:

- Cross Entropy (CSE)
- Mean Square Error (MSE)

Hyperparameters tuning

- Number of epochs
- Batch size
- Dropout rate

# OUR IMPLEMENTATION

## MODEL TRAINING AND EVALUATION

Loss function

$$\mathcal{L} = \frac{1}{2}CSE + \frac{1}{2}MSE$$

combination of:

- Cross Entropy (CSE)
- Mean Square Error (MSE)

Hyperparameters tuning

- Number of epochs
- Batch size
- Dropout rate

Metrics

- Accuracy
- Precision
- Recall
- $F_1$ score

# OUR IMPLEMENTATION

## MODEL TRAINING AND EVALUATION

Loss function

$$\mathcal{L} = \frac{1}{2}CSE + \frac{1}{2}MSE$$

combination of:

- Cross Entropy (CSE)
- Mean Square Error (MSE)

Hyperparameters tuning

- Number of epochs
- Batch size
- Dropout rate

Metrics

- Accuracy
- Precision
- Recall
- $F_1$ score

Original paper: (probably) metrics only on positive class

Our experiments: metrics both on positive class and **aggregated** with `macro` average

# OUR IMPLEMENTATION

## RESULTS – HYPERPARAMETERS TUNING

Number of epochs

| n_epochs | batch_size | dropout_rate | macro $F_1$ | $F_{1_{pos}}$ |
|---|---|---|---|---|
| 8 | 32 | 0.2 | 0.593 | 0.929 |
| 10 | 32 | 0.2 | 0.620 | 0.923 |
| 12 | 32 | 0.2 | 0.535 | 0.933 |
| 17 | 32 | 0.2 | 0.614 | 0.904 |

Batch size

| n_epochs | batch_size | dropout_rate | macro $F_1$ | $F_{1_{pos}}$ |
|---|---|---|---|---|
| 10 | 32 | 0.2 | 0.620 | 0.923 |
| 10 | 34 | 0.2 | 0.663 | 0.931 |
| 10 | 36 | 0.2 | 0.597 | 0.928 |
| 10 | 38 | 0.2 | 0.616 | 0.929 |

Dropout rate

| n_epochs | batch_size | dropout_rate | macro $F_1$ | $F_{1_{pos}}$ |
|---|---|---|---|---|
| 10 | 34 | 0.2 | 0.663 | 0.931 |
| 10 | 34 | 0.4 | 0.614 | 0.924 |
| 10 | 34 | 0.6 | 0.540 | 0.930 |
| 10 | 34 | 0.8 | 0.497 | 0.935 |

Best configuration: `n_epochs=10`, `batch_size=34`, `dropout_rate=0.2`

# OUR IMPLEMENTATION

## RESULTS – BASELINES

Baselines

- LSTM
- SVM
- Multinomial Naïve Bayes

# OUR IMPLEMENTATION

## RESULTS – BASELINES

Baselines

- LSTM
- SVM
- Multinomial Naïve Bayes

Results on test set

| baseline | $a$ | $p_{pos}$ | $r_{pos}$ | $F_{1_{pos}}$ | macro $F_1$ |
|---|---|---|---|---|---|
| BiLSTM + Attention | 0.943 | 0.916 | 0.988 | 0.951 | **0.687** |
| SVM | 0.913 | 0.881 | 0.979 | 0.927 | 0.489 |
| NB | 0.910 | 0.874 | 0.981 | 0.924 | 0.468 |
| LSTM | 0.908 | 0.910 | 0.922 | 0.916 | 0.615 |

# OUR IMPLEMENTATION

## RESULTS – BASELINES

Baselines

- LSTM
- SVM
- Multinomial Naïve Bayes

Results on test set

| baseline | $a$ | $p_{pos}$ | $r_{pos}$ | $F_{1\,pos}$ | macro $F_1$ |
|---|---|---|---|---|---|
| BiLSTM + Attention | 0.943 | 0.916 | 0.988 | 0.951 | **0.687** |
| SVM | 0.913 | 0.881 | 0.979 | 0.927 | 0.489 |
| NB | 0.910 | 0.874 | 0.981 | 0.924 | 0.468 |
| LSTM | 0.908 | 0.910 | 0.922 | 0.916 | 0.615 |

Possible cause: highly **unbalanced** dataset ⟶ predictions **biased** towards positive class

# OUR IMPLEMENTATION

## RESULTS – BASELINES

Baselines

- LSTM
- SVM
- Multinomial Naïve Bayes

Results on test set

| baseline | $a$ | $p_{pos}$ | $r_{pos}$ | $F1_{pos}$ | macro $F_1$ |
|---|---|---|---|---|---|
| BiLSTM + Attention | 0.943 | 0.916 | 0.988 | 0.951 | **0.687** |
| SVM | 0.913 | 0.881 | 0.979 | 0.927 | 0.489 |
| NB | 0.910 | 0.874 | 0.981 | 0.924 | 0.468 |
| LSTM | 0.908 | 0.910 | 0.922 | 0.916 | 0.615 |

Possible cause: highly **unbalanced** dataset ⟶ predictions **biased** towards positive class

Apply the model on a balanced dataset ⟶ Extension I

# OUR IMPLEMENTATION

## MODEL EXPLAINABILITY

Two levels of granularity:

- **Sentence-level**: attention weights distribution over test reviews
  identify which **words** in a review have greater importance

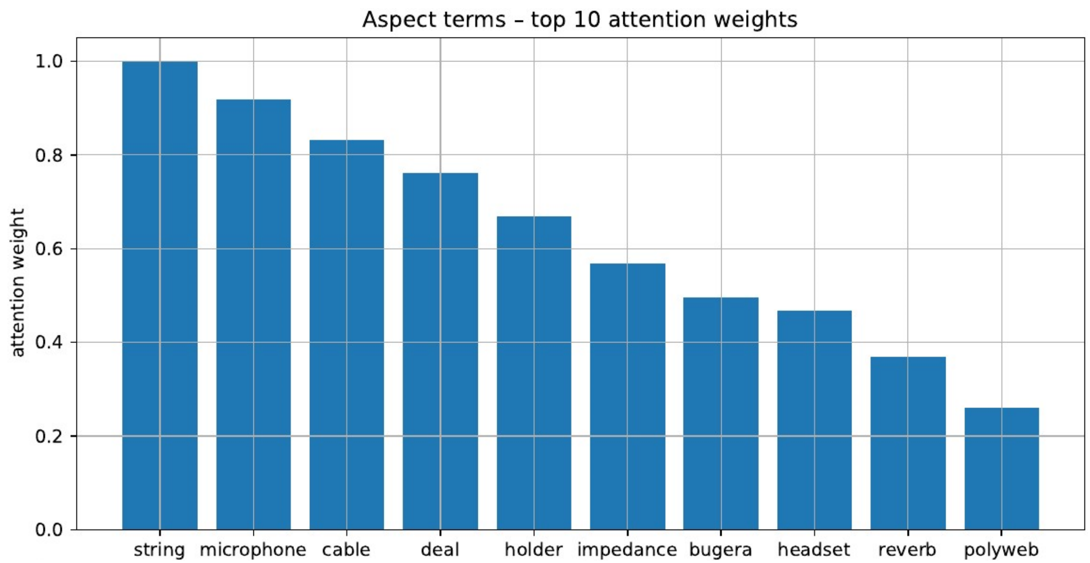
| ix | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **word** | nice | cable | make | good | material | attractive | need | foot | cable | glad | store |
| **attention weight** | 0.245300 | 0.155488 | 0.014427 | 0.127360 | 0.081708 | 0.096103 | 0.037665 | 0.044435 | 0.093608 | 0.061604 | 0.042301 |

# OUR IMPLEMENTATION

## MODEL EXPLAINABILITY

Two levels of granularity:

- **Sentence-level**: attention weights distribution over test reviews
  identify which **words** in a review have greater importance
- **Corpus-level**: attention weights distribution over aspect and sentimental words
  identify most important **products**, **features** and **feelings**

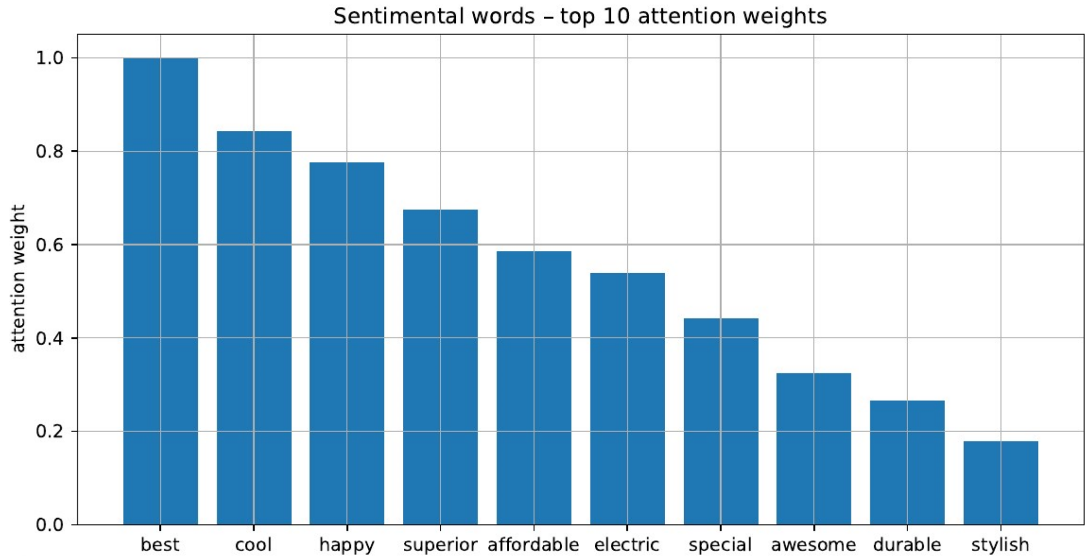


# OUR IMPLEMENTATION

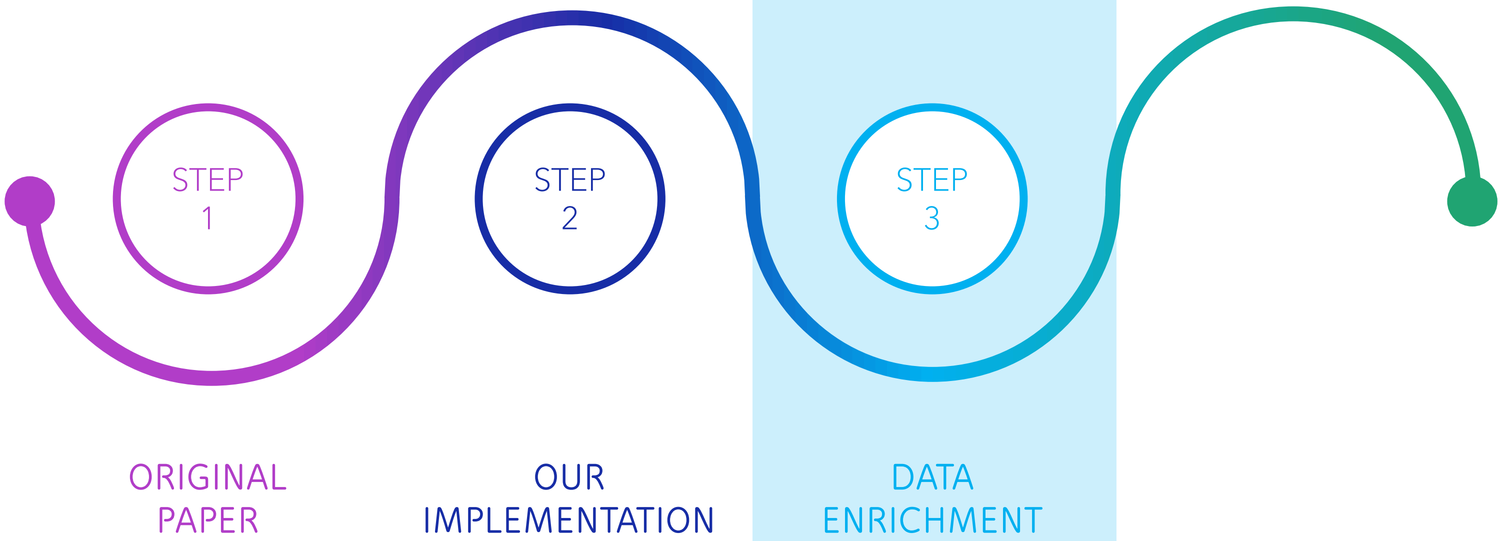
## MODEL EXPLAINABILITY

Two levels of granularity:

- **Sentence-level**: attention weights distribution over test reviews
  identify which **words** in a review have greater importance
- **Corpus-level**: attention weights distribution over aspect and sentimental words
  identify most important **products**, **features** and **feelings**



# WORKFLOW


STEP 1
STEP 2
STEP 3
ORIGINAL PAPER
OUR IMPLEMENTATION
DATA ENRICHMENT

# IMDB MOVIE REVIEWS DATASET

## DATASET DESCRIPTION AND PRE-PROCESSING

- Much bigger amount of records for training (but beware of computational constraints!)
- Perfect **balance** between the two sentiments
- Inspect model performance in a **different domain** and see effects in interpretability
- Removal of meaningless tokens such as HTML tags

# IMDB MOVIE REVIEWS DATASET

## HYPERPARAMETERS TUNING

Same dataset splitting and hyperparameters set

| n_epochs | batch_size | dropout_rate | macro $F_1$ | $F_{1_{pos}}$ |
|---|---|---|---|---|
| 8 | 32 | 0.2 | 0.863 | 0.860 |
| 10 | 32 | 0.2 | 0.853 | 0.845 |
| 12 | 32 | 0.2 | 0.842 | 0.839 |
| 8 | 34 | 0.2 | **0.867** | 0.860 |
| 8 | 34 | 0.4 | 0.864 | 0.862 |
| 10 | 34 | 0.2 | 0.853 | 0.846 |
| 12 | 34 | 0.2 | 0.843 | 0.851 |
| 17 | 34 | 0.2 | 0.854 | 0.845 |
| 8 | 36 | 0.2 | 0.861 | 0.865 |
| 10 | 36 | 0.2 | 0.851 | 0.847 |
| 12 | 36 | 0.2 | 0.854 | 0.848 |
| 8 | 38 | 0.2 | 0.861 | 0.867 |
| 10 | 38 | 0.2 | 0.864 | 0.860 |

Best configuration: `n_epochs=8`, `batch_size=34`, `dropout_rate=0.2`

# IMDB MOVIE REVIEWS DATASET

## RESULTS ON TEST SET

Same baselines

| | $a$ | $p_{pos}$ | $r_{pos}$ | $F_{1\,pos}$ | macro $F_1$ |
|---|---|---|---|---|---|
| BiLSTM + Attention | 0.886 | 0.866 | 0.894 | 0.880 | **0.885** |
| SVM | 0.810 | 0.785 | 0.832 | 0.808 | 0.806 |
| NB | 0.791 | 0.796 | 0.782 | 0.789 | 0.791 |
| LSTM | 0.831 | 0.812 | 0.861 | 0.836 | 0.829 |

We managed to reach a 20% improvement with respect to the original paper!

# IMDB MOVIE REVIEWS DATASET

## SENTENCE ATTENTION WEIGHTS

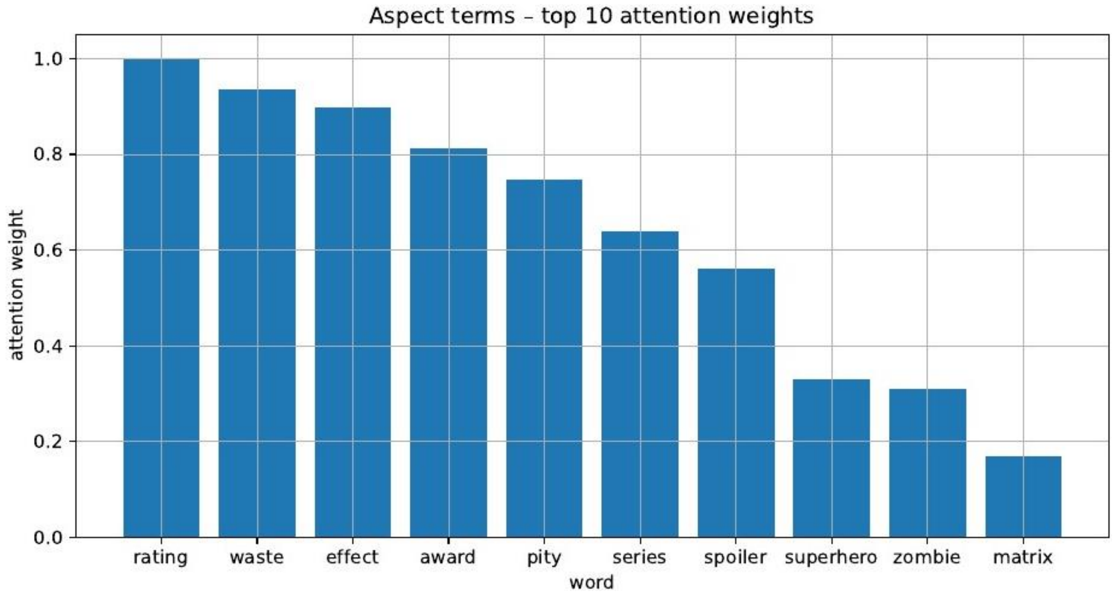
| ix | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **word** | davis | electrify | performance | hard | remember | female | player | perfect | part | wonderful |
| **attention weight** | 0.047242 | 0.132870 | 0.202740 | 0.018956 | 0.038704 | 0.055603 | 0.108019 | 0.127337 | 0.150193 | 0.118335 |

Higher weights are assigned to:

- Nouns
- Adjectives which express a certain sentiment

# IMDB MOVIE REVIEWS DATASET

## ASPECT TERMS ATTENTION WEIGHTS

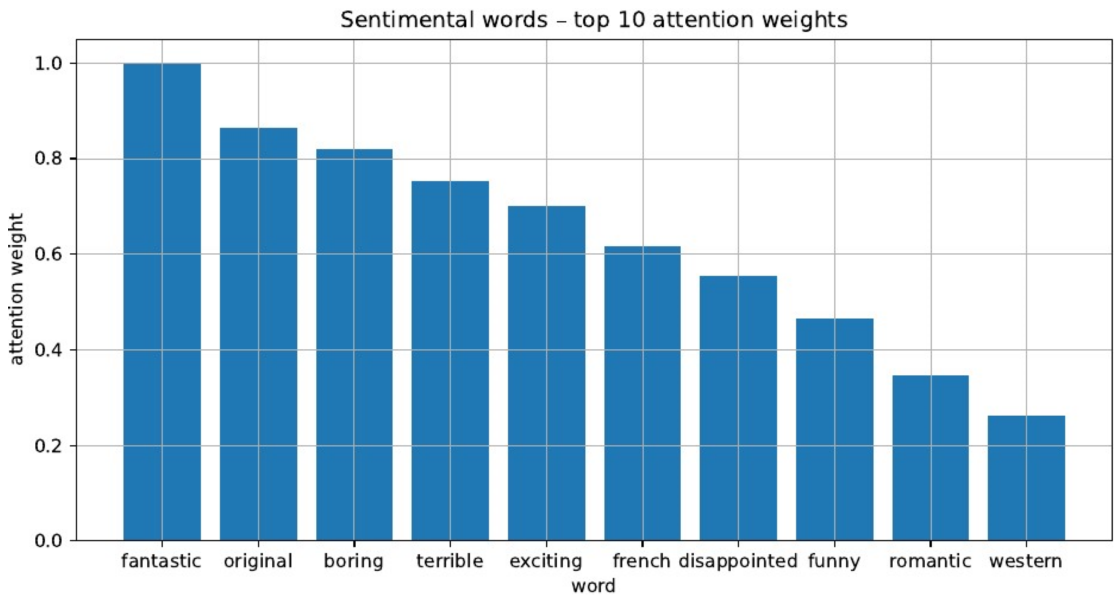



Distribution is concentrated at:

- Cinema industry jargon
- Movie characters and titles

# IMDB MOVIE REVIEWS DATASET

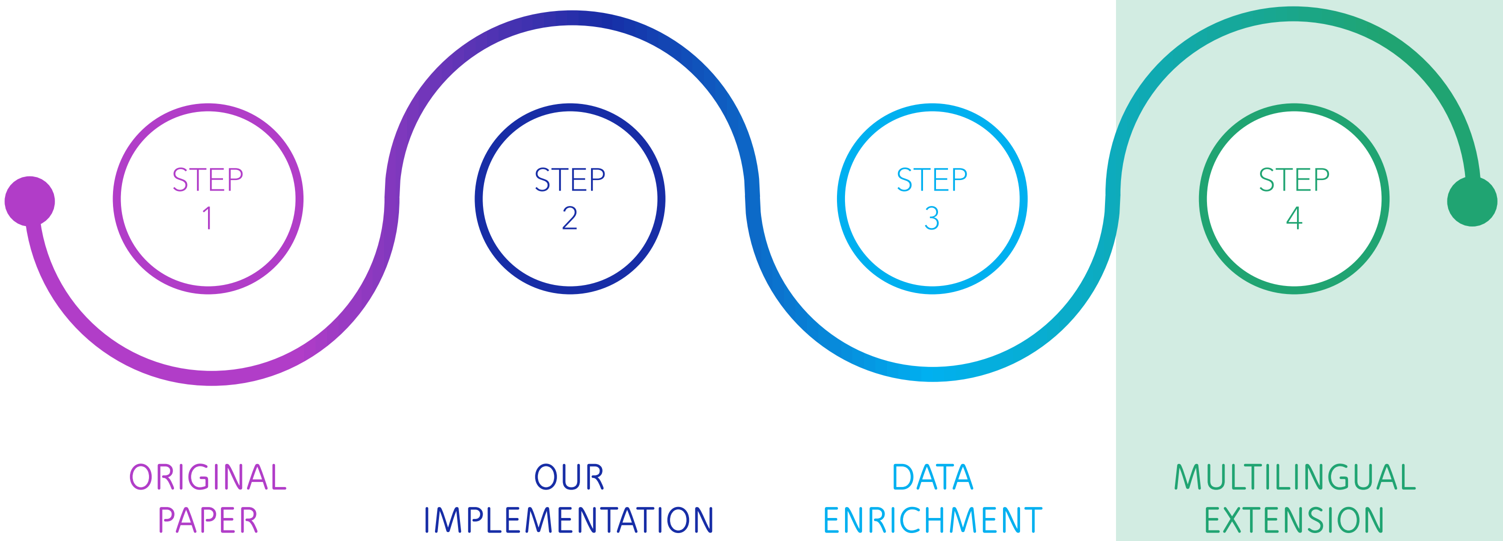
## SENTIMENTAL WORDS ATTENTION WEIGHTS




Distribution is concentrated at:

- Adjectives conveying feeling (both positive and negative!)
- Movie genres, film/actor nationality

# WORKFLOW


STEP 1
STEP 2
STEP 3
STEP 4
ORIGINAL PAPER
OUR IMPLEMENTATION
DATA ENRICHMENT
MULTILINGUAL EXTENSION

# MULTILINGUAL AMAZON REVIEWS CORPUS

## DATASET DESCRIPTION AND PRE-PROCESSING

- 6 languages and many records for each (we have to restrict the focus)
- Filtering and label assignment in order to get a comparable perfectly balanced dataset
- Multilingual approach based on translation to a **target language**
- Exploitation of a pre-trained **Neural Machine Translation** model

# MULTILINGUAL AMAZON REVIEWS CORPUS

## HYPERPARAMETERS TUNING

Same dataset splitting and hyperparameters set

| n_epochs | batch_size | dropout_rate | macro $F_1$ | $F_{1_{pos}}$ |
|---|---|---|---|---|
| 8 | 32 | 0.2 | 0.816 | 0.815 |
| 10 | 32 | 0.2 | 0.804 | 0.801 |
| 12 | 32 | 0.2 | 0.797 | 0.806 |
| 8 | 34 | 0.2 | 0.830 | 0.831 |
| 8 | 34 | 0.4 | 0.830 | 0.822 |
| 10 | 34 | 0.2 | 0.821 | 0.821 |
| 12 | 34 | 0.2 | 0.807 | 0.818 |
| 8 | 36 | 0.2 | **0.835** | 0.835 |
| 8 | 36 | 0.4 | 0.824 | 0.817 |
| 10 | 36 | 0.2 | 0.823 | 0.818 |
| 12 | 36 | 0.2 | 0.811 | 0.821 |
| 8 | 38 | 0.2 | 0.824 | 0.821 |
| 10 | 38 | 0.2 | 0.811 | 0.816 |
| 12 | 38 | 0.2 | 0.801 | 0.809 |

Best configuration: `n_epochs=8`, `batch_size=36`, `dropout_rate=0.2`

# MULTILINGUAL AMAZON REVIEWS CORPUS

## RESULTS ON TEST SET

Same baselines

| | $a$ | $p_{pos}$ | $r_{pos}$ | $F1_{pos}$ | macro $F_1$ |
|---|---|---|---|---|---|
| BiLSTM + Attention | 0.852 | 0.850 | 0.852 | 0.851 | **0.854** |
| SVM | 0.782 | 0.787 | 0.775 | 0.781 | 0.782 |
| NB | 0.777 | 0.770 | 0.786 | 0.778 | 0.776 |
| LSTM | 0.813 | 0.801 | 0.832 | 0.816 | 0.812 |

Still better than the original paper, but slightly less with respect to Extension I (due to the additional task of machine translation)

# MULTILINGUAL AMAZON REVIEWS CORPUS

## SENTENCE ATTENTION WEIGHTS

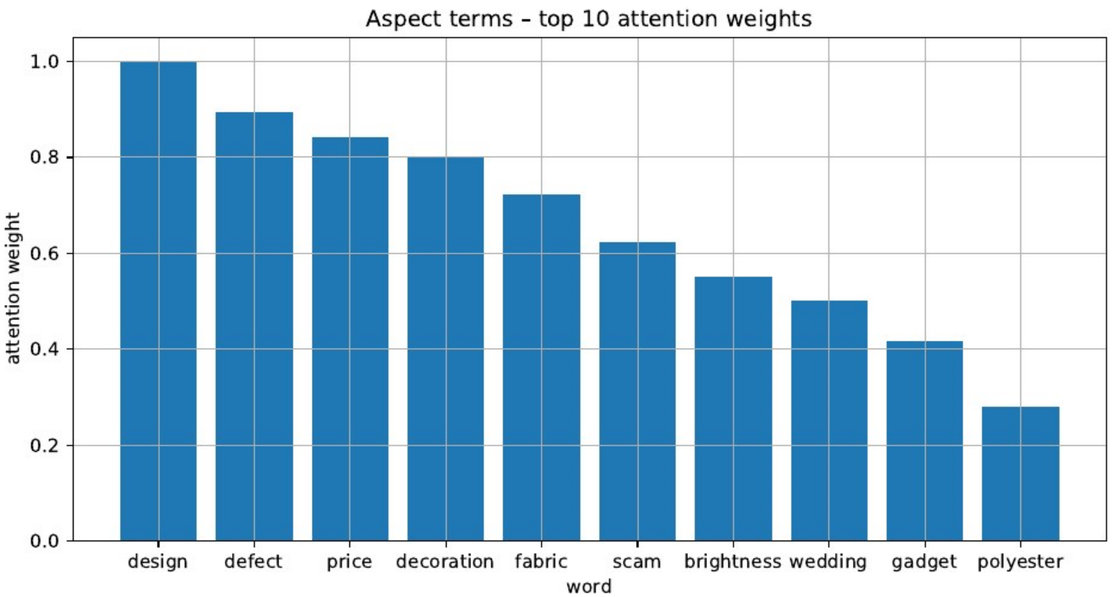
| ix | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **word** | disappointed | enough | coffee | mill | throw | half | away | not | recommend | product |
| **attention weight** | 0.158814 | 0.049000 | 0.119626 | 0.019194 | 0.060020 | 0.031756 | 0.129640 | 0.181143 | 0.156706 | 0.094101 |

Higher weights are assigned to:

- Words describing products
- Adjectives which express a certain sentiment

# MULTILINGUAL AMAZON REVIEWS CORPUS

## ASPECT TERMS ATTENTION WEIGHTS

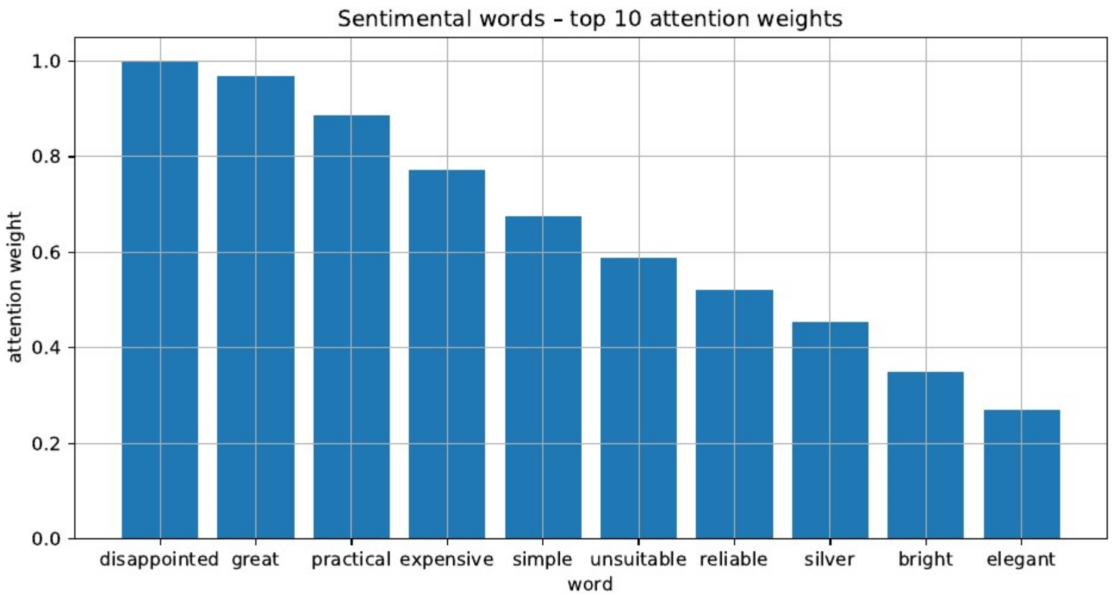



Distribution is concentrated at:

- Words describing product features
- Terms related to product category

# MULTILINGUAL AMAZON REVIEWS CORPUS

## SENTIMENTAL WORDS ATTENTION WEIGHTS




Distribution is concentrated at:

- Terms expressing the reviewer's opinion about the product
- Adjectives defining the qualities of the item

# CONCLUSIONS

## TAKEAWAYS

- Model gets poor results with the original dataset: alternative directions could go towards a direct handling of this issue
- Model is profitably adaptable to a **different domain** of reviews
- Thanks to Neural Machine Translation, model is greatly extended to a **multilingual context**
- **Interpretability analysis** shows in a nice and intuitive graphical way which terms are taken into account by the black-box to make predictions

# CONCLUSIONS

## MODEL USAGE & REPRODUCIBILITY

### Examples with Amazon Music dataset

#### Train

```
main.py train --dataset music --encoder bert --n_epochs <epochs> --batch_size <batch_size> --dropout_rate <dropout>
```

#### Test

```
main.py test  --dataset music --encoder bert --from_pretrained <pretrained_path>
```

#### Baselines

```
main.py baseline lstm --dataset music --encoder bert --n_epochs <epochs> --batch_size <batch_size> --dropout_rate <dropout>

main.py baseline  svm --dataset music

main.py baseline   nb --dataset music
```

All experiments are available on Google Colaboratory

GitHub repository: https://github.com/gallipoligiuseppe/SentiModel

# THANK YOU FOR YOUR ATTENTION