**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 plongement 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 peur 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**

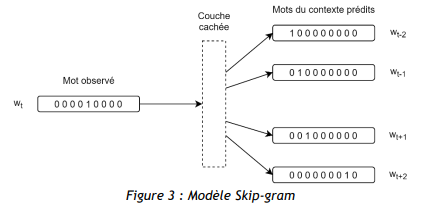
Le bag of Word (sac de mots), permet de décrire un document (il vectorise les mot dans un texte et tient compte de son histogramme dans le corpus d’apprentissage)

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)