Activités du Lundi 29 avril au Vendredi 10 mai :

Du 29 avril au 3 mai :

- **Création de regex pour transformer la sortie des dépendances en format JSON**
  - Cela nous permettra de parcourir les dictionnaires de dictionnaires lors de l'ouverture des fichiers JSON.
    - Nous aurions pu directement faire les traitements sans mettre le résultat des dépendances mais comme l'analyse en dépendance est assez longue il était préférable de tout mettre dans des fichiers puis les ouvrir ultérieurement pour les traiter automatiquement et rapidement.

- **Résolution de problèmes :**
  - Il manquait des events / timexs / signaux dans les CSV fusionnés. Les problèmes venaient essentiellement de la tokenization qui a été améliorée de façon à ce qu'il n'y ait plus d'events et signaux manquants. Les timex nécessitent encore des améliorations.

Du 6 mai au 10 mai :

- **Résolution des problèmes des timex manquants**
- **Application sur python des regex précédemment créées pour transformer la sortie en dépendances au format JSON (extrait d'un fichier au format JSON plus bas)**
  - Dans les dépendances au format JSON nous avons deux informations importantes :
    - L'identifiant de phrase (valeur de Dependencies)
    - L'identifiant de mot (valeur de l'id de phrase)

    Ces infos nous permettrons donc de retrouver les events à partir du **csv des identifiants artificiels** :

	docID	word	idWord	idSent	id	event	idEvent	timex	idTimex	signal	idSignal
0	APW199808(	1998-08-07#t	1	1	t0			1998-08-07#t0	1		
1	APW199808(	.	2	1							
2	APW199808(	Explosions	1	2							
3	APW199808(	rock	2	2							
4	APW199808(	U.S.	3	2							
5	APW199808(	embassies	4	2							
6	APW199808(	in	5	2							
7	APW199808(	Tanzania	6	2							
8	APW199808(	,	7	2							
9	APW199808(	Kenya	8	2							
10	APW199808(	.	9	2							
11	APW199808(	NAIROBI	1	3							
12	APW199808(	,	2	3							
13	APW199808(	Kenya	3	3							
14	APW199808(	(	4	3							
15	APW199808(	AP	5	3							
16	APW199808(	)	6	3							
17	APW199808(	_	7	3							
18	APW199808(	Suspected	8	3							
19	APW199808(	bombs	9	3							
20	APW199808(	exploded#e1	10	3	e1	exploded#e1	1				
21	APW199808(	outside	11	3							
22	APW199808(	the	12	3							
23	APW199808(	U.S.	13	3							
24	APW199808(	embassies	14	3							
25	APW199808(	in	15	3							
26	APW199808(	the	16	3							
27	APW199808(	Kenyan	17	3							
28	APW199808(	and	18	3							
29	APW199808(	Tanzanian	19	3							
30	APW199808(	capitals	20	3							
31	APW199808(	Friday#t1	21	3	t1			Friday#t1	2		
32	APW199808(	,	22	3							
33	APW199808(	killing#e2	23	3	e2	killing#e2	2				

Par exemple, le premier event « exploded » se trouve dans la phrase dont l'identifiant est « 3 » à la position de mot numéro « 10 ». Nous n'aurons pas de problèmes de tokenization entre la tokenization pour créer les id artificiels et pour les dépendances car nous utilisons la même tokenization.

On utilise dependency_parser.parse_sents() qui attend, pour chaque fichier, une liste de listes de phrases tokenizées, elles-même tokenizées en mots. Cette liste de liste nous la récupérons lors de la tokenization dans la fonction de création des identifiants artificiels.

Exemple :

[['19980108', '.'], ['On', 'the', 'other', 'hand', ',', 'it', "'s", 'turning', 'out', 'to', 'be', 'another', 'very', 'bad', 'financial', 'week', 'for', 'Asia', '.'], ['The', 'financial', 'assistance', 'from', 'the', 'World', 'Bank', 'and', 'the', 'International', 'Monetary', 'Fund', 'are', 'not', 'helping', '.'],…]

- **Correction de regex / replace**
    - En passant au format JSON, je remplaçais les « ‘ » par des « ’’ » pour que la syntaxe soit correcte mais ça a aussi modifié les « ‘s » en « ’’s » (de même pour les « n’t », les « ‘re » et « ‘d »)

- **Transformation de tous les fichiers au format json + vérification qu'ils sont valides.**


```
{
    "Dependencies": {
        "1": [{
            "0": {
                "address": 0,
                "ctag": "TOP",
                "deps": [{
                    "defaultdict(<class 'list')": {
                        "root": [1]
                    }
                }],
                "feats": null,
                "head": null,
                "lemma": null,
                "rel": null,
                "tag": "TOP",
                "word": null
            },
            "1": {
                "address": 1,
                "ctag": "CD",
                "deps": [{
                    "defaultdict(<class 'list')": {}
                }],
                "feats": "_",
                "head": 0,
                "lemma": "_",
                "rel": "root",
                "tag": "CD",
                "word": "19980108"
            }
        }],
        "2": [{
            "0": {
                "address": 0,
                "ctag": "TOP",
                "deps": [{
                    "defaultdict(<class 'list')": {
                        "root": [8]
                    }
                }],
```