# Word Limit

**1. Emotion Can Be Expressed in Very Few Words**

Some emotional states can be expressed with **just one word**, like:

* *“Sad.”*
* *“Hopeless.”*
* *“Why?”*
* *“I give up.”*

Even **very short phrases** (2–6 words) can strongly indicate emotion:

* *“I can’t do this.”*
* *“Nothing makes sense anymore.”*
* *“Everything hurts.”*

So in theory, **5–10 words** could already signal **emotion**, **mood**, or **mental state** — especially if those words are emotionally charged.

**2. Research Backing (Short Text Emotion Detection)**

Studies in psychology and NLP show:

* Even **tweets** (~15–20 words max) can be used to **detect depression** or **anxiety** with reasonable accuracy.
* Labeled datasets like **GoEmotions** (by Google) use short Reddit comments (~10–30 words) for emotion classification across 27 emotions.

**Practical Rule of Thumb:**

| **Emotion Detection Level** | **Approx. Word Count** |
| --- | --- |
| Quick flag (e.g., "sad", "tired") | **1–5 words** |
| Emotion type + cause | **10–30 words** |
| Symptom severity + narrative | **50–100+ words** |

For your Reddit project:

* **Short posts (~10–30 words)** can already reveal **mental health signals**
* **Longer posts** give context, triggers, symptom progression

**If Using NLP Models:**

Models like BERT or RoBERTa can handle **short snippets very well**.  
Even a message like:

*“I’m done pretending I’m okay.”*  
…can be classified as **sadness, hopelessness, depression**.

**TL;DR:**

* **Minimum**: 1–5 words can signal emotion (e.g., “Help me.”)
* **Reliable detection**: 10–30+ words for models and analysis
* **Deeper insight**: 50–300 words gives full context of mental health

Would you like to:

* Detect emotion in short messages?
* Classify posts into symptom categories (e.g., anxiety, burnout, hopelessness)?
* Build a real-time emotion detector?

Let’s take the next practical step!

**You said:**

if i am plaanig to use bERT what would be maximum word I need to determine sypmotms?

| **Model** | **Max Word Count (approx.)** | **Max Token Limit** |
| --- | --- | --- |
| BERT (base/large) | ~350–400 words | **512 tokens** |

**Why the Limit?**

* BERT has a **maximum input of 512 tokens**.
* A token is **not the same as a word** — BERT splits text into subword tokens (WordPiece).
* So:
  + 512 tokens ≈ **350–400 words** (on average, depending on the text)

**Practical Guidelines for Symptom Detection**

| **Use Case** | **Recommended Max Words** | **Why** |
| --- | --- | --- |
| Emotion/Symptom classification (depression, anxiety) | **100–300 words** | Enough to capture emotion, context, symptom language |
| Reddit post classification | **250–400 words** | Captures full post without losing context |
| Multi-sentence reasoning (e.g., therapy intent, suicidal ideation) | **300–400 words** | Longer narrative often needed |

**What Happens If Text Is Longer?**

* You need to **truncate** or **split**:
  + text[:512] tokens if using basic BERT
* OR
  + Use **Longformer**, **BigBird**, or **RoBERTa + chunking** to handle longer text (1000+ tokens)

**TL;DR for BERT + Mental Health Symptoms:**

* Stick to **< 350–400 words** (≈ 512 tokens)
* That’s **enough to detect detailed symptoms**
* If longer, truncate or use models made for long texts

A screenshot of a computer

AI-generated content may be incorrect.

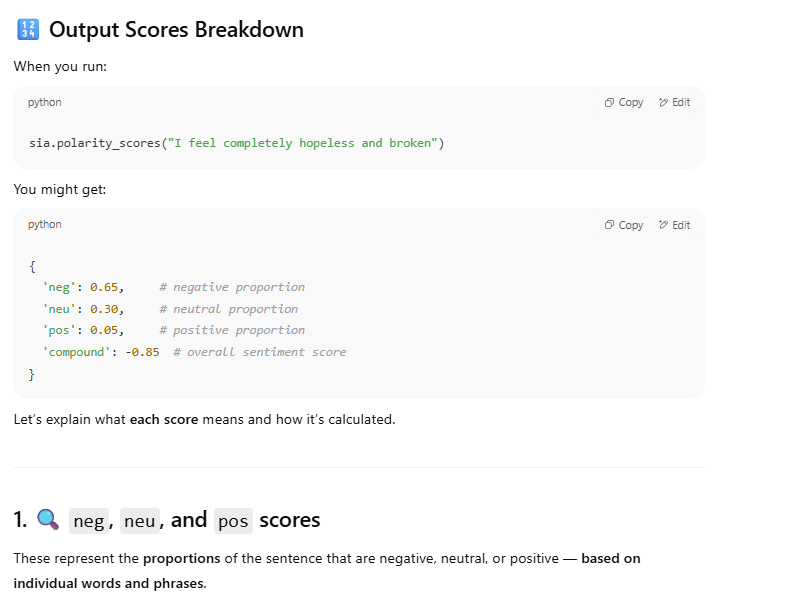
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Valence Aware Dictionary for Sentiment Reasoning



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a math test

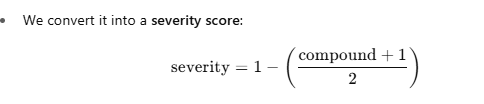
AI-generated content may be incorrect.

A white background with black text

AI-generated content may be incorrect.

A close-up of a number

AI-generated content may be incorrect.



A white text with black text

AI-generated content may be incorrect.

First thing I again goes for data preprocessing and the things I did were:

* + 1. I changed text for other languagses like devganari hindi language to text for making every text into englihs for further processing and this is done by:

from translatepy import Translator

translator = Translator()

text = "mujhe marna hai abhi koi tarika batao kaise maru bina dard ke"

result = translator.translate(text, "English")

print(result.result)

output:

I want to die right now tell me how to die without pain

* + 1. Then

**🔢 Output Scores Breakdown**

When you run:

python

CopyEdit

sia.polarity\_scores("I feel completely hopeless and broken")

You might get:

python

CopyEdit

{

'neg': 0.65, # negative proportion

'neu': 0.30, # neutral proportion

'pos': 0.05, # positive proportion

'compound': -0.85 # overall sentiment score

}

Let’s explain what **each score** means and how it’s calculated.

**1. 🔍 neg, neu, and pos scores**

These represent the **proportions** of the sentence that are negative, neutral, or positive — **based on individual words and phrases**.

**✅ How VADER does this:**

1. It **tokenizes** the text (splits it into words/phrases).
2. For each token, it **looks up a valence score** from its sentiment lexicon (dictionary of about 7,500 terms).
   * Example: "hopeless" → -2.4, "broken" → -1.7
3. It applies **rules** to adjust scores:
   * Intensifiers (e.g. "very", "extremely")
   * Negations (e.g. "not good")
   * Punctuation (e.g. "!!!", "??")
   * Capitalization
   * Conjunctions (e.g. “but”, “however”)
4. It calculates a **total sentiment score** for the sentence.
5. Finally, it computes:

python

CopyEdit

neg = total\_negative / total\_abs

neu = total\_neutral / total\_abs

pos = total\_positive / total\_abs

Where total\_abs = total\_negative + total\_neutral + total\_positive

So for:

arduino

CopyEdit

"I feel completely hopeless and broken"

Assume the scores assigned were:

* "hopeless" → -2.4
* "broken" → -1.7
* "feel" → neutral
* "completely" → intensifier (+15%)  
  The adjusted negative score would be higher, giving something like:

python

CopyEdit

neg = 0.65

neu = 0.30

pos = 0.05

**2. 🎯 compound score (most important)**

This is a **single summary score** from **-1 (most negative)** to **+1 (most positive)**.

**✅ How VADER calculates it:**

1. Take **all word valence scores** after applying rules.
2. **Sum** the adjusted valences → sum\_v.
3. Normalize using this formula:

compound=sum\_vsum\_v2+α\text{compound} = \frac{\text{sum\\_v}}{\sqrt{ \text{sum\\_v}^2 + \alpha }}compound=sum\_v2+α​sum\_v​

Where α = 15 (a normalization constant that controls the curve).

**🔢 Example:**

Let’s say the total valence score of your sentence is **-4.5** (after all adjustments):

compound=−4.5(−4.5)2+15=−4.520.25+15=−4.535.25≈−4.55.94≈−0.757\text{compound} = \frac{-4.5}{\sqrt{(-4.5)^2 + 15}} = \frac{-4.5}{\sqrt{20.25 + 15}} = \frac{-4.5}{\sqrt{35.25}} \approx \frac{-4.5}{5.94} \approx -0.757compound=(−4.5)2+15​−4.5​=20.25+15​−4.5​=35.25​−4.5​≈5.94−4.5​≈−0.757

So, the **compound score** becomes approximately -0.76.

severity = 1 - ((compound - min) / (max - min))

At first I did severity calculation according to sub reddit class but what I found that:

Subreddit-level Averaging Issues:

You're assigning all posts in a subreddit the same "severity" during training.

This leads to weak supervision — the model is learning subreddit characteristics more than true emotional severity.

Label Leakage & Overfitting:

The subreddit-level severity (mean sentiment) is used as the target — but this may generalize poorly when a new post from the same subreddit has a very different emotional tone.

So I Switch to Per-Post Labeling Using VADER

Assigning the same severity score to all posts within a subreddit — based on the average sentiment of that subreddit — causes the model to learn subreddit-specific traits rather than the severity of each individual post. This leads to weak supervision and reduces the model’s ability to detect the true emotional severity on a per-post basis.

Then

I switch to use Fine-Tune MentalBERT (mental-bert-base-uncased) and it is under process

A screenshot of a computer

AI-generated content may be incorrect.

**How to incorporate all three**

You can calculate **severity** using more than just compound. One approach:

1. Get neg, pos, and neu from VADER.
2. Define severity as higher when:
   * neg is high
   * or emotional polarity (max(pos, neg)) is high
   * or emotional fluctuation (pos and neg both high) is present.

**Pipeline integrating BERT/RoBERTa + LIWC/DLATK + Emotion Mapping for Severity**

**1. Prepare data for psycholinguistic tools**

* Create unique message\_id and rename your text column (e.g., processed\_text → message)
* Save to CSV for DLATK processing as you did

**2. Extract psycholinguistic features with LIWC/DLATK**

* Run LIWC via DLATK on your text to get features like negemo, posemo, anx, cogproc etc.
* Load the output CSV and merge with your main dataframe

**3. Use pretrained BERT/RoBERTa for sentiment and emotion**

* Fine-tune or use pretrained models (e.g., HuggingFace transformers) for:
  + Sentiment classification (positive, negative, neutral)
  + Emotion detection (joy, sadness, anger, fear, etc.)
  + Mental health symptom detection (if you have labeled data)
* For each post, generate:
  + Sentiment/emotion probabilities or scores
  + Mental health risk scores (if applicable)

**4. Map detected emotions to the Circumplex Model of Affect**

* Convert emotion probabilities to **valence** and **arousal** scores (weighted sum based on known mappings, e.g., joy → high valence/high arousal, sadness → low valence/low arousal)
* This gives a 2D affect representation per post

**5. Combine psycholinguistic + emotion + BERT-based features**

* Normalize and possibly weight the LIWC features, valence/arousal scores, and BERT outputs
* Example severity score formula (customized):

python

CopyEdit

severity = w1 \* LIWC\_negemo + w2 \* (1 - valence) + w3 \* arousal + w4 \* bert\_mental\_health\_score

* Weights (w1, w2, w3, w4) can be tuned based on validation data or domain knowledge

**6. Normalize final severity scores**

* Scale severity between 0 and 1 for interpretability
* You can apply min-max scaling or logistic functions

**7. Save and analyze results**

* Save the enriched dataframe with severity scores and all extracted features
* Perform downstream analysis, visualization, or use in predictive models