Лабораторна робота 11 ІАД

Вступ до Natural Language Processing (NLP)

Мета: Познайомитися з основними поняттями, методами та підходами у сфері обробки природної мови (NLP). Провести порівняльний аналіз популярних алгоритмів та інструментів, а також підготувати презентацію на цю тему.

1. Основні етапи NLP

1.1 Токенізація

```
# Приклад токенізації
text = "Natural Language Processing (NLP) is a fascinating field of AI!"
tokens = word_tokenize(text)
print("Токени:", tokens)
```

1.2 Лемматизація та стемінг

```
# Стемінг
stemmer = PorterStemmer()
stemmed_tokens = [stemmer.stem(token) for token in tokens]
print("Стемінг:", stemmed_tokens)

# Лемматизація
lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [lemmatizer.lemmatize(token.lower()) for token in tokens]
print("Лемматизація:", lemmatized_tokens)
```

1.3 Векторизація тексту

1.3.1 Bag of Words

```
# Bag of Words
corpus = ["I love programming.", "Programming is fun!", "I love natural language processing."]
vectorizer = CountVectorizer()
X_bow = vectorizer.fit_transform(corpus)
print("Bag of Words:\n", X_bow.toarray())
```

1.4 Класифікація тексту

```python
# Демонстрація простої класифікації тексту на основі Bag of Words
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB

# Дані
texts = ["I love programming.", "I hate bugs.", "Debugging is fun.", "I dislike errors."]
labels = [1, 0, 1, 0]  # 1 - позитивний, 0 - негативний

# Векторизація
X = vectorizer.fit_transform(texts)
X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.2, random_state=42)

# Модель
model = MultinomialNB()
model.fit(X_train, y_train)
accuracy = model.score(X_test, y_test)
print("Точність класифікації:", accuracy)
```

1.5 Розпізнавання сутностей (NER)

```python
# NER за допомогою spaCy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Barack Obama was the 44th President of the United States.")
for ent in doc.ents:
    print(f"{ent.text}: {ent.label_}")
```

2. Порівняльний аналіз методів векторизації тексту

Метод	**Переваги**	Недоліки	**Застосування**	Складність реалізації
Bag of Words	Простота, швидкість	Ігнорує порядок слів, розмірність зростає	Простий аналіз тексту	Низька
TF-IDF	Враховує важливість слів у документі	Ігнорує семантику	Аналіз документів, пошукові системи	Середня
Word Embeddings	Враховує семантичну близькість слів	Вимагає багато даних для тренування	Машинний переклад, чат-боти	Висока

3. Огляд інструментів для NLP

Інструмент		Підтримка мов	Простота використання	Особливості

	Основні функції			
NLTK	Токенізація, стемінг, лемматизація	Багато	Середня	Широкий функціонал, але не завжди оптимальний
SpaCy	NER, токенізація, лемматизація	Англійська та ін.	Висока	Оптимізований для продуктивного використання
Hugging Face	Трансформери, GPT, BERT	Багато	Середня	Сучасні попередньо навчені моделі
Gensim	Word Embeddings, TF-IDF	Англійська	Висока	Сильна підтримка векторизації тексту

4. Застосування NLP

1. **Аналіз тональності:** Виявлення позитивних чи негативних відгуків.
2. **Чат-боти:** Автоматизація взаємодії з клієнтами.
3. **Рекомендаційні системи:** Персоналізація контенту на основі тексту.

Висновок

У ході виконання роботи було проведено дослідження основних етапів обробки природної мови (NLP), що включає токенізацію, лемматизацію, стемінг, векторизацію тексту та класифікацію. Також були розглянуті популярні інструменти та бібліотеки для NLP, такі як NLTK, SpaCy, Gensim та Hugging Face Transformers, із зазначенням їх основних переваг, недоліків і сфер застосування.

Основні результати:

1. **Токенізація:** Виділення окремих слів або фраз із тексту є базовим етапом NLP, який забезпечує основу для подальших обчислень.

2. **Лемматизація та стемінг:** Лемматизація дозволяє звести слово до його початкової форми, враховуючи контекст, тоді як стемінг – більш простий метод, що відкидає закінчення слів.
3. **Векторизація тексту:**
    a. **Bag of Words (BoW):** Проста та ефективна техніка для задач класифікації тексту.
    b. **TF-IDF:** Враховує частоту появи слів у документі та їх унікальність, що робить його більш точним для аналізу документів.
    c. **Word Embeddings (Word2Vec):** Успішно представляє семантику слів і виявляє схожість між ними.
4. **Класифікація тексту:** Застосування наївного баєсового класифікатора дозволило побудувати просту модель для визначення тональності тексту.
5. **Розпізнавання сутностей (NER):** За допомогою бібліотеки SpaCy успішно визначено іменовані сутності в тексті (імена, локації тощо).

Порівняння методів:

- Порівняльний аналіз методів векторизації тексту показав, що кожен підхід має свої переваги та недоліки, і вибір методу залежить від конкретної задачі.
- Сучасні моделі на основі Word Embeddings (Word2Vec, GloVe) та трансформери є найбільш ефективними для складних задач NLP.

Застосування:

NLP має широкий спектр застосувань, включаючи аналіз тональності, створення чат-ботів, автоматизацію перекладів, пошук інформації та рекомендаційні системи.