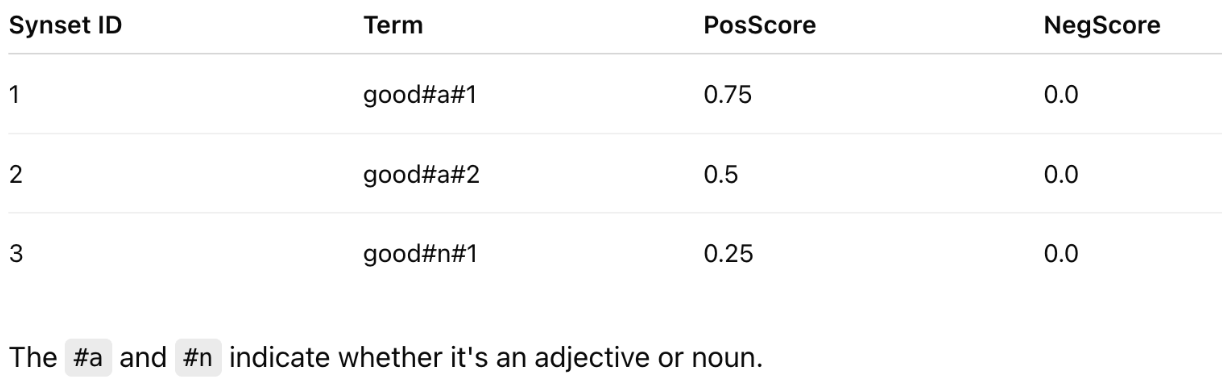
https://github.com/fnielsen/afinn/blob/master/afinn/data/AFINN-111.txt

SentiWordNet is built on **WordNet**, a large lexical database of English, and assigns **three sentiment scores** to each **synset** (a set of synonyms representing a concept):

- **PosScore** (positive sentiment): between 0 and 1
- **NegScore** (negative sentiment): between 0 and 1
- **ObjScore** (objective/neutral sentiment): implicitly defined as `1 - (PosScore + NegScore)`

Each word can have **multiple senses**, each with different scores.



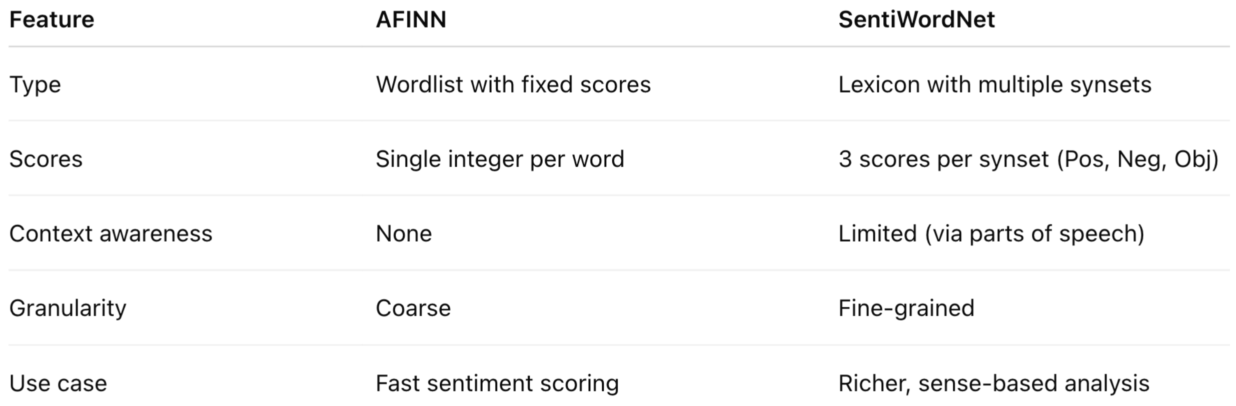
Synset ID	Term	PosScore	NegScore
1	good#a#1	0.75	0.0
2	good#a#2	0.5	0.0
3	good#n#1	0.25	0.0

The `#a` and `#n` indicate whether it's an adjective or noun.

```
import nltk
from nltk.corpus import sentiwordnet as swn
nltk.download('sentiwordnet')
nltk.download('wordnet')
```

```
list(swn.senti_synsets('good', 'a'))
```

```
syn = list(swn.senti_synsets('good', 'a'))[0]
print("Pos:", syn.pos_score())
print("Neg:", syn.neg_score())
print("Obj:", syn.obj_score())
```



Feature	AFINN	SentiWordNet
Type	Wordlist with fixed scores	Lexicon with multiple synsets
Scores	Single integer per word	3 scores per synset (Pos, Neg, Obj)
Context awareness	None	Limited (via parts of speech)
Granularity	Coarse	Fine-grained
Use case	Fast sentiment scoring	Richer, sense-based analysis

Use SentiWordNet if:

- You want more **linguistic depth**.
- You care about **sense disambiguation** (e.g., "bright" = intelligent vs. full of light).
- You are analyzing **longer texts** where multiple senses of a word are possible.

If you want a fast, interpretable method, **AFINN** is simpler.

VADER analyzes a sentence and returns a **dictionary of 4 sentiment scores**:

- `pos`: probability of the text being **positive**
- `neu`: probability of the text being **neutral**
- `neg`: probability of the text being **negative**
- `compound`: a **normalized score** from -1 (most negative) to +1 (most positive)

VADER uses:

- A lexicon of ~7,500 words with associated sentiment intensities.
- Rules to handle punctuation, capitalization, intensifiers (e.g., "very"), negations, emojis, and even slang.

```
pip install nltk
```

```
import nltk
nltk.download('vader_lexicon')
```

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# Create the analyzer
sia = SentimentIntensityAnalyzer()

# Your text
text = "I really love this amazing movie! It's so inspiring."

# Get the sentiment scores
scores = sia.polarity_scores(text)

# Print the result
print(scores)
```

https://github.com/cjhutto/vaderSentiment/blob/master/vaderSentiment/vader_lexicon.txt

Example:

"The movie was absolutely amazing, but the ending was a bit disappointing."

Step-by-step Breakdown:

1. Identify sentiment-bearing words

From the sentence:

- **"absolutely amazing"** → strong intensifier + positive
- **"a bit disappointing"** → mild modifier + negative

2. Apply VADER-like rules

VADER doesn't just average the sentiment values — it:

- Boosts intensity with **modifiers** (like *"absolutely"*)
- Dampens intensity with **downtoners** (like *"a bit"*)
- Accounts for **contrastive conjunctions** (like *"but"*)

Step 1: Analyze “absolutely amazing”

- Base score = **2.8**
- "Absolutely" is an **intensifier** → boost by ~25% (VADER heuristic)

```
2.8 × 1.25 = 3.5
```

Step 2: Analyze “a bit disappointing”

- Base score = **-2.1** (typical VADER value for "disappointing")
- "A bit" is a **downtoner** → reduce intensity by ~50%

```
-2.1 × 0.5 = -1.05
```

Step 3: Contrastive shift using “dri

In VADER:

> When there is a **"but"** in the sentence, emphasis is shifted toward the *clause after* "but".

So:

- The **positive impact** of "amazing" is **down-weighted**
- The **negative impact** of "disappointing" is **up-weighted**

Let's assume:

- Pre-“but” (positive part): weight = 0.5
- Post-“but” (negative part): weight = 1.5

> Adjusted positive = 3.5 × 0.5 = **1.75**
> Adjusted negative = -1.05 × 1.5 = **-1.575**

Step 4: Final Compound Score

> Total = 1.75 + (-1.575) = **+0.175**

This is a **slightly positive sentiment**, which aligns well with how humans interpret this sentence:

> “Overall good, but with a weak ending.”