

</ Similar questions
predictor on
StackOverflow />

} /> [

Houda Ait Abdeslam

Description:

In this project, my task is to develop a tool designed to predict similar questions on StackOverflow, a popular platform for programmers to ask and answer technical questions.



Dataset:

Dataset: The dataset is sourced from a similar project on GitHub and contains StackOverflow questions. Each row of the data set contains a pair of questions, their tags, title and id.

OId	OTitle	OTags	OBody	DId	DTitle	DTags	DBody
4	How to convert a Decimal to a Double in C#?	<c#><floating-point><type-conversion><double><...>	<p>I want to use a <code>TrackBar</code> to c...	51027658	How to echo a JS variable to php?	<javascript><php>	<p>Is that possible to pass a JS variable to P...
9	How do I calculate someone's age based on a Da...	<c#><.net><datetime>	<p>Given a <code>DateTime</code> representing ...	2194999	How to calculate an age based on a birthday	<c#><asp.net-mvc><date-arithmetic>	<blockquote>\n<p>Possible Duplicate:</...>
11	Calculate relative time in C#	<c#><datetime><time><datediff><relative-time-...>	<p>Given a specific <code>DateTime</code> valu...	7392566	How to get datetime in words like "today",	<php><zend-framework><datetime><datetime-format>	<blockquote>\n<p>Possible Duplicate:...

1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 0 1

Preprocessing:

Steps:

- 1- Used BeautifulSoup to strip HTML tags.
- 2- Used NLTK's word_tokenize to split text into words or tokens.
- 3- Eliminated common stop words (e.g., "and", "the", "is").
- 4- Used NLTK's WordNetLemmatizer to convert words to base forms.
- 5- Used WordNet Library to replace words with synonyms.

```
# Function to preprocess text
def preprocess_text(text):
    # Remove HTML tags
    text = BeautifulSoup(text, 'html.parser').get_text()
    # Normalize text (handle contractions, remove special characters)
    text = re.sub(r'\b(can\t|cannot)\b', 'can not', text)
    text = re.sub(r'\b(don\t)\b', 'do not', text)
    text = re.sub(r'^a-zA-Z0-9\s', '', text)
    # Tokenize text
    tokens = word_tokenize(text.lower())
    # Remove stop words and non-alphanumeric tokens
    tokens = [lemmatizer.lemmatize(token) for token in tokens
              if token.isalnum() and token not in stop_words]
    # Replace words with synonyms
    tokens = synonym_replacement(tokens)
    return tokens
```

word2Vec model training:

Corpus:

```
# Combine all preprocessed titles, bodies
corpus = (
    df['OTitle_preprocessed'].tolist() +
    df['DTitle_preprocessed'].tolist() +
    df['OBody_preprocessed'].tolist() +
    df['DBody_preprocessed'].tolist() )
```

Model:

```
# Train Word2Vec model
word2vec_model = Word2Vec(sentences=corpus, vector_size=100, window=5, min_count=1, workers=4, epochs=50)
```

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Similarity function:

In the Word2Vec function, we generate token lists for each question, compute the average word vector by taking the mean of word vectors for all tokens, and use cosine similarity to measure the similarity between the mean vectors.

```
77 # Function to calculate similarity using Word2Vec
78 def word2vec_similarity(row, model):
79     # Combine tokens from titles and bodies for original and duplicate questions
80     otitle_tokens = row['OTitle_preprocessed'] + row['OBody_preprocessed']
81     dtitle_tokens = row['DTitle_preprocessed'] + row['DBody_preprocessed']
82
83     # If no tokens found for either question, return similarity score of 0
84     if not otitle_tokens or not dtitle_tokens:
85         return 0.0
86
87     # Calculate average word vectors for original and duplicate questions
88     otitle_vec = np.mean([model.wv[token] for token in otitle_tokens], axis=0)
89     dtitle_vec = np.mean([model.wv[token] for token in dtitle_tokens], axis=0)
90
91     # If any of the vectors have zero norm, return similarity score of 0
92     if np.linalg.norm(otitle_vec) == 0 or np.linalg.norm(dtitle_vec) == 0:
93         return 0.0
94
95     # Calculate cosine similarity between the mean vectors
96     cosine_sim = cosine_similarity([otitle_vec], [dtitle_vec])[0][0]
97     return round(cosine_sim * 100, 2)
```

Result:

Question 1:

Question 2:

Similarity %:

How do I calculate someone's age based on a DateTime type birthday?	How to calculate an age based on a birthday	88.64
Calculate relative time in C#	How to get datetime in words like "today", "yesterday" or "25 minutes ago"?	91.32
Determine a user's timezone	get user timezone	88.23
Difference between Math.Floor() and Math.Truncate()	What is the difference in floor function and truncate function?	72.44
Binary Data in MySQL	Save OpenSSL encrypted strings in MySQL database	75.83
Throw an error preventing a table update in a MySQL trigger	Check constraint in MySQL	86.07
Best way to allow plugins for a PHP application	Plugin architecture in PHP	95.14
Multiple submit buttons in an HTML form	One form, multiple submit buttons and submit links	94.01

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Result discussion and improvement:

The approach
successfully identifies
similar questions based
on their semantic
meaning.

- A bigger dataset
- Refine the preprocessing step

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1