Camembert est un modèle de traitement automatique des langues (TAL) basé sur l'architecture Transformer, dérivé de BERT (Google, 2018) et de RoBERTa (Facebook, 2019). Il est spécifiquement conçu pour le français.

**2. Architecture de Camembert**

L'architecture de Camembert repose sur celle des Transformers, introduite par Vaswani et al. en 2017. Voici les composants clés:

* **Embedding Layer**: Convertit les tokens en vecteurs denses.
* **Self-Attention Mechanism**: Permet à chaque token de prêter attention à tous les autres tokens de la séquence pour capturer les dépendances à longue portée.
* **Feed Forward Neural Networks**: Après le mécanisme d'attention, les vecteurs sont passés à travers des réseaux de neurones entièrement connectés pour plus de transformation.
* **Positional Encoding**: Ajoute des informations de position à chaque token pour conserver l'ordre de la séquence.

**3. Tokenization**

La tokenisation dans Camembert utilise l'algorithme **SentencePiece**. Il segmente le texte en sous-mots plutôt qu'en mots complets, résolvant ainsi le problème de vocabulaire hors domaine.

* **Exemple**:
  + Phrase originale: "Il faut que tu trouves l'adresse de l'entreprise de plasturgie"
  + Segmentation SentencePiece: ['▁Il', '▁faut', '▁que', '▁tu', '▁trouve', 's', '▁l', "'", '▁adresse', '▁de', '▁l', "'", '▁entreprise', '▁de', '▁', 'plast', 'ur', 'gie']

**4. Fine-Tuning**

Le fine-tuning est crucial pour adapter Camembert à des tâches spécifiques comme l'analyse des sentiments. Voici les étapes générales:

1. **Préparation des Données**: Les données d'entraînement doivent être bien étiquetées. Par exemple, pour l'analyse des sentiments, les phrases sont étiquetées avec des sentiments comme "positif", "négatif", ou "neutre".
2. **Configuration du Modèle**: Charger le modèle pré-entraîné Camembert et ajouter une couche de classification adaptée à la tâche.
3. **Entraînement**: Utiliser une fonction de perte appropriée (par exemple, la cross-entropy) et un optimiseur (comme Adam) pour ajuster les poids du modèle sur les données d'entraînement.
4. **Évaluation**: Valider la performance du modèle sur un jeu de données de test.

**5. Utilisation de Camembert pour l'Analyse des Sentiments**

L'utilisation de Camembert pour l'analyse des sentiments se fait généralement par les étapes suivantes:

1. **Tokenization**: Convertir le texte en tokens utilisant SentencePiece.
2. **Input Embeddings**: Passer ces tokens par la couche d'embedding pour obtenir des vecteurs.
3. **Transformer Layers**: Appliquer les couches de self-attention et les réseaux de neurones feed-forward.
4. **Classification Layer**: La sortie des couches de Transformers est passée à travers une ou plusieurs couches de classification pour prédire le sentiment.
5. **Softmax Layer**: Appliquer une fonction softmax pour obtenir des probabilités pour chaque classe de sentiment.

**6. Softmax**

La fonction softmax est utilisée pour normaliser les scores de la couche finale en probabilités

**Détail de l'Architecture Transformer**

1. **Input Embeddings**:
   * Les tokens sont convertis en vecteurs denses via une couche d'embedding.
   * Les embeddings de position sont ajoutés pour conserver l'ordre des tokens.
2. **Self-Attention**:
   * Chaque token génère des requêtes (queries), des clés (keys), et des valeurs (values).
   * Les scores d'attention sont calculés en multipliant les requêtes par les clés, puis en appliquant une softmax.
   * Les tokens utilisent ces scores pour agréger les valeurs, permettant au modèle de se concentrer sur différentes parties de la séquence.
3. **Feed Forward Network**:
   * Après l'attention, les vecteurs passent par un réseau de neurones entièrement connecté.
   * Chaque neurone applique une transformation non-linéaire aux vecteurs.
4. **Stacking**:
   * Les couches de self-attention et les feed-forward sont empilées plusieurs fois (par exemple, 12 couches dans BERT base).
   * Chaque couche enrichit les représentations des tokens.