**ANLYSE DE LA BASE PRC (Privacy right clearinghouse)**

C’est une base aux Etats Unis qui rassemble les notifications de violations des données signalées publiquement par les agences gouvernementales américaines dans une base de données unique et consultable.

Cette base contient environ : 76 987 notifications de violation de données, 27 833 lettres de notification, 8.28 milliards de personnes touchées.

Variable dans la base :

* Type d’organisation : BSF, MED, …etc.
* Méthode de violation : Hack, Port, … etc.
* Classification UNKN (inconnu)

Variable intéressante :

* Id
* organization\_type
* incident\_details
* breach\_type
* reported\_date
* breach\_date
* end\_breach\_date
* total\_affected
* residents\_affected
* breach\_location\_city
* breach\_location\_state
* breach\_location\_country

PRC database does not directly provide the loss associated with an event, but reports the number of records (that is the number of user accounts) affected by the breach.

**Analyse descriptive de la base**

**Réseau de neurone**

Préparation des données textuel à analyser via un perceptron multicouche

**Word embedding**

Pour traiter les données textuelles il faut une représentation mathématique plus simple. Le **word embedding** (ou prolongement lexical en français) ; c’est un ensemble de techniques permettant de transformer un mot ou ensemble de mot sous forme de vecteur.

Il s’agit d’un forme intelligente de réduction de dimensionnalité. Car gérer les données de grande dimensions est très difficile.

**Comment ça fonctionne ? :**

* Représentation en contexte : Un peu comme le mot chat – chien, homme-femme, roi reine (utilisé dans le même contexte – les mots qui entour le mot cible).
* Avec un corpus de texte plus étoffé les réprésentation seront plus précis

Tiens compte du texte et du contexte

**Pourquoi est-ce important ?**

L’importance de réprésention vectorielle est qu’elle permet d’effectuer lee calcules algébrique classique et par conséquent facile à entraîner est un objectif de prédiction.

**Différentes méthodes du Word Embedding**

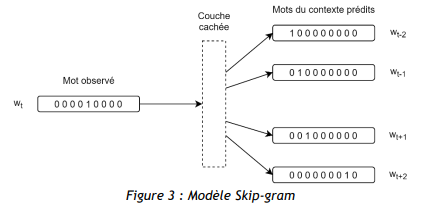
Le bag of Word (sac de mots), permet de décrire un document (il vectorise les mot dans un texte et tient compte de son histogramme dans le corpus d’apprentissage)

CBOW

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Skip-gram



Première approche

**BOW :** ne n’apprend pas le sens des mots.

**Word2Vec :** Groupe de réseau de neurone avec relativement peu de couches cachées (2) et qui vectorise les mots dans un espace de faible dimension deux couches cachées, Il implémente de CBOW et le Skig-gram.

Algorithme du Word2vec

* Les données d’entraînement doivent être mises sur bon format
* # données d'entraînement tokenisation
* p\_1=document\_1.split(" ")
* p\_2=document\_2.split(" ")
* p\_3=document\_3.split(" ")
* corpus=[p\_1,p\_2,p\_3]

# Paramétrage de l'entraînement

model = Word2Vec(min\_count=1, vector\_size=5)

# creation du dictionnaire

model.build\_vocab(corpus)

model.train(corpus, total\_examples=model.corpus\_count, epochs=model.epochs)

model.wv['chat']

#model.wv['souris']

print(model.wv.most\_similar(positive=['chien'] ,topn=3))

Conclusion:

BERT

Nous avons exploré différentes méthodes simples pour effectuer des plongements de mots

**Préparation de la donnée**

Mettre en forme les données dans un réseau de neurone :

* Tokenisation
* Constituer un dictionnaire
* Vectorisation des mots
* Limite (Full one hot encoding) tenseur enorme
* Corriger (on hot vector (on perd l’ordonessement des mots)
* , embedding (petit vecteur portant un sens)

Représentation vectorielle des mots conservant un sens sémantique pour des analyse beaucoup plus fine et compréhensible des textes par la machine.

**Word Embedding**

Techniques plus courantes d’encodage : Word2Vec, GloVe et FastText.

* Word2Vec (Skip-Gram, CBOW) : une méthode populaire pour créer des embeddings basés sur le contexte.
* GloVe (Global Vector) : utilise des statistiques globales pour générer des vecteurs de mots.
* FastText : améliore Word2Vec en considérant les sous-mots.
* BERT

**Techniques de prétraitement en traitement du langage naturel**

Les techniques de prétraitement en NLP sont des étapes indispensables qui organiser les données brutes textuels non structurées, pour les rendre exploitable tout en gardant le maximum d’information, comme le prétraitement des données structurée dans un processus d’analyse des données. La maitrise du prétraitement de données permet de maximiser la performance des modèles NLP.

Majuscules

Accents

Les Stopwords

Stemming

Lemmatisation

Tokenisation

**NLP (Natural Language Processing)**

LSTMs

RNNs

GRUs

Seq2Seq models

Beam Search VS Greedy Search

Pour chaque jeu de données, nous avons appliqué les pré-traitements suivants : — suppression des ponctuations; — suppression des caractères spéciaux; — changement des majuscules en minuscules; — suppression des stop words; — lemmatisation consistant à ne conserver que le radical des mots, pour regrouper sous le même radical tous les mots d’une famille. Chaque texte peut être associé à zéro, une ou plusieurs classes selon le jeu de données. Ces classes sont utilisées comme sortie de la prédiction.

|  |  |  |  |
| --- | --- | --- | --- |
| Word embedding | | Avantages | Inconvénients |
| Word2vec | Skip Gram | Bon pour saisir les relations de mots rares. | Calcul **plus lent**car il prédit plusieurs mots de contexte pour chaque mot cible |
| CBOW | Ce modèle est **plus rapide**à entraîner et fonctionne bien avec **des mots fréquents** | Moins efficace pour les termes peu courants car il a tendance à lisser les représentations de mots rares. |
| GloVe | | L'intégration du contexte local et global par GloVe permet la formation d'intégrations plus significatives qui capturent à la fois les relations à court terme (par exemple, les mots dans des contextes similaires) et les relations sémantiques plus larges (par exemple, les relations globales entre les mots). | Comme Word2Vec, GloVe génère **des représentations vectorielles continues statiques**, ce qui signifie que chaque mot possède le même vecteur, quel que soit son contexte dans différentes phrases. De plus, la création d'une matrice de cooccurrence est complexe et consomme beaucoup de ressources. |
| Transformer | BERT | * Il est plus puissant pour des tâches telles que la réponse aux questions, la classification de texte et la reconnaissance d'entités nommées, car il est efficace pour capturer le contexte dans lequel les mots apparaissent. * Les mots ont des intégrations différentes en fonction du contexte de leur phrase, ce qui rend BERT capable de gérer la polysémie (mots à significations multiples). | * Coûteux en termes de calcul et nécessitant davantage de ressources (en raison de sa nature bidirectionnelle). * La préformation et le réglage fin peuvent prendre du temps. |
| GPT |

Exercice formateur

Suzi, hasnna

Edito