FAST-NU Peshawar

Natural Language Processing

Sami Uddin Shinwari

samu.shinwari@gmail.com
MS Data Science

Part I
Lemmatization

1 Introduction

Lemmatization in Natural Language Processing (NLP) is the process of reducing a word to its base or dictionary form, known as the lemma. Unlike stemming, which simply removes suffixes and prefixes to achieve a root form, lemmatization considers the context and the morphological analysis of the word. This results in more accurate normalization of words, which is particularly useful for tasks like text analysis, information retrieval, and machine learning.

2 Key Concepts of Lemmatization

2.1 Base Form Reduction

For example, the word "better" is reduced to "good", and "running" is reduced to "run".

2.2 Part of Speech (POS) Tagging

Lemmatization requires the correct POS tag to accurately transform the word to its base form. For instance, "running" as a verb becomes "run", but "better" as an adjective becomes "good".

2.3 Algorithms and Tools

- **WordNet Lemmatizer**: One of the most common tools used for lemmatization in Python is the WordNet lexical database.
- **spaCy**: Another popular NLP library that includes robust lemmatization capabilities.

3 Example in Python using NLTK

```
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet

lemmatizer = WordNetLemmatizer()

def get_wordnet_pos(word):
    """Map POS tag to first character lemmatize() accepts"""
    tag = nltk.pos_tag([word])[0][1][0].upper()
```

```
    tag_dict = {"J": wordnet.ADJ,
                "N": wordnet.NOUN,
                "V": wordnet.VERB,
                "R": wordnet.ADV}
    return tag_dict.get(tag, wordnet.NOUN)

words = ["running", "ran", "runs", "runner", "better"]
lemmas = [lemmatizer.lemmatize(word, get_wordnet_pos(word)) for word in words]

print(lemmas)
# Output: ['run', 'run', 'run', 'runner', 'good']
```

4 Use Cases

- **Search Engines**: Improves the relevance of search results by understanding the context of the words.
- **Text Analysis**: Enhances the accuracy of sentiment analysis, topic modeling, and other text-related tasks.
- **Machine Learning**: Essential preprocessing step for text classification, clustering, and other NLP applications.

5 Considerations

- **Context Sensitivity**: Lemmatization is more context-sensitive than stemming, making it more computationally intensive.
- **POS Tagging Requirement**: Accurate lemmatization depends on the correct POS tagging of words.
- **Language Dependency**: Lemmatization rules and tools are often language-specific and require different resources for different languages.

6 Comparison to Stemming

- **Lemmatization**: Converts words to their meaningful base forms using context and morphological analysis.
- **Stemming**: Cuts off prefixes and suffixes to get a root form, which may not be a real word.

7 Example in Python using spaCy

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("running ran runs runner better")

lemmas = [token.lemma_ for token in doc]

print(lemmas)
# Output: ['run', 'run', 'run', 'runner', 'good']
```

8 Summary

Lemmatization is a crucial step in NLP for achieving more accurate and contextually appropriate word normalization, thereby improving the performance and reliability of various text processing applications.