Text Feature Extraction: N-grams & POS

🔗 N-grams

Contiguous sequences of N words

Examples:

```
Bigrams (2): "machine learning"
Trigrams (3): "natural language
processing"
```

- ✓ Capture local word order
- ✓ Identify common phrases
- ✓ Provide more context than single words

🏷️ POS Tagging

Identifies grammatical roles of words

Tags:

```
Noun, Verb, Adjective, Adverb...
```

- ✓ Identify syntactic patterns
- ✓ Analyze sentence structure
- ✓ Distinguish word usage contexts

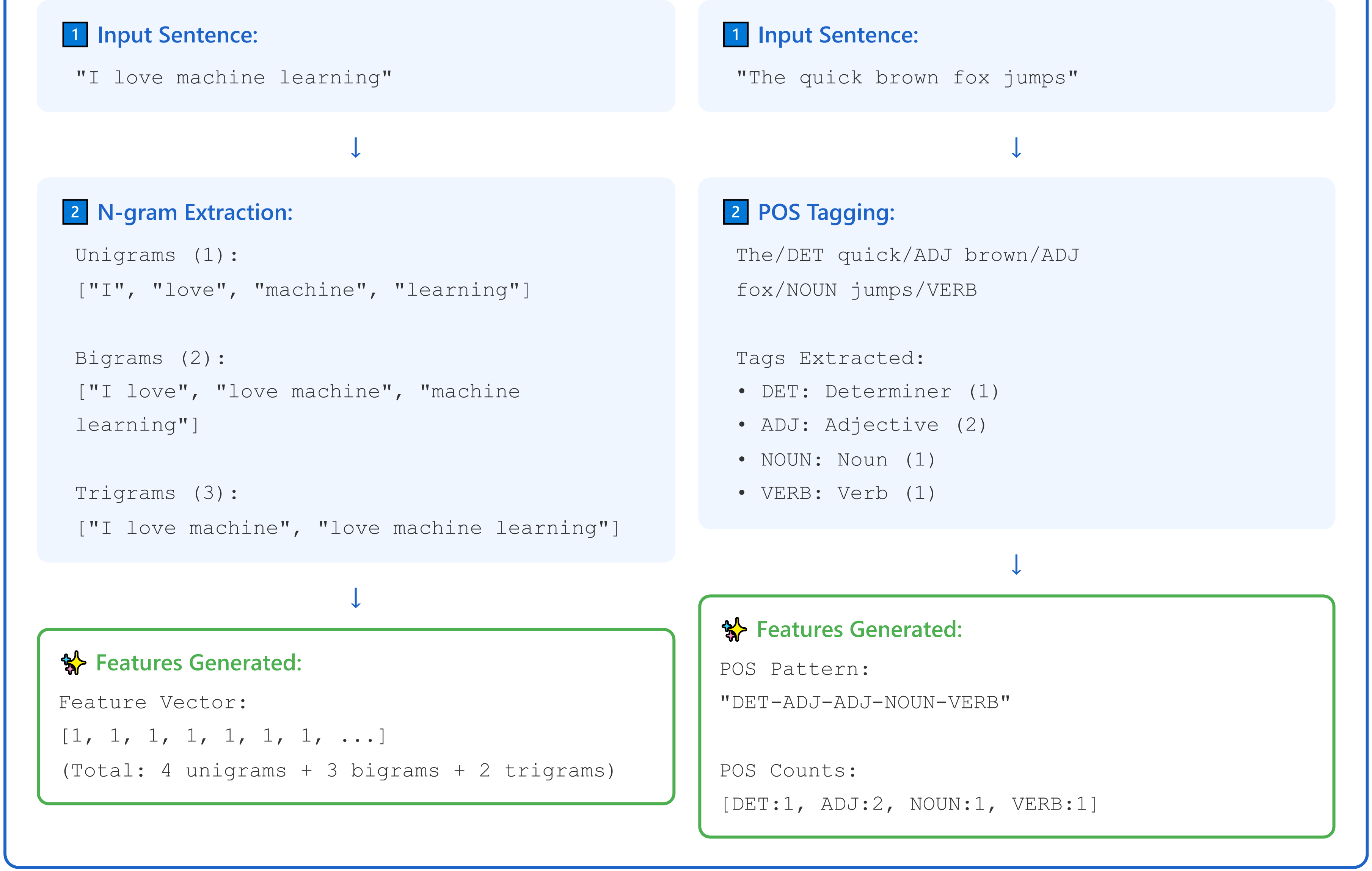
🤝 Combined Benefits

Using N-grams and POS tags together improves text classification and information extraction

⚖️ Trade-off

Between context capture and computational complexity (N-grams increase feature space)

💡 Practical Processing Examples



1 **Input Sentence:**

```
"I love machine learning"
```

↓

2 **N-gram Extraction:**

```
Unigrams (1):
["I", "love", "machine", "learning"]

Bigrams (2):
["I love", "love machine", "machine
learning"]

Trigrams (3):
["I love machine", "love machine learning"]
```

↓

✨ **Features Generated:**

```
Feature Vector:
[1, 1, 1, 1, 1, 1, 1, ...]
(Total: 4 unigrams + 3 bigrams + 2 trigrams)
```

1 **Input Sentence:**

```
"The quick brown fox jumps"
```

↓

2 **POS Tagging:**

```
The/DET quick/ADJ brown/ADJ
fox/NOUN jumps/VERB

Tags Extracted:
• DET: Determiner (1)
• ADJ: Adjective (2)
• NOUN: Noun (1)
• VERB: Verb (1)
```

↓

✨ **Features Generated:**

```
POS Pattern:
"DET-ADJ-ADJ-NOUN-VERB"

POS Counts:
[DET:1, ADJ:2, NOUN:1, VERB:1]
```