

Analyse des Approches IA pour le Projet

1 Tableau comparatif des approches IA possibles

Approche IA	Description	Avantages	Inconvénients
Réseaux de neurones (DL)	Utilisation de modèles comme RNN, LSTM, ou Transformers pour comprendre et générer du langage	Bonne compréhension du langage, personnalisation possible	Besoin de beaucoup de données et de puissance de calcul
Systèmes de recommandation	Basé sur le filtrage collaboratif, contenu ou modèles hybrides	Personnalisation efficace, bon ROI sur la fidélisation	Peut nécessiter un historique utilisateur important
Systèmes experts	Règles logiques définies à la main pour guider les interactions	Transparent, explicable, rapide à prototyper	Peu adaptatif, ne gère pas bien les conversations complexes
Systèmes NLP classiques	TF-IDF, Word2Vec, chatbot à base de règles ou similarité sémantique	Moins coûteux à implémenter, rapide	Limité pour des conversations riches et personnalisées
Large Language Models (LLMs)	Modèles pré-entraînés (GPT, BERT, LLaMA...) capables de comprendre et générer des conversations	Réponses naturelles, haute compréhension contextuelle	Complexité d'intégration, coût, possible hallucination
Systèmes multi-agents (SMA)	Plusieurs agents IA (ex. : dialogue, recommandation, analyse) coopèrent pour une tâche commune	Modulaire, distribué, adaptatif, facilement extensible	Complexité de coordination et de développement

Approche la plus adaptée : **Systèmes Multi-Agents (SMA)**

2 Approches NLP (Traitement du langage naturel)

Approche NLP	Description	Avantages	Inconvénients
TF-IDF / BM25	Pondération des mots selon leur fréquence	Simple, rapide, utile pour la recherche ou matching basique	Ignore le contexte et la sémantique
Word2Vec/GloVe	Représentation sémantique des mots	Bon pour la similarité sémantique	Ne prend pas bien en compte le contexte global
BERT / Distil-BERT	Encodage profond de phrase, compréhension contextuelle	Excellent pour la classification, l'analyse d'intention, le dialogue	Plus lent, nécessite du fine-tuning ou des embeddings adaptés
GPT (OpenAI, LLaMA, etc.)	Modèles génératifs pour dialogue, reformulation, personnalisation	Réponses naturelles et contextuelles, possible few-shot/fine-tuning	Coûteux, hallucinations possibles, besoin d'un encadrement
Sentence Transformers	Embeddings de phrases ou de documents avec BERT optimisé pour la similarité	Idéal pour les recherches sémantiques, matching utilisateur-produit	Nécessite un index ou vectordb pour l'efficacité

Approche NLP la plus efficace : Sentence Transformers ou GPT, selon la tâche

- Pour la **compréhension et génération de dialogue** → **GPT**
- Pour la **recommandation contextuelle via matching** → **Sentence Transformers**

3 Approches CNN (utiles dans certains modules du SMA)

Utilisation CNN	Description	Avantages	Inconvénients
Analyse d'images (profil)	Pour analyser des images de produit, photos clients, ou profil visuel	Extraction de caractéristiques visuelles utile pour la recommandation	Hors du cœur du chatbot, mais utile si multimodalité
CNN pour NLP (TextCNN)	Applique CNN sur des représentations textuelles (embeddings)	Rapide, bon pour la classification (ex : intention, sentiments)	Moins performant que les Transformers sur du dialogue complexe
Hybrid CNN-LSTM	Mélange CNN (local features) et LSTM (séquence)	Meilleure capture des patterns textuels qu'un CNN seul	Plus complexe, mais pas toujours supérieur à BERT

Approche CNN la plus efficace : **TextCNN** pour des modules spécialisés (ex. détection d'intention rapide)

4 Orchestrateurs d'agents (LangGraph, CrewAI, Rasa, etc.)

Orchestrateur / Framework	Description	Avantages	Inconvénients
CrewAI	Coordination explicite d'agents autonomes avec rôles et outils	Facile à intégrer avec LLMs, agents spécialisés, bonne modularité	Encore jeune, limité pour des workflows complexes
LangGraph (LangChain)	Graphes d'agents avec flux de conversation dynamique	Parfait pour workflows adaptatifs, bonne gestion du contexte	Complexe à maintenir, dépendance forte à LangChain
Rasa	Framework open source de chatbot orienté dialogue et NLP	Excellent pour dialogues structurés, intégration avec actions backend	Moins flexible pour l'IA générative ou multi-agent
Autogen/ Auto-genStudio	Outils Microsoft pour la coordination d'agents LLM	Fort potentiel, très modulaire, collaboration inter-agents poussée	Encore en évolution, nécessite une bonne configuration initiale
OpenAgents (OpenAI)	Coordination d'outils et agents via OpenAI	Simplicité d'usage, bon pour tâches ciblées	Fermé, peu personnalisable pour SMA avancé

Approche d'orchestration la plus efficace : **LangGraph** pour SMA flexible, adaptatif et contextuel

5 Systèmes de recommandation utilisables dans un SMA

Méthode de recommandation	Description	Avantages	Inconvénients
Filtrage collaboratif	Basé sur les préférences d'utilisateurs similaires	Très performant avec historique riche	Problème du démarrage à froid, biais

Méthode de recommandation	Description	Avantages	Inconvénients
Filtrage basé sur le contenu	Basé sur les caractéristiques des produits ou profils	Recommandation personnalisée possible	Moins adaptatif, peut stagner sans nouveauté
Modèles hybrides	Combine les deux approches ci-dessus	Très bonne performance globale	Plus complexe à mettre en œuvre
Recommandation par LLM	Utilise GPT ou BERT pour générer des recommandations explicites	Personnalisé et conversationnel	Peut halluciner, nécessite garde-fous

Approche recommandation la plus efficace : Modèle hybride + embeddings (ex. FAISS + SentenceTransformer)

6 Stockage / Base de connaissances / Mémoire des agents

Outil / Technologie	Description	Avantages	Inconvénients
FAISS / Chroma / Weaviate	Indexation vectorielle pour similarité sémantique	Parfait pour la recherche d'info contextuelle	Nécessite du pré-traitement
Redis(VectorDB)	Stockage clé-valeur ou vecteur rapide	Très rapide, bonne intégration	Moins adapté pour grands corpus
GraphDB/Neo4j	Base de données de graphes	Idéal pour modéliser des relations complexes	Moins optimisé pour NLP pur

Approche la plus efficace : Chroma ou FAISS avec Sentence Embeddings pour mémoire contextuelle