**1. Bag of Words (BoW)**

**Principe :** Le modèle Bag of Words représente un texte par un vecteur de mots, sans tenir compte de l'ordre des mots.

* **Comment ça marche :**
  + Chaque document est transformé en une liste de mots.
  + On crée un vocabulaire (une liste de tous les mots uniques présents dans tous les documents).
  + Chaque document est ensuite représenté par un vecteur où chaque dimension correspond à un mot du vocabulaire. La valeur de chaque dimension est le nombre de fois que le mot apparaît dans le document.
* **Avantages :**
  + Simple et facile à implémenter.
  + Fonctionne bien pour des tâches simples de classification de texte.
* **Inconvénients :**
  + Ignore l'ordre des mots et la sémantique.
  + Peut produire des vecteurs très grands et creux (sparse) pour de grands vocabulaires.

**2. TF-IDF (Term Frequency - Inverse Document Frequency)**

**Principe :** Le TF-IDF améliore le Bag of Words en pondérant les mots en fonction de leur importance dans le document et dans le corpus.

* **Comment ça marche :**
  + **TF (Term Frequency)** : mesure la fréquence d'un mot dans un document.
  + **IDF (Inverse Document Frequency)** : mesure l'importance d'un mot à travers tous les documents. Les mots communs à de nombreux documents ont un IDF faible.
  + **TF-IDF** : Produit du TF et de l'IDF, donnant plus de poids aux mots importants et moins aux mots courants.
* **Avantages :**
  + Réduit l'importance des mots très fréquents et peu informatifs.
  + Meilleure représentation des documents que le simple BoW.
* **Inconvénients :**
  + Ne capture toujours pas l'ordre des mots ni leur contexte.
  + Peut être sensible au bruit (mots rares).

**3. Word2Vec**

**Principe :** Word2Vec est un modèle de plongement de mots qui représente chaque mot par un vecteur dense en capturant les similarités contextuelles.

* **Comment ça marche :**
  + Utilise des réseaux de neurones pour apprendre les représentations vectorielles des mots.
  + Deux architectures principales : CBOW (Continuous Bag of Words) et Skip-Gram.
  + Les mots qui apparaissent dans des contextes similaires obtiennent des vecteurs similaires.
* **Avantages :**
  + Captures des relations sémantiques et contextuelles entre les mots.
  + Vecteurs denses et de taille fixe, indépendamment de la taille du vocabulaire.
* **Inconvénients :**
  + Ne capture pas bien les nuances contextuelles pour des mots polysémiques (mots ayant plusieurs sens).
  + Modèle statique : le sens d'un mot est fixe et ne varie pas selon le contexte.

**4. BERT (Bidirectional Encoder Representations from Transformers)**

**Principe :** BERT est un modèle de langage pré-entraîné qui utilise des transformers pour comprendre le contexte bidirectionnel (gauche et droite) des mots dans une phrase.

* **Comment ça marche :**
  + Entraîné avec des tâches comme la prédiction de mots masqués (Masked Language Model) et la prédiction de la prochaine phrase (Next Sentence Prediction).
  + Utilise des couches de transformers qui permettent de traiter chaque mot en prenant en compte le contexte de tous les autres mots dans la phrase.
* **Avantages :**
  + Captures des contextes riches et bidirectionnels.
  + Très performant sur une large gamme de tâches de NLP (Natural Language Processing).
  + Permet de traiter les nuances contextuelles et les sens variés des mots.
* **Inconvénients :**
  + Très gourmand en ressources computationnelles.
  + Entraînement et déploiement complexes par rapport aux autres techniques.

**En résumé:**

* **Bag of Words** : Compte la fréquence des mots sans tenir compte de leur ordre.
* **TF-IDF** : Pondère les mots en fonction de leur importance locale et globale.
* **Word2Vec** : Représente les mots en tant que vecteurs denses basés sur leur contexte.
* **BERT** : Utilise des transformers pour comprendre le contexte bidirectionnel des mots dans les phrases.