**ANLYSE DE LA BASE PRC 2022 (Privacy right clearinghouse)**

1. **Généralité sur la Base PRC**

PRC 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.

Dans la base PRC 2022, on note 9015 incidences d’attaques cyber de type Data Breach déclarés ; les variables recensées sont :

* Date de déclaration
* Type de compagnie
* Ville
* L’Etat
* Type de brèche
* Type d’organisation
* Total des enregistrements
* Description des incidences
* Information sur la source
* Source URL
* L’année de l’attaque
* La latitude
* La longitude

Cette base ne fournit directement pas les pertes associées aux événements d’attaques cyber, mais le total of records (qui es le nombre total des comptes des utilisateurs exposés) touchée par l’attaque.

II-1 Traitement de test

Laurie Chapelle Teamstailor

Le processus de prétraitement du texte a été conçu pour nettoyer, normaliser et structurer les descriptions d’incidents contenues dans un corpus issu d’un jeu de données. Dans un premier temps, les descriptions sont extraites sous forme de liste, en remplaçant les valeurs manquantes par des chaînes vides pour éviter les erreurs lors du traitement.

Chaque texte subit ensuite une série de nettoyages : suppression des sauts de ligne, des espaces superflus, des caractères Unicode non standard, ainsi que des liens URL, des balises HTML et des adresses IP. Les dates, qu'elles soient sous forme écrite ou chiffrée, ainsi que les noms de mois et de jours de la semaine, sont également supprimés afin de neutraliser tout repère temporel dans les textes. Les nombres sont nettoyés, notamment les chiffres avec virgules (ex. "4,000"), et toute la ponctuation ainsi que les caractères non alphabétiques sont éliminés.

Une étape importante consiste à uniformiser les textes : les contractions anglaises sont développées (par exemple, "don't" devient "do not"), puis les phrases sont converties en minuscules. Un point est ajouté à la fin de la phrase si nécessaire pour assurer une structure linguistique correcte.

Le texte est ensuite lemmatisé, c’est-à-dire que les mots sont ramenés à leur forme de base à l’aide d’un analyseur grammatical (POS tagging). Cette opération s’accompagne d’un filtrage des stopwords (mots vides) à partir d’une liste enrichie incluant des connecteurs logiques tels que "however", "therefore", etc. Les mots de moins de trois lettres sont également supprimés afin de conserver uniquement les termes significatifs.

Enfin, pour enrichir l’analyse sémantique, des bigrammes (groupes de deux mots fréquemment associés) sont générés à partir du corpus lemmatisé. Ces combinaisons permettent de mieux capturer le contexte sémantique et les expressions courantes. Le résultat final est une liste de listes de tokens nettoyés et enrichis, prête à être utilisée pour des analyses plus poussées comme la modélisation de thèmes (topic modeling) ou la classification de textes.

**Étapes du prétraitement des descriptions d’incidents**

1. **Extraction des données textuelles**  
   Les descriptions d’incidents sont extraites depuis un jeu de données, avec conversion en liste. Les valeurs manquantes sont remplacées par une chaîne vide afin de garantir l’intégrité du traitement ultérieur.
2. **Nettoyage syntaxique et suppression de contenu non pertinent**  
   Chaque texte est d’abord normalisé par la suppression des caractères spéciaux (sauts de ligne, caractères Unicode comme \xa0, espaces multiples), des balises HTML, des adresses IP, et des URLs. Cela permet de réduire le bruit textuel.
3. **Élimination des repères temporels**  
   Toutes les formes de dates sont supprimées, qu’elles soient numériques (ex. : "12/03/2022") ou textuelles (ex. : "March 12, 2022"). Les noms de mois et les jours de la semaine sont également retirés afin d’éviter toute influence temporelle dans l’analyse.
4. **Uniformisation linguistique**  
   Les contractions en anglais sont développées (par exemple, "can't" devient "cannot") afin de faciliter le traitement morphologique. L’ensemble du texte est converti en minuscules, et un point est ajouté à la fin de chaque phrase si besoin, afin d’assurer une structure linguistique cohérente.
5. **Suppression de la ponctuation et des chiffres**  
   Toute la ponctuation ainsi que les caractères numériques sont éliminés. Cette étape permet de ne conserver que les mots significatifs pour l’analyse sémantique.
6. **Lemmatisation avec étiquetage grammatical (POS tagging)**  
   Les textes sont tokenisés puis chaque mot est associé à sa catégorie grammaticale (nom, verbe, adjectif, etc.) à l’aide de l’étiquetage POS. La lemmatisation est ensuite effectuée selon ces catégories, permettant de ramener chaque mot à sa forme canonique.
7. **Filtrage des mots vides (stopwords)**  
   Une liste de mots vides standard (en anglais) est enrichie de connecteurs logiques (ex. : "however", "furthermore", "indeed"). Tous les mots appartenant à cette liste sont supprimés. Les mots de deux lettres ou moins sont également retirés, considérés comme peu informatifs.
8. **Construction de bigrammes**  
   Afin de capturer des associations sémantiques fréquentes, des bigrammes (paires de mots souvent co-occurrents) sont générés à l’aide d’un modèle de phrases. Ces combinaisons enrichissent la représentation du texte.

**II-2 Analyse descriptive de la base PRC**

**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**

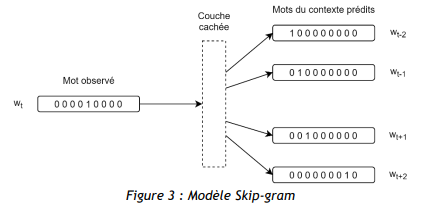
Le bag of Word (sac de mots), permet de décrire un document (il vectorise les mot dans un texte et tient compte de son histogramme dans le corpus d’apprentissage)

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