**Implementation Scheme of an Intelligent NLP-Based Knowledge Extraction and Recommender System**

The Intelligent Natural Language Processing (NLP) based Knowledge Extraction and Recommender System aims to revolutionize customer service experience in an e-commerce platform. With the help of AI and machine learning technologies, the system will provide real-time responses to customers' queries, offer personalized product recommendations and enhance overall user experience. It is designed with a focus on user-friendliness, efficiency, and functionality.

The system will be developed using AWS cloud services, specifically the services under Amazon Machine Learning for Natural Language Processing (NLP) tasks, and will be implemented using Python, a popular programming language for AI applications.

System Modules

1. Front-End User Interface: The User Interface (UI) will be the primary interaction point for customers. It will be designed using React.js due to its efficiency and flexibility. AWS Amplify can be used to deploy and host the front-end and integrate with AWS Cognito for secure user registration and login.

2. Query Understanding and Answer Generation Module: This module, powered by Amazon Comprehend, will handle NLP tasks like understanding user queries and generating appropriate responses. It will also use Amazon Elasticsearch for quick keyword searches.

3. Recommendation System Module: This module will leverage Amazon Personalize, which uses machine learning to create high-quality recommendations for users. Based on the user's history and preferences, the system will suggest products they might be interested in.

4. Speech Input Module (Recommended): This module will be implemented using AWS Transcribe, enabling voice queries from users.

5. i18n Multi-Language Interface Support (Recommended): AWS Translate can be integrated to support multiple languages.

6. AI Voice Packet Reply (Optional): This can be achieved by integrating Amazon Polly to convert text into lifelike speech.

7. Sentiment Analysis Module (Optional): Using Amazon Comprehend, this module can analyze customer reviews and feedback to gauge sentiment and improve the recommendation system.

Data Circulation

In the Intelligent NLP-Based Knowledge Extraction and Recommender System, data will play a significant role. The management of this data, from its generation to storage, will be executed efficiently to ensure smooth system performance. In our proposed design, Amazon DynamoDB, a fast and flexible NoSQL database service, is chosen as the data storage solution due to its scalability and reliability.

When a user interacts with the system—be it asking a question, making a purchase, or browsing products—each action generates data. This data will be used by the system to understand user behavior, formulate responses, and generate personalized product recommendations.

For instance, when a user submits a query, the system records it and sends it to the Query Understanding and Answer Generation Module. Here, Amazon Comprehend, an NLP service, processes and understands the query, generating an appropriate response. This response data is then stored back in DynamoDB. The same process applies to the Recommendation System Module; the system records user interactions and, using Amazon Personalize, processes this data to provide personalized recommendations.

The Sentiment Analysis Module (optional) will also analyze user-generated content such as product reviews and feedback. Using Amazon Comprehend's sentiment analysis capabilities, it can evaluate whether the sentiment expressed in the text is positive, neutral, negative, or mixed. The results of this analysis can then be used to refine the product recommendations and overall customer service.

To ensure the swift transfer of data between the front-end user interface and the backend modules, we will use AWS Lambda for serverless backend operations. AWS Lambda will be in charge of receiving requests from the front-end, processing these requests using the appropriate services, and returning responses.

By implementing such a robust data circulation scheme, we ensure that every piece of information is effectively used to enhance user experience and system efficiency.

User Interaction Scheme

The interaction between the user and the system begins at the front-end user interface, designed with user-friendliness in mind. When a user logs in (authentication facilitated by AWS Cognito), they are greeted by a personalized dashboard showing product recommendations generated by the Recommendation System Module.

When a user submits a query, either through text input or voice input (for the latter, AWS Transcribe is used), the system forwards this query to the Query Understanding and Answer Generation Module. Leveraging Amazon Comprehend, this module processes the query and formulates an appropriate response. This response is then displayed to the user in a user-friendly format on the dashboard.

If the user browses products or makes a purchase, the Recommendation System Module takes note of this interaction. Using Amazon Personalize, it processes the user's history and preferences to update the product recommendations on the user's dashboard.

Moreover, the system provides multi-language support (AWS Translate) and AI voice packet replies (Amazon Polly), making it more accessible and interactive.

The design of the user interaction scheme prioritizes usability and efficiency, ensuring that users can effectively navigate and utilize the system to its full potential.

System Integration Scheme

Integrating the various system components effectively is vital to ensure smooth operation. In this system, AWS Lambda serves as the backbone connecting the different modules. Amazon API Gateway manages the data flow between these services, ensuring a well-integrated, robust, and scalable system.

The front-end user interface, developed using React.js, communicates with the backend services (Query Understanding and Answer Generation, Recommendation System, etc.) via APIs managed by API Gateway. These APIs trigger AWS Lambda functions which execute serverless operations involving various AWS services like Amazon Comprehend, Amazon Personalize, AWS Transcribe, AWS Translate, and Amazon Polly.

Data generated from these services is stored in Amazon DynamoDB. When needed, AWS Lambda retrieves this data from DynamoDB and sends it back to the front-end through API Gateway.

The system's scalability is maintained with AWS's auto-scaling feature, ensuring the system can handle fluctuating workloads by automatically adjusting capacity. In essence, the integration scheme guarantees a seamless user experience by facilitating efficient communication between the system's various components.

Five-Week Schedule

Our goal is to deliver a fully functional, efficient, and user-friendly Intelligent NLP-Based Knowledge Extraction and Recommender System in five weeks. Here is a detailed schedule for the project.

Week 1: Design and Develop Front-End User Interface. We will begin by designing an intuitive and user-friendly interface using React.js. Once the design is approved, we will develop the interface and integrate AWS Cognito for secure user registration and login functionality.

Week 2: Implement Query Understanding and Answer Generation Module. This week will be devoted to integrating Amazon Comprehend for processing and understanding user queries. We will set up AWS Lambda functions to handle requests and responses and use Amazon Elasticsearch for swift keyword searches.

Week 3: Implement Recommendation System Module. Leveraging Amazon Personalize, we will develop a recommendation system that offers users high-quality, personalized product suggestions based on their interaction history and preferences.

Week 4: Implement Speech Input, i18n Multi-Language Interface Support, and AI Voice Packet Reply. We will enhance system accessibility by integrating AWS Transcribe for voice queries and AWS Translate for multi-language support. Additionally, we'll incorporate Amazon Polly to enable AI voice packet replies.

Week 5: Implement Sentiment Analysis Module and Final Testing. The last week will involve integrating Amazon Comprehend for sentiment analysis of user reviews and feedback. Following this, we will perform extensive testing and debugging of the entire system to ensure seamless performance.

By adhering to this schedule, we can systematically tackle each aspect of the system, ensuring that every component is designed, developed, and integrated efficiently within the proposed timeframe.

The development process will be a five-week venture focusing on system design, implementation, testing, and fine-tuning of each module. This structured approach will ensure the completion of the project on time and with the desired outcome. The potential of this system in boosting sales and improving customer experience makes it a worthwhile investment for the e-commerce platform.

Literature Reading List:

1. Manning, C. D., & Schütze, H. (1999). "Foundations of Statistical Natural Language Processing". MIT Press.

2. Shalev-Shwartz, S., & Ben-David, S. (2014). "Understanding Machine Learning: From Theory to Algorithms". Cambridge University Press.

3. Bishop, C. M. (2006). "Pattern Recognition and Machine Learning". Springer.

4. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). Mining of massive data sets. Cambridge university press.

5. Parsing, C. (2009). Speech and language processing. Power Point Slides.

6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26.

7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

8. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.

9. Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. ACM computing surveys (CSUR), 52(1), 1-38.

10. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

Reading Plan:

Week 1: Read "Foundations of Statistical Natural Language Processing" and "Speech and Language Processing" for a foundation in natural language processing, which will be crucial for understanding user queries.

Week 2: Read "Understanding Machine Learning: From Theory to Algorithms", "Pattern Recognition and Machine Learning", and "Mining of Massive Datasets". These will provide a solid understanding of machine learning and its applications, particularly for the recommendation system.

Week 3: Read "Distributed representations of words and phrases and their compositionality", "Attention is all you need", and "Sequence to sequence learning with neural networks". These papers discuss advanced NLP techniques like word embeddings and sequence-to-sequence models, which are fundamental for the query understanding and answer generation module.

Week 4: Read "Recommender system with deep learning: A survey" and "Matrix Factorization Techniques for Recommender Systems". These will provide a comprehensive understanding of deep learning-based recommender systems and collaborative filtering, important for the recommendation system module.

Week 5: Review and summarize the learnings from each week and apply them to your project.

Correlation Between Literature:

The literature recommended for weeks 1 and 3 are related to NLP and are essential for creating a system that understands and generates human language effectively. The literature for week 2 provides a general background on machine learning, forming a foundation for understanding more specialized machine learning techniques used in the system (as discussed in the week 3 and 4 readings).

Correlation Between Literature and System Implementation:

The books and papers for weeks 1 and 3 correlate directly with the Query Understanding and Answer Generation Module, helping you understand how to process and generate human language using NLP techniques. The literature for week 2 will help you understand the core machine learning concepts needed for this project, while the papers for week 4 are directly applicable to the Recommendation System Module. By combining the knowledge gained from all the literature, you will have a robust understanding of how to build and optimize your system.