*khallil doudou*

**NLP Knowledge**

Here is all my collected technique for Natural Language Processing technique

**Bag of Words**

Un bag of words est une technique qui prends un document le découpe en mots, et pour chaque label donné il va compter le nombre de mots dans chacun de ses labels. Ensuite avec ce compte, on peut prendre une nouvelle phrases labelisé, créer un vector pour cette phrases de la taille du vocabulaire, et utiliser le one hot encoding sur chaque x du vector où est un mot de la phrase labélisé est dans le vocabulaire, ce qui nous donne un vector avec beaucoup de 0 et quelques 1, en utilisant ces vectors qu’on dit de type Sparse, on peut entrainer un réseau de neurones en passant ces énormes vecteurs dans l’input layer.

**Words Embedding**

**Word2Vec**

L’écologie c’est bon pour la santé

ecologie est bon santé

L’écologie c’est pour les prolo

ecologie est bon pour santé prolo

There are two main training algorithms that can be used to learn the embedding from text; they are Continuous Bag-of-Words (CBOW) and skip grams. We will not get into the algorithms other than to say

that they generally look at a window of words for each target word to provide context and in

turn meaning for words.

Glove lui se concentre sur les caractères autour du mot

**Voir GaussianEmbedding → Word2Gaussian**

nucléaire = gaz\_a\_effet\_de\_serre

- Un mot est reconnu en fonction des mots qui l’entoure

- Prendre un mot et compter sa fréquence dans X Documents de différents types Y,

si la fréquence d’apparition du mot dans chaque document se rapproche de la fréquence d’apparation dans un autre mot alors on peut déduire un rapprochement

**np2Vec**

L’idée est de pré-labalisé les np dans la phrase avant d’entraîner le model d’embedding afin d’avoir une compréhension plus précise de la relation entre les np en général

Texas, Starcraft, Nivea, Pierre

**End-to-end Memory Network (MemN2N)**

Le but est de pouvoir prendre des phrases ordonné dans le temps, et en créant des embedding vectors pour chaque mot et en ajoutant un vector de (time embedding) pour chaque phrase en fonction de leurs ordre d’apparition dans le texte, on peut alors définir répondre a des questions simples sur le texte. Good pour : Q&A

**Variantes** :

-Key value end to end memory network (Kvmemn2n)

**ReadComprehension**

LSTM and Answer Pointer network, which allow to understand the text and to answer question

**BIST Dependency**

Permet d’apprendre les dépendances entres les mots dans une phrase, en s’entrainant de base avec un réseau bidirectionnel lstm sur un dataset de phrase labélisées

**Chunker (PosTagging)**

PosTagging des noms, prénoms et les différents types d’objets du langage dans une phrase

**Named Entity Recognition (Spacy NER) ( DialogFlow gère ça avec sys.xxx)**

Extraction basic information on a sentence like the Name of a city, the location ,etc ...

! DialogFlow permet déja ça avec le dictionnaire de synonymes

**NP segmentation**

Allow to understand when a phrase is a descriptive structure or a collocation structure, this means that for a np “dog” a tree is displayed for each semantic meaning of the word dog, then base of the MLP result we can assign the np to a tree result of dog word and define the semantic result

**Intent Extraction**

The goal is to first find the intent, then fills the parameters of the intent (date, location,etc…). There is the open source SNIPS dataset, and ATIS from Microsoft, see

**Word sense**

Prédir une définit d’un mot a partir d’un dataset dictionnaire (definition+hypernyme) et surtout cela permet d’établir une classement du sens le plus problable en fonction de la fréquence d’utilisation de ce sens avec le mot.

**HandCraft Dialog (DialogFlow)**

Trigger un event en fonction de la ressemblance de l’input avec les phrases d’entrainement de l’intent supposé trigger l’event.