**SentenceTransformers - Utilisation et Optimisation**

**Optimisation des embeddings pour la recherche sémantique (40 J 4 x 1 semaine à 2 )**

1. Tester différentes méthodes d’encodage des textes.
2. Comparer les performances en recherche d’information.
3. Analyser l’impact des embeddings sur la qualité des résultats.
4. Ajuster les techniques de normalisation et d’indexation.

**Intégration des embeddings (20 J 2 semaines à 2)**

1. Définir le format d’encodage des données.
2. Optimiser l’indexation pour accélérer les requêtes.
3. Évaluer l’impact du stockage sur la performance globale.
4. Ajuster les paramètres pour minimiser l’espace mémoire utilisé.

**Tokenisation (40 J 4 x 1 semaine à 2)**

1. Préparer un corpus de texte pour entraîner ou tester la tokenisation.
2. Appliquer la tokenisation et analyser sa performance (ex : longueur moyenne des tokens).
3. Comparer différents types de tokenisation pour choisir la plus adaptée au modèle.

**Encodage des positions (30 J 3 semaines à 2)**

1. Sélectionner un type d’encodage de position (sinusoïdal, appris, rotatif).
2. Comprendre comment l’encodage affecte la compréhension du contexte par le modèle.
3. Visualiser les encodages pour voir leur répartition dans l’espace des embeddings.
4. Expérimenter différentes configurations pour améliorer les performances du modèle.

**Données et stockage**

**Nettoyage des données**

1. Identifier et supprimer les doublons ou erreurs dans les données.
2. Normaliser les textes (lowercase, suppression des caractères spéciaux).
3. Éliminer les biais et analyser la distribution des données.
4. Mettre en place un pipeline de validation continue.

**Création et mise en place de bases de données vectorielles (60 J 3 x 2 semaine à 2 )**

1. Choisir un moteur de base de données vectorielle performant.
2. Définir la structure des embeddings et la taille optimale.
3. Optimiser l’indexation pour accélérer la recherche.
4. Tester la robustesse et la scalabilité avec différents jeux de données.

**Optimisation, gestion et veille**

**Optimisation des transformers (80 J tache de fond pour environ 1 J par semaine)**

1. Identifier les goulots d’étranglement dans l’architecture actuelle du transformer.
2. Tester des optimisations comme le **pruning**, la **quantification** ou le **distillation**.
3. Comparer les performances (vitesse, mémoire, qualité des prédictions).
4. Appliquer l’optimisation sur un modèle réel et mesurer l’impact.

**Optimisation de l'enregistrement des données (50 J 4 x 1 semaine à 2)**

1. Définir un format de stockage adapté (parquet, json, pickle, base de données NoSQL).
2. Mettre en place des mécanismes de compression pour économiser de l’espace.
3. Organiser les données pour accélérer leur accès (indexation, sharding).
4. Vérifier l’intégrité des données et implémenter des sauvegardes automatiques.

**~~Optimisation de la gestion de la mémoire~~**

1. ~~Identifier les parties du pipeline consommant le plus de mémoire (profiling).~~
2. ~~Appliquer des techniques comme~~ **~~offloading~~**~~,~~ **~~mixed precision~~**~~, ou~~ **~~gradient checkpointing~~**~~.~~
3. ~~Tester l’impact sur la latence et la consommation mémoire.~~
4. ~~Automatiser la gestion mémoire avec des outils comme DeepSpeed ou ZeRO.~~

**Optimisation de l’utilisation des modèles et de la gestion de la mémoire (50 J 4 x 1 semaine à 2)**

1. Déployer les modèles en **batch processing** pour éviter les rechargements inutiles.
2. Utiliser des techniques de **caching** des embeddings ou des résultats intermédiaires.
3. Expérimenter des modèles plus petits pour certaines tâches spécifiques.
4. Surveiller l’utilisation mémoire en temps réel et ajuster dynamiquement les ressources.
5. Tester l’impact sur la latence et la consommation mémoire.
6. Automatiser la gestion mémoire

**LLM RAG - Infrastructure et mise en place**

**Mise en place d’une infrastructure virtuelle pour modèle LLM (20j 2 semaines à 2)**

1. Déployer l’environnement avec Docker, Kubernetes ou un serveur dédié.
2. Configurer l’accélération matérielle (GPU, TPU, FPGA).
3. Automatiser le déploiement et la mise à jour du modèle avec CI/CD.

**Mise en place d’une infrastructure RAG (20j 2 semaines à 2)**

1. Mettre en place un pipeline d’indexation des documents pour la récupération.
2. Tester différentes stratégies d’encodage pour améliorer la qualité des résultats.
3. Évaluer la latence et l’évolutivité du système en fonction du volume de requêtes.

**LLM RAG - Fiabilité et évaluation**

**Mesure de l'incertitude dans les réponses du LLM pour renforcer leur fiabilité ( 20J 2 semaines à 2)**

1. Déterminer un score d’incertitude via l’entropie des sorties ou le calibrage des probabilités.
2. Utiliser des approches comme le **MC Dropout** ou les **ensembles de modèles**.
3. Vérifier si les scores d’incertitude sont corrélés aux erreurs du modèle.
4. Ajuster le seuil d’acceptation des réponses selon le niveau de confiance souhaité.

**Implémentation d’algorithmes de score de confiance vis-à-vis des prédictions du RAG( 30 J 3 semaines à 2)**

1. Définir une métrique de confiance (cosine similarity, probability calibration).
2. Tester différentes méthodes d’estimation de la fiabilité des réponses.
3. Mettre en place un système de validation des prédictions avec feedback utilisateur.
4. Ajuster les paramètres du modèle en fonction des retours sur les scores de confiance.

**Analyse et statistiques**

**Création et affichage de statistiques sur l’affectation des destinataires (20 J 2 semaines à 2)**

1. Récupérer les données de distribution des destinataires.
2. Analyser les écarts et les erreurs d’affectation.
3. Mettre en place des visualisations interactives.
4. Automatiser la mise à jour des statistiques en temps réel.

**Création de statistiques pour l’utilisation du RAG(20 J 2 semaines à 2)**

1. Collecter les logs des requêtes et des résultats retournés.
2. Analyser la pertinence des réponses fournies par le modèle.
3. Identifier les requêtes difficiles ou mal interprétées.

**Expérimentation et mise en pratique**

**Mise en pratique expérimentale des travaux théoriques sur les données réelles d'un transporteur (Stéphane J)**

1. Sélectionner un ensemble de données réelles représentatif.
2. Tester le modèle avec différents scénarios réels.
3. Comparer les résultats avec les attentes théoriques.
4. Ajuster les paramètres et optimiser le modèle pour les cas spécifiques.

**Encodage à la volée (40 J 4 semaines à 2)**

1. Déterminer si l’encodage doit être fait en batch ou en temps réel.
2. Sélectionner une architecture adaptée (streaming, lazy loading).
3. Tester la vitesse et la précision de l’encodage.
4. Optimiser le système pour réduire la latence et la consommation mémoire.