FastText Word Embeddings

Overview

FastText is an extension of the Word2Vec model developed by Facebook's AI Research lab. Unlike traditional word embedding models that treat words as atomic units, FastText represents each word as a bag of character n-grams. This approach allows FastText to capture subword information and morphology, enabling it to generate embeddings for words, including those not seen during training (out-of-vocabulary words). ?cite?turn0search0?

Why Use FastText?

- **Handling Out-of-Vocabulary Words**: By incorporating character n-grams, FastText can generate embeddings for words that were not present in the training data, addressing the out-of-vocabulary problem common in NLP tasks.
- **Capturing Morphological Features**: FastText's consideration of subword information allows it to capture the internal structure of words, making it effective for morphologically rich languages.
- **Improved Performance on Small Datasets**: FastText often outperforms other embedding models on smaller datasets due to its ability to utilize subword information. ?cite?turn0search7?

Prerequisites

- **Python**: Ensure Python is installed on your system.
- **Gensim Library**: Install the Gensim library, which provides an efficient implementation of FastText.

```
pip install gensim
```

Files Included

- **fasttext_example.py**: Python script demonstrating the training of a FastText model using Gensim.

Code Description

The provided code demonstrates how to train a FastText model using the Gensim library.

1. **Import Necessary Libraries**:

```
from gensim.models import FastText
```

   This imports the FastText class from the Gensim library.

2. **Prepare Training Data**:

```
sentences = [
    "This is the first sentence.",
    "This is the second sentence.",
    "A third sentence for training.",
    "And a fourth sentence.",
    "More sentences for the model to learn from.",
    "Sentence six.",
    "Sentence seven.",
    "Sentence eight.",
    "Sentence nine.",
    "Tenth and final sentence."
]
```

   A list of sentences is defined to serve as the training corpus.

3. **Train the FastText Model**:

```
fasttext_model = FastText(sentences, vector_size=50, window=3, min_count=1, worker
fasttext_model.train(sentences, total_examples=len(sentences), epochs=10)
```

   - `vector_size=50` : Sets the dimensionality of the word vectors to 50.
   - `window=3` : Specifies the maximum distance between the current and predicted word within a sentence.
   - `min_count=1` : Ignores all words with a total frequency lower than this.
   - `workers=4` : Uses four worker threads to train the model.

4. **Obtain Word Vector**:

```
word_vector = fasttext_model.wv['six']
print(word_vector)
```

   Retrieves and prints the vector representation of the word 'six'.

Expected Outputs

After running the script, the output will be the 50-dimensional vector representation of the word 'six'. The exact values will vary due to the random initialization of the model.

Use Cases

- **Text Classification**: Improving the performance of classifiers by providing rich word representations.
- **Machine Translation**: Enhancing translation models by capturing subword information.

- **Named Entity Recognition (NER)**: Improving the recognition of entities by understanding morphological variations.

Advantages

- **Efficient Training**: FastText is computationally efficient and can be trained quickly on large corpora.
- **Robustness to Rare Words**: By considering subword information, FastText provides meaningful embeddings for rare or misspelled words.

Future Enhancements

- **Hyperparameter Tuning**: Experiment with different hyperparameters like `vector_size`, `window`, and `min_count` to improve model performance.
- **Subword Information**: Adjust the `min_n` and `max_n` parameters to capture different lengths of character n-grams.
- **Pre-trained Models**: Leverage pre-trained FastText models for better performance on specific tasks.

References

- Gensim FastText Documentation
- Word Embeddings Using FastText - GeeksforGeeks
- Introduction to FastText Embeddings and its Implication
