Analyse de la dynamique des mêmes textuels avec Memetracker

Gabriel Chicheportiche, Mamadou Mustapha, Thomas Debernardi, Hussein El Gouch

Avril 2025

Introduction

Ce projet s'inscrit dans l'exploration des dynamiques de diffusion de l'information sur le web. Nous utilisons **Memetracker**, une base de données massive issue de l'article de *Leskovec et al. (2009)*, qui suit des fragments textuels circulant dans la presse et les blogs. L'objectif est de comprendre comment les mêmes apparaissent, se propagent, et èvoluent dans le temps, à partir d'une analyse computationnelle du langage.

L'enjeu est à la fois technique et sociologique : comment automatiser l'identification de phrases répétées (les mêmes), regrouper leurs variantes, puis tracer leur diffusion temporelle et leur impact médiatique ?

Nous avons d'abord développé comme approche principale une **vectorisation TF-IDF** couplée à un **bucketing MiniBatchKMeans**, permettant de gérer efficacement des centaines de milliers de phrases. En comparaison, l'approche classique par distance de Levenshtein (edit distance) sert de baseline pour illustrer le compromis précision vs performance.

1 Approche principale : Vectorisation TF-IDF Bucketing

1.1 Pipeline de traitement

- Extraction de phrases depuis `quotes_2008-08.txt`
- Nettoyage : liste locale de stopwords, longueur minimale de 2 mots, fréquence minimale de 2 occurrences
- TF-IDF (ngrams 1–2, $\min_d f = 2) etr duction de dimension Mini Batch KMeans en 500 buckets (random_stat$ 42)

— Dans chaque bucket : construction de graphes partiels via `overlap(p,q,k)`

Listing 1 – Clustering TF-IDF+Buckets et similarité par chevauchement

```
vectorizer = TfidfVectorizer(ngram_range=(1,2), min_df=2)
X = vectorizer.fit_transform(phrases)
kmeans = MiniBatchKMeans(n_clusters=500, random_state=42)
labels = kmeans.fit_predict(X)

buckets = defaultdict(list)
for p, lbl in zip(phrases, labels):
    buckets[lbl].append(p)

for bucket in buckets.values():
    for p in bucket:
        for q in bucket:
            if p!=q and overlap(p,q,8):
                G.add_edge(p,q)
```

Cette méthode offre un **gain de temps** de l'ordre de 10× par rapport à la comparaison brute, tout en conservant une précision suffisante pour la plupart des mêmes.

2 Approche alternative : Distance de Levenshtein

Pour référence, nous avons implémenté l'approche classique :
— Sélection de 500 phrases
— Graphe complet avec liens dès que $d_{rel} = \frac{d_{Lev}}{\max(|p|,|q|)} \leq 0.3$

Listing 2 – Graphe par distance d'édition relative

```
for i in range(n):
  for j in range(i+1,n):
    d = edit_distance(p[i],p[j])
    if d/max(len(p[i]),len(p[j])) <= 0.3:
      G.add_edge(p[i],p[j], weight=1-d/max(...))
```

Limites : complexité $O(n^2)$, prohibitif au-delà de quelques milliers de phrases, mais granularité maximale sur les petites modifications.

3 Comparaison des résultats

Scalabilité TF-IDF+Buckets ≫ Levenshtein

Précision Levenshtein » TF-IDF (variations minuscules)

Bruit Buckets peut regrouper des phrases non reliées (vocabulaire commun)

Performance TF-IDF+Buckets traite l'équivalent de 10% du dataset, Levenshtein reste limité à <0.1%

4 Visualisations Approche principale

cf plus bas pour les visualisations. Nous retrouvons dans nos données certains mèmes déjà décrits dans memetracker.pdf, notamment celui issu du discours d'Obama fin août dans la perspective de son élection. Cela montre d'abord la validité de notre méthode : nous parvenons à extraire et reproduire des mèmes identifiés dans l'étude de référence. Ensuite, cela souligne la portée sociologique du discours politique : un même extrait peut dominer le débat médiatique, révélant des logiques d'agenda-setting et de viralisation même dans un contexte antérieur. Nous n'avons pas traité les spams, certains mêmes le sont : "best shopping experiences" De nombreux mêmes ne sont pas en anglais comme "huellas de la campa a del desierto"

5 Lien avec l'article de Leskovec et al. (2009)

— **Courbes de fréquence** : nos courbes montrent les mêmes phases : *emergence*, *pic*, *décroissance*.
— **Clustering canonique** : notre approche est simplifiée mais cohérente avec l'idée de groupe de variantes.
— **Pic du 28/08/2008** : probable référence au discours de Barack Obama à la Convention Démocrate.

6 Analyse temporelle

Suivi quotidien des 10 plus grands clusters (août 2008) :

— Pics liés aux conventions politiques (discours du 28/08)
— Phase **build-up rapide** puis décroissance exponentielle
— Décalage de 2.5h entre mainstream et blogs

7 Analyse socio-économique

— Mise en lumière des processus d'« agenda-setting » En quantifiant les pics d'occurrence de certaines phrases, on observe comment les médias et les blogs convergent sur quelques « mèmes » clés — souvent issus de discours politiques majeurs — avant de rapidement passer à autre chose. Cela illustre le pouvoir des acteurs médiatiques à structurer l'attention publique et à produire des agendas partagés, confirmant le rôle central du journalisme et des relais blogosphériques dans la construction des priorités politiques et sociales.
— Rôle structurant de la technologie dans la diffusion de l'information La comparaison entre approche par distance d'édition (rigoureuse mais limitée à un sous-ensemble de 500 phrases) et vectorisation/bucketing (scalable mais plus bruitée) montre à quel point les choix techniques (nettoyage, métriques, réduction de dimension) orientent les résultats et, par conséquent, les conclusions qu'on peut tirer sur les dynamiques sociales. Cela met en garde contre une croyance excessive dans l'objectivité « automatique » : les modèles numériques portent en eux des arbitrages qui déterminent la forme du « laboratoire social » qu'ils produisent.
— Surveiller en continu les flux (blogs, presse, réseaux sociaux) permet de détecter instantanément un pic de mème et de le relayer avant qu'il ne retombe. Seuls les médias ou équipes disposant d'outils et de personnel automatisés y parviennent, creusant une inégalité numérique.

Conclusion

L'approche TF-IDF+Buckets constitue la méthode principale pour un corpus massif, garantissant un équilibre temps/précision. La distance de Levenshtein reste indispensable pour de petites collections ou un affinage précis.

Contributions individuelles

Gabriel Chicheportiche : Aux premier abords je trouvé notre sujet assez peu intéressant et je ne voyais pas trop son intérêt en plus c'étaient des outils que je ne maitriser pas très bien. Puis une fois le travail de groupe lancé j'ai compris l'intérêts du sujet et sa complexité. J'ai trouvé très intéressant le fait de mettre en œuvre des concepts d'informatique vue plus tôt dans l'année et l'appliqué à un cas concret. Je fus assez étonné de la pertinence de méthode informatique pour les sciences sociale, les résultats qu'on trouve explique beaucoup de phénomène et nous montre bien l'importance des réseaux sociaux/ blog (même à l'époque) . Ça illustre bien le phénomène de buzz, et comment il fonctionne, il y a une quantification/ modèle théorique mathématique sur phénomène sociaux et je trouve cela super intéressant. Nous avons des trouvé des résultats un peu diffèrent de l'article car notre modèle n'tat pas exactement pareil ça nous montre que tout ces résultats qu'on voit sur internet dépendent énormément du modèle et qu'une petite variation peut changer beaucoup de chose et donc qu'il faut faire attention quand on lit des articles avec des grosse bases de données. En conclusion j'ai beaucoup aimé de ce projet et j'ai appris beaucoup de choses que ce soit en info ou en science sociale. visualisations.

Mamadou Mustapha : Ce projet m'a permis d'allier informatique et science sociale. J'étais emballé dès le début sur le memeTracker car je voulais vraiment savoir comment modéliser une

évolution de "meme" (qui relève plus d'un aspect science sociale") à travers des outils plus formels et techniques tels que l'informatique. J'ai trouvé intéressant aussi de constater à quel point ces choix techniques (comme le type de distance, le nettoyage des textes, le choix arbitraire de "stepwords") influencent directement les résultats. Cela montre une certaine neutralité des outils numériques. Ce projet m'a aussi permis de m'améliorer sur le plan technique avec le découverte de nouveaux outils python et de prendre plus en considération la manière de coder, de telle sorte de réduire le temps de calcul.

Thomas Debernardi : Ce projet m'a permis de mieux comprendre les défis liés à l'analyse de données textuelles en grande quantité. J'ai trouvé intéressant de devoir adapter notre approche pour tenir compte des contraintes de temps de calcul. Par exemple, utiliser des "buckets" pour regrouper les phrases proches avant de les comparer a été une solution efficace, même si cela impliquait une légère perte de précision. En travaillant sur les données, j'ai constaté à quel point des problèmes comme les doublons ou les spams peuvent fausser l'analyse. Malgré des traitements pour nettoyer la base de données, il restait toujours une part de bruit difficile à supprimer, ce qui compliquait l'interprétation finale. Un aspect que j'ai trouvé moins satisfaisant est que nous ne sommes pas arrivé à retrouver les memes présentés dans l'article de référence. Cela montre qu'il faut accepter une part d'incertitude dans ce genre de projet : malgré les efforts techniques, certains facteurs extérieurs restent difficiles à contrôler. Au final, ce projet m'a permis de progresser sur l'analyse de données textuelles, sur l'optimisation d'algorithmes simples, et sur la prise de recul face aux résultats.

Hussein El Gouch : J'ai particulièrement apprécié la dimension pluridisciplinaire de ce projet, qui m'a permis de découvrir les sciences sociales computationnelle un univers auquel je ne m'attendais pas mais que j'ai trouvé passionnant. J'ai également pris plaisir à me familiariser avec de nouvelles bibliothèques Python (NetworkX, scikit-learn) et à optimiser des pipelines de traitement de texte. En revanche, faire tourner l'ensemble sur mon ordinateur personnel a mis ma machine à rude épreuve... Ce travail m'a surtout ouvert les yeux sur l'influence déterminante des choix algorithmiques et de prétraitement : selon que l'on utilise la distance de Levenshtein, la vectorisation TF-IDF ou les bucketing, on peut obtenir des résultats radicalement différents. Cela illustre combien les données « parlent » différemment selon le traitement qu'on leur applique, et souligne les inégalités potentielles dans l'interprétation des résultats, un même ensemble de données peut « mentir » ou « dire la vérité » selon les paramètres du pipeline. Enfin, j'ai vraiment apprécié notre démarche collaborative : partager des retours d'expérience, et comparer nos graphes « classiques » Ce projet m'a non seulement enrichi techniquement, mais il m'a aussi sensibilisé à la responsabilité des data scientists : chaque décision de nettoyage ou de métrique influe sur la connaissance que l'on produit.

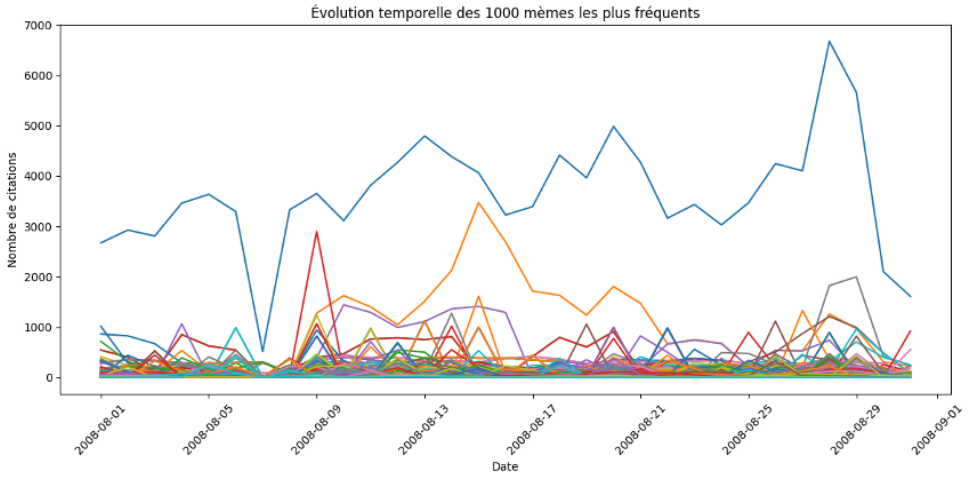

FIGURE 1 – Evolution temporelle des 1000 mêmes les plus fréquents août 2009$\leq$ 0.3

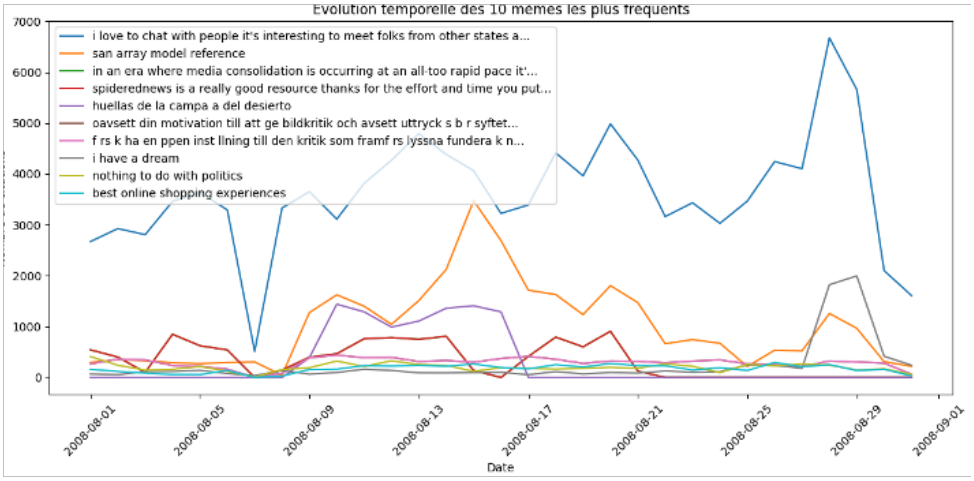

FIGURE 2 – Evolution temporelle des 10 mêmes les plus fréquents août 2009 ≤ 0.3