Power of Topic Modeling: Crafting a Compelling LinkedIn "About" Section

Overview:
In the fast-paced world of professional networking, where first impressions matter, our LinkedIn profile is our digital business card. Among its various sections, the "About" information is a crucial space where we can showcase our professional identity and leave a lasting impression on potential connections. One powerful technique to enhance the effectiveness of our LinkedIn "About" section is topic modeling, a method in natural language processing that uncovers hidden thematic structures within a text. By employing algorithms to analyze content and identify recurring themes or topics, topic modeling not only organizes information but also boosts the discoverability of key skills, experiences, and interests.

Importance of Topic Modeling on LinkedIn:
Showcasing Expertise:
Utilizing topic modeling in "About" section enables to highlight expertise in a structured manner. By identifying and emphasizing key themes related to our professional background, we can provide a clearer picture of our skills and competencies.

Enhancing Searchability:
With millions of users on LinkedIn, making their profile easily discoverable is crucial. Implementing topic modeling ensures that our profile aligns with relevant industry keywords, increasing the likelihood of appearing in search results.

Tailoring Content for Our Audience:
Different connections may be interested in various aspects of our professional journey. Topic modeling allows us to customize our "About" section based on the preferences of our target audience, making our profile more engaging and relevant to those who visit it.

Optimizing First Impressions:
First impressions matter, and our "About" section is often the first thing people read when they visit our profile. Topic modeling helps us craft a concise and impactful introduction, providing a quick overview of our skills and experiences.

Various Methods of Topic Modeling:
1. Latent Dirichlet Allocation (LDA)
2. Non-Negative Matrix Factorization (NMF)
3. BERTopic
4. Correlated Topic Model (CTM)
5. Word Embeddings (Word2Vec, GloVe, FastText)

For our LinkedIn data, we employed the **Latent Dirichlet Allocation (LDA)** model.

Implementation of Topic Modeling:

1. Imported all the required libraries and packages.
2. Imported cleaned LinkedIn CSV data file containing attributes such as follower count, number of connections, and the "about_section_linkedin" information**.**

```
df = pd.read_csv("/Users/apurva/Downloads/linkedin-instructor-cleaned.csv", encoding='ISO-8859-
df
```

	index_number	followers_count_linkedin	number_of_connections_mapped_linkedin	about_section_linkedin
0	2422	4517	1	Professor of the School of Engineering Technol...
1	757	3070	1	I have been working with JavaScript from the b...
2	510	1100	1	Try https://robotrecipes.co and https://produc...
3	337	2294	1	Accomplished Technology Leader offering over t...
4	1741	4153	1	I am a staff software engineer (Tech Lead) wit...

3. Removal of stop words and stemming:

- Stop words removal involves eliminating common, non-substantive words from text data to enhance computational efficiency in topic modeling.
- Stemming reduces words to their root form, aiding in grouping similar terms but potentially sacrificing semantic precision.

```
[['professor', 'school', 'engin', 'technolog', 'appli', 'scienceprofessor', 'school', 'engin', 'technolog', 'appli', 'scienc'],
['work', 'javascript', 'begin', 'web', 'develop', 'free', 'time', 'like', 'work', 'golang', 'rust', 'flutter', 'also', 'like',
'build', 'like', 'learn', 'henc', 'even', 'share', 'learn', 'internet', 'via', 'articl', 'video', 'teach', 'eventu', 'learn', '
http', 'work', 'javascript', 'begin', 'web', 'develop', 'career', 'free', 'time', 'like', 'work', 'golang', 'rust', 'flutter',
'also', 'like', 'build', 'game', 'like', 'learn', 'henc', 'even', 'share', 'learn', 'internet', 'via', 'articl', 'video', 'teac
h', 'eventu', 'learn', 'http'], ['tri', 'http', 'http', 'http', 'http'], ['accomplish', 'technolog', 'leader', 'offer', 'ten',
'year', 'demonstr', 'career', 'successdevelop', 'execut', 'oper', 'strategi', 'promot', 'organiz', 'growth', 'optimalutil', 'te
chnolog', 'experi', 'lead', 'oper', 'technolog', 'businessdevelop', 'applic', 'develop', 'full', 'p', 'l', 'respons', 'recogn',
```

4. Train the Latent Dirichlet Allocation (LDA) model

- LDA is a generative probabilistic model widely used in topic modeling. It assumes that documents are mixtures of topics, and topics are mixtures of words.
- LDA helps unveil underlying thematic structures and extract meaningful insights from large datasets.

a) Function to calculate coherence score:

This function evaluates the optimal number of topics in a topic modeling task by comparing coherence scores for different topic numbers.

```python
# Function to calculate coherence score
def calculate_coherence(dictionary, corpus, texts, limit, start=2, step=3):
    coherence_values = []
    model_list = []
    for num_topics in range(start, limit, step):
        model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary)
        model_list.append(model)
        coherencemodel = CoherenceModel(model=model, texts=texts, dictionary=dictionary, cohere
        coherence_values.append(coherencemodel.get_coherence())

    return model_list, coherence_values
```

b) Function to find the optimal number of topics

This function utilizes the previously defined calculate_coherence function to determine the optimal number of topics for a given topic modeling task.

```python
# Function to find the optimal number of topics
def find_optimal_topics(dictionary, corpus, texts, limit, start=2, step=3):
    model_list, coherence_values = calculate_coherence(dictionary, corpus, texts, limit, start,

    # Find the number of topics with the highest coherence score
    optimal_num_topics = start + (coherence_values.index(max(coherence_values)) * step)

    return optimal_num_topics
```

c) Create a function to apply topic modeling with stopword removal and stemming

This function processes a given text for topic modeling by removing repetitions, stopwords, and applying stemming.

```python
def apply_topic_modeling(text):
    # Check if the input is a non-null string
    if isinstance(text, str):
        # Remove repetitions of sentences
        sentences = re.split(r'[.!?]', text)
        unique_sentences = list(set(sentences))
        cleaned_text = ' '.join(unique_sentences)

        # Split the cleaned text into words
        words = re.findall(r'\b\w+\b', cleaned_text)

        # Remove stopwords
        stop_words = set(stopwords.words('english'))
        words = [word for word in words if word.lower() not in stop_words]

        # Apply stemming
        stemmer = PorterStemmer()
        words = [stemmer.stem(word) for word in words]

        # Remove specific common words
        words = [word for word in words if word.lower() not in ['of', 'the', 'also']]

        # Create a Dictionary from the processed words
        dictionary = Dictionary([words])

        # Create a corpus from the processed words
        corpus = [dictionary.doc2bow([word]) for word in words]

        # Find the optimal number of topics
        optimal_num_topics = find_optimal_topics(dictionary, corpus, [words], limit=10, start=2, step=

        # Train the LDA model with the optimal number of topics
        lda_model = LdaModel(corpus, num_topics=optimal_num_topics, id2word=dictionary, passes=10)

        # Return the topics and associated keywords
        return lda_model.print_topics(num_topics=optimal_num_topics, num_words=5)
```

d) Apply the function to each entry in 'about_section_linkedin'

This code applies the apply_topic_modeling function to each entry in the 'about_section_linkedin' column of a DataFrame, storing the resulting topics and keywords in a new column.

```
# Apply the function to each entry in 'about_section_linkedin'
df['lda_topics'] = df['about_section_linkedin'].apply(apply_topic_modeling)

# Display the updated DataFrame
print(df)
```

5. Selection of optimal number of topics using coherence scores

- The selection of the number of topics and words for topic modeling depends on the nature of our data, our research objectives, and the interpretability of the results.
- Coherence score is used to evaluate the quality of topics for different numbers.

Output:

Example 1 :

Text:

"Experienced Software Engineer with a demonstrated history of working in the information technology, services industry, data science and machine learning fields. Skilled in Python, Java, Scala, Oracle, Hadoop, IBM DB2. Strong software engineering professional with a Masters in Computer Science"

After executing the code for topic modelling, we got optimal number of topics as 6.

Result:

[(0, '0.220*"engin" + 0.119*"histori" + 0.119*"oracl" + 0.119*"machin" + 0.017*"softwar"'), (1, '0.132*"profession" + 0.132*"python" + 0.132*"scala" + 0.132*"servic" + 0.019*"softwar"'), (2, '0.146*"softwar" + 0.079*"skill" + 0.079*"learn" + 0.079*"work" + 0.079*"science√¢"'), (3, '0.108*"java" + 0.108*"hadoop" + 0.108*"demonstr" + 0.108*"data" + 0.108*"technolog"'), (4, '0.149*"scienc" + 0.149*"experienc" + 0.149*"comput" + 0.021*"softwar" + 0.021*"engin"'), (5, '0.149*"inform" + 0.149*"focus" + 0.149*"db2" + 0.021*"engin" + 0.021*"softwar"')]

Interpretation of output:

Number of topics	Keywords	Interpretation
Topic 0	engineering, history, Oracle, machine, software	This topic seems to focus on software engineering with an emphasis on historical aspects and Oracle technologies.

Topic 1	profession, Python, Scala, service, software	This topic appears to revolve around professional services and proficiency in programming languages like Python and Scala.
Topic 2	software, skill, learn, work, science	This topic emphasizes software skills and continuous learning, possibly related to the field of data science.
Topic 3	Java, Hadoop, demonstrate, data, technology	This topic suggests a focus on Java, Hadoop, and technologies related to data processing.
Topic 4	science, experience, computer, software, engineering	This topic centers around computer science and practical experience in the field.
Topic 5	information, focus, DB2, engineering, software	This topic seems to concentrate on information-related aspects, with a specific mention of DB2 and software engineering.

Each keyword in the output is associated with a weight, indicating its significance within the respective topic. For example, in Topic 3, "Java" has a higher weight compared to other words, suggesting that it is a more dominant keyword in that topic. The interpretation is somewhat subjective and may require domain-specific knowledge to fully understand the context of the topics.

Example 2:

Text:
"I am passionate about studying and teaching"

Result:
[(0, '0.567*"studi" + 0.221*"passion" + 0.212*"teach"'), (1, '0.426*"teach" + 0.419*"passion" + 0.155*"studi"')]

Interpretation of output:

Number of topics	**Keywords**	**Interpretation**
Topic 0	study, passion, teach	This topic suggests a focus on studying, with a significant emphasis on passion and teaching.
Topic 1	teach, passion, study	This topic appears to revolve around teaching and passion, with a mention of studying as well.

Conclusion:

In the competitive world of professional networking, leveraging the power of topic modeling in our LinkedIn "About" section is a strategic move that can set us apart. By utilizing this technique, we can create a profile that not only reflects our expertise but also resonates with our target audience, making meaningful connections more likely in the ever-evolving professional landscape. Craft our narrative, showcase our skills, and unveil the power of topic modeling for a compelling LinkedIn presence.

Note: Python code used for building above Topic model is available on below GitHub link:
https://github.com/apurvavaze22/Topic-Modeling