Extraction de chaînes cohérentes en vue de reconstuire la Trajectoire de l'information

Charles Huyghues-Despointes, Leila Khouas, Julien Velcin, Sabine Loudcher

Extraction de chaînes cohérentes en vue de reconstuire la Trajectoire de l'information

Charles Huyghues-Despointes[*,**], Leila Khouas[**], Julien Velcin[*] et Sabine Loudcher[*]

[*]Laboratoire ERIC
[**]Bertin IT

Résumé. Sur Internet, l'information se propage en particulier au travers des documents textuels. Cette propagation soulève de nombreux défis : identifier une information, suivre son évolution dans le temps, comprendre les mécanismes qui régissent sa propagation, etc. Étant donné un document parmi un grand corpus dans lequel de nombreuses informations circulent, pouvons-nous retrouver les chemins empruntés par l'information pour arriver à ce document ? Nous proposons de définir la notion de trajectoire comme l'ensemble des chemins le long desquels de l'information s'est propagée et nous élaborons un algorithme pour l'estimer. Nous avons proposé et mis en œuvre un protocole d'évaluation permettant de déterminer humainement la qualité des trajectoires calculées. Nous montrons que les évaluations concordent pour une majorité des chemins et que notre algorithme est efficace pour retrouver les bons chemins.

1 Introduction

L'information se propage. Lorsqu'elle est reçue, une information est ingérée, nuancée, et reformulée pour être à nouveau transmise. Cette propagation se déroule à tous les niveaux de communication : lors d'une conversation, à la radio, à la télévision, mais aussi lorsque nous publions du contenu, par exemple sur Internet. Les documents que nous partageons, que ce soit du texte, de l'audio ou de la vidéo, contiennent de multiples informations qui proviennent d'autres documents. L'étude de la propagation de l'information a trouvé un essor au début des années 2000 dans de nombreux domaines, notamment avec l'émergence des réseaux sociaux (Kempe et al., 2003) et l'arrivée des médias et des blogs sur Internet (Leskovec et al., 2009). Nous identifions plusieurs problèmes soulevés par la littérature. Un premier réside dans l'identification et la traque des informations. C'est un problème d'autant plus difficile lorsque l'information évolue dans son fond ou dans sa forme. Un autre problème est de retrouver le réseau sous-jacent à la propagation, retrouver quels sont les différents relais de l'information. Le dernier problème que nous identifions est la recherche des sources d'une information. Nous proposons de traiter un problème différent, à savoir retrouver les chaînes de documents que l'information emprunte. Nous voulons retrouver ces chaînes sans identifier à priori les informations qui y circulent. Cela diffère de la plupart des travaux qui cherchent à retrouver les sources d'une information : nous ne partons pas d'une information identifiée dont on voudrait retrouver la propagation.

Dans un corpus, il existe des chaînes de documents le long desquelles au moins une information s'est propagée. Nous nommons l'ensemble de ces chaînes la Trajectoire de l'information. La Trajectoire dépend du phénomène de propagation, or le déroulement exact de ce phénomène nous est inconnu. Nous n'observons que des documents qui sont des témoignages de cette propagation. Aussi, estimer avec exactitude la Trajectoire est une tâche qui semble difficile alors que cela donnerait lieu à de multiples applications. D'un document, nous connaîtrions les documents l'ayant directement influencé. Nous pourrions aussi naviguer dans la généalogie des documents. Cela révélerait des liens plus lointains entre les documents et permettrait aussi de remonter à leurs sources d'information. Nous aurions une structure pour identifier les informations qui se propagent avec un contexte plus précis, pour comprendre leurs mutations et la manière dont elles se comportent au sein du corpus. C'est une manière d'aborder le problème de la propagation d'information qui n'a pas, à notre connaissance, été traitée dans la littérature. Nous inscrivons notre travail comme un premier pas dans cette direction.

Nous proposons de calculer des ensembles de chaînes de documents, que nous appelons des trajectoires, comme des approximations de la Trajectoire. La Trajectoire n'explicite pas l'information qui se propage le long d'une de ses chaînes mais témoigne juste de l'existence de cette information. Puisque nous ne sélectionnons pas à priori les informations, nous nous basons sur une mesure de la similarité sémantique de documents textuels. Nous avons élaboré un algorithme pour calculer de telles chaînes cohérentes. Aussi, nous proposons un protocole d'évaluation par l'humain pour estimer la qualité des chaînes. Nous constatons que les évaluateurs sont d'accord sur la manière de qualifier la majeure partie des chaînes. Nous interprétons ce constat ainsi : l'humain est capable d'estimer la qualité d'une chaîne ce qui nous conforte dans le fait que le problème est bien défini et justifie notre volonté d'automatiser le processus à l'aide de notre algorithme. Nous montrons également que notre algorithme construit de bonnes chaînes vis-à-vis de nos évaluations.

Dans une première partie, nous nous intéressons aux travaux existants sur la propagation de l'information. La deuxième partie revient sur la notion de Trajectoire et nous y expliquons notre algorithme pour le calcul de chaînes cohérentes. Nous présentons ensuite notre protocole d'évaluation de chaînes et les résultats qui en découlent. Nous concluons sur les perspectives d'utilisation d'une trajectoire et nos idées pour améliorer et approfondir notre approche.

2 Travaux sur la propagation de l'information

Nous détaillons dans cette section trois problèmes proches du nôtre qui ont été traités dans la littérature. Nous expliquons également en quoi notre problème diffère de ceux-ci.

Un premier problème réside dans l'extraction et l'identification de fragments d'informations (comme une citation ou une URL) et la manière dont ils évoluent. Nous notons les travaux menés par Leskovec et al. (2009) où les auteurs cherchent ces fragments et leurs mutations, qu'ils nomment des mèmes, dans des articles de presse et de blog. Des travaux soulignent la difficulté de la tâche et proposent des méthodes de résolution comme Snowsill et al. (2011) et Yang et Zha (2013). La définition de ces mèmes a donné lieu à de nouvelles analyses sur la manière dont ils se propagent. Myers et Leskovec (2012) et Zarezade et al. (2017) étudient ainsi la manière dont s'entraident ou s'entravent les informations dans le phénomène de propagation.

Un autre problème est de retrouver le graphe support de la propagation : c'est le graphe dont les sommets sont des diffuseurs d'informations (comme des auteurs ou des utilisateurs

d'un réseau social). Un arc dans ce graphe signifie que de l'information circule d'un sommet vers l'autre. Plusieurs travaux ont cherché à estimer le graphe support lorsque l'on a une connaissance explicite des sommets du graphe où telle ou telle information est passée. La plupart des travaux traitent cette question comme un problème d'optimisation continu, notamment Gomez-Rodriguez et al. (2011), Zarezade et al. (2017) ou Zhao et al. (2015).

D'autres chercheurs posent comme objectif de retrouver les sources de la propagation d'une information dans un réseau social comme Prakash et al. (2012), Pinto et al. (2012) et Farajtabar et al. (2015).

Notre travail diffère des travaux évoqués sur plusieurs points. En effet, nous cherchons à construire des chaînes de documents le long desquelles il est plausible qu'une ou plusieurs informations se soient propagées. Ce n'est pas un graphe, cela ne correspond pas non plus aux documents contenant une information particulière puisqu'il s'agit d'un ensemble de chaînes de documents. De plus, nous ne sélectionnons pas l'information qui circulerait à priori le long de nos chaînes. Nous ne nommons jamais dans notre approche l'information qui se propage le long d'une chaîne et nous montrons qu'il est possible d'obtenir des chaînes cohérentes sans avoir à le faire. L'identification de l'information qui circule est une étape ultérieure pour notre travail alors que c'est un pré-requis dans les problèmes que nous citons précédemment.

3 Notre approche : extraction des chaînes de propagation

Nous divisons cette partie en plusieurs sections. La première détaille notre objectif, son contexte et précise les termes que nous utilisons. La deuxième détaille notre approche et l'algorithme que nous proposons. Nous discutons ensuite de sa complexité et de nos stratégies heuristiques. Après un exemple de déroulement de l'algorithme, nous commentons le critère de cohérence et la notion d'attachement.

3.1 Objectif

Le contexte du problème est le suivant : nous analysons un ensemble de documents textuels (un corpus) dont nous connaissons certaines méta-données, comme la date de publication, les auteurs, etc. Chaque document a un contenu et véhicule donc des informations.

Nous appelons **une information** une unité sémantique qui peut être arbitrairement complexe, connotée et ambiguë. Les exemples sont nombreux en passant par "Le ciel est dégagé cet après-midi." ou "SOS". Usuellement, les travaux de propagation choisissent un sous-ensemble de l'information qui est identifiable syntaxiquement comme les phrases, les citations ou les hashtags sur Twitter.

Les informations **se propagent** dans notre corpus, c'est-à-dire que durant le processus de création des documents, les auteurs récupèrent, interprètent et reformulent différentes informations issues de documents antérieurs du corpus. Cela a deux implications. La première est que notre corpus est toujours lacunaire au sens de la propagation : il manque toujours des intermédiaires qui sont en dehors du radar. La seconde est que les informations sont susceptibles de muter.

Une **mutation d'une information** est une information dérivée de telle sorte qu'on puisse déterminer un lien sémantique fort avec l'information dont elle dérive. Dans leurs travaux sur la traque de mèmes, Leskovec et al. (2009) et Snowsill et al. (2011) suivent par exemple les

```
1 Donald Trump Is Also an Outlier in Political Science
2 Donald Trump Is Forcing Ted Cruz to Rewrite His Playbook
3 The Republican Establishment Is Losing at Its Own Game
4 Ted Cruz and Allies Work to Halt Donald Trump's Gains
5 Donald Trump Blasts Ted Cruz Using the Senator's Own Words
```

FIG. 1: Chaîne de propagation plausible tirée d'articles du New York Times (titres affichés)

troncatures comme les mutations des citations ("je ne sais rien" est une troncature du "Tout ce que je sais, c'est que je ne sais rien."). Cela implique qu'on ne peut pas comparer les documents sur la base d'une correspondance exacte de l'information véhiculée.

On appelle **chaîne de propagation** une chaîne de documents le long de laquelle au moins une information s'est propagée au sens évoqué ci-dessus. Nous appelons **la Trajectoire** l'ensemble des chaînes de propagation. C'est une construction abstraite qui exprime la manière dont s'est réalisée la propagation. Nous ne cherchons pas à la calculer explicitement, mais nous cherchons une approximation plausible de celle-ci.

Nous appelons **trajectoire** un ensemble de chaînes de documents. On dit qu'une chaîne de documents est une **chaîne de propagation plausible** si des évaluateurs humains s'accordent pour dire qu'il a pu y avoir une propagation d'information le long de cette chaîne. Un exemple de chaîne de propagation plausible est donné en Fig. 1 et un exemple de trajectoire est donné en Fig. 2.

Notre objectif est le suivant : calculer une trajectoire contenant le plus de chaînes de propagation plausibles, c'est-à-dire cohérentes, et le moins de chaînes non plausibles.

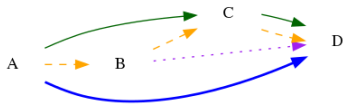


FIG. 2: Exemple de trajectoire avec 4 chaînes : ACD, ABCD, AD et BD

3.2 Algorithme d'extraction de chaîne

Notre approche consiste à parcourir toutes les chaînes possibles et à sélectionner celles qui satisfont un certain critère de cohérence. Construire toutes les chaînes sur n documents a une combinatoire élevée : il y en a au moins $\Omega(2^n)$. En effet, il y a au moins autant de chaînes que de sous-ensembles du corpus de documents. Nous utilisons plusieurs propriétés pour réduire cette combinatoire. La première est une propriété de croissance : une chaîne $ABCD$ est une chaîne de propagation à condition que CD le soit aussi. Sinon, cela veut dire qu'une information circule le long de $ABCD$ sans qu'aucune ne circule le long de CD. Ainsi, si CD ne satisfait pas notre critère de cohérence, nous n'explorons pas les chaînes qui passent par CD. La seconde exploite la date de publication des documents. Effectivement, la propagation d'information est un phénomène temporel, une information se propage toujours du document le plus ancien vers le document le plus récent.

Nous calculons pour chaque document D les chaînes qui finissent en D, que nous notons $FinishIn(D)$. Pour cela, nous calculons l'ensemble des chaînes candidates pour D, que nous notons $Candidates(D)$. Les chaînes candidates pour D sont toutes les chaînes formées de documents publiés avant D. Étant donnée notre propriété de croissance, nous parcourons les chaînes qui finissent en C à la condition que la chaîne CD satisfasse notre critère de cohérence. Une fois tous les candidats accumulés, les chaînes qui finissent en D sont le résultat de notre stratégie de sélection $select$. La trajectoire calculée T est l'union de toutes les chaînes calculées. Le pseudo-code de l'algorithme est donné en Algorithme 1.

Data : un corpus de document $Corpus$, une stratégie de sélection $select$
Result : T l'ensemble des chaînes calculées
$Treated \leftarrow \emptyset$;
$T \leftarrow \emptyset$;
for $D \in Corpus$ *par date de publication croissante* **do**
  $FinishIn(D) \leftarrow \emptyset$;
  $Candidates(D) \leftarrow \{D\}$;
  for $C \in Treated$ *vérifiant* $date(C) < date(D)$ *et* $select(\{CD\}) \neq \emptyset$ **do**
    $Candidates(D) \leftarrow Candidates(D) \cup \{chain.D, chain \in FinishIn(C)\}$;
  end
  $FinishIn(D) \leftarrow select(Candidates(d))$;
  $T \leftarrow T \cup FinishIn(D)$;
  $Treated \leftarrow Treated \cup \{D\}$
end
return T

Algorithme 1 : Calcul d'une trajectoire de l'information

3.3 Complexité et heuristique de sélection

La complexité dans le pire cas de l'algorithme est inchangée. Il est toujours possible de construire l'ensemble de toutes les chaînes. Pour réduire la complexité, on limite le nombre de chaînes que nous sélectionnons. Nous choisissons au plus les k meilleurs chaînes au sens du critère de cohérence. On suppose que l'opération de sélection ne dépend que du nombre de chaînes, bornée par $k \times n$ et de la taille de la plus grande chaîne m. Ainsi, le coût de sélection est borné par $O(S(k \times n, m))$. Dans le cas favorable où S est linéaire par rapport à ses deux arguments, le coût total de l'algorithme est de la forme $(|E| \times n \times m)$ où E est l'ensemble des couples de documents (C, D) tels que la chaîne CD satisfasse notre critère de cohérence.

3.4 Exemple de déroulement

La Tab. 1 présente le déroulement de l'algorithme sur trois nœuds sachant que les nœuds A, B et C ont déjà été traités. Les états avant et après le déroulement sont donnés en Fig. 3. Les flèches pleines indiquent les couples de documents qui sont valides au sens du critère de cohérence. Les flèches en pointillés indiquent un morceau de chaîne.

A, B et C ont déjà été traités. Les chaînes AB et AC ont été calculées.

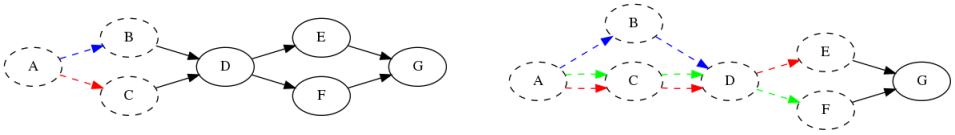


FIG. 3: Deux moments du déroulement de l'algorithme. Avant D à gauche. Après F et E à droite.

Nous traitons D. Seuls B et C sont des ancêtres valides pour D. Nous agglomérons les chaînes candidates à partir des chaînes finissant en B : B et AB ce qui nous donne BD et ABD. On fait de même pour C, tous les candidats sont alors D, CD, BD, ABD et ACD. À l'étape de sélection, toutes les chaînes sont conservées.

Nous traitons maintenant E. Seul D est un ancêtre valide de E. Nous agglomérons les chaînes candidates à partir des chaînes finissant en D ce qui nous donne tous les candidats : E, DE, BDE, CDE, $ABDE$ et $ACDE$. À l'étape de sélection, on ne conserve que DE, CDE et $ACDE$. L'étape en F est identique à celle en E.

TAB. 1: Déroulement de l'algorithme pour les nœuds D, E et F

Nœud X	Étape	$Candidates(X)$	$FinishIn(X)$
D	Traitement de B	D / ABD / BD	$\emptyset$
D	Traitement de C	D / CD / BD / ACD / ABD	$\emptyset$
D	Sélection	D / CD / BD / ACD / ABD	D / CD / BD / ACD / ABD
E	Traitement de D	E / DE / BDE / CDE / ABDE / ACDE	$\emptyset$
E	Sélection	E / DE / BDE / CDE / ABDE / ACDE	DE / CDE / ACDE
F	Traitement de D	F / DF / BDF / CDF / ABDF / ACDF	$\emptyset$
F	Sélection	F / DF / BDF / CDF / ABDF / ACDF	DF / CDF / ACDF

3.5 Critère de cohérence et attachement

Nous avons besoin dans notre démarche de déterminer si une chaîne est suffisamment cohérente pour être gardée. Nous définissons la mesure d'attachement d'un document à une chaîne comme une mesure de la vraisemblance de l'ajout dudit document à la fin de ladite chaîne. Cette mesure est pertinente pour notre algorithme où on se pose la question d'ajouter ou non un document à une chaîne déjà existante (et qui a donc déjà passé l'étape de sélection). Dans nos expériences, nous avons donné à l'attachement du document D à la chaîne ABC la forme suivante :

$$attachement(D, ABC) = F(sim(A, D), sim(B, D), sim(C, D))$$

où sim est une fonction de similarité sémantique entre documents. Nous avons expérimenté pour F plusieurs fonctions d'agrégation simples comme des moyennes, ou le minimum. Nous avons aussi essayé d'ajouter une pondération sur la distance de manière à tolérer des similarités plus faibles pour des documents éloignés dans la chaîne.

Nous définissons ainsi notre critère de cohérence à partir d'une fonction de similarité, d'une mesure d'attachement et d'un seuil d'admissibilité au-dessus duquel l'attachement est jugé cohérent. La stratégie de sélection que nous utilisons consiste à garder les k chaînes qui maximisent le critère de cohérence.

4 Expérimentations

La construction de trajectoire sans fixer la nature exacte de l'information véhiculée étant un problème neuf à notre connaissance, nous nous sommes tournés vers l'évaluation humaine de manière à construire des jeux de données annotés. L'objectif est double. D'une part valider que le problème est bien posé en analysant les résultats d'une évaluation humaine. D'autre part construire une vérité terrain de manière à pouvoir évaluer notre algorithme d'estimation de trajectoire. Nous commençons par présenter les jeux de données dont nous sommes partis puis nous détaillons notre protocole d'évaluation avant de discuter nos résultats.

4.1 Jeux de données

Nous avons pris deux jeux de données anglophones. Le premier est le Citation Network Dataset V1 d'AMINER [1] construit par Tang et al. (2008). Il est composé de résumés de papiers scientifiques extraits principalement des collections ACM et DBLP. Le jeu comprend 629 814 résumés de papiers. Notre second jeu de données correspond à l'ensemble des articles du Huffington Post version US sur la période du 1er juillet au 30 novembre 2016 pour un total de 49 648 articles.

Nous voulons évaluer des trajectoires. Cela entend d'évaluer toutes les chaînes qui les composent. Or, des corpus de cette taille contiennent beaucoup de chaînes. Aussi, nous avons créé deux jeux de données dérivés contenant moins de documents pour avoir un nombre de chaînes à évaluer raisonnable. Pour AMINER, nous avons choisi de sélectionner 150 résumés au hasard dans le corpus. Pour le Huffington Post, nous avons été plus précis. Nous avons filtré les articles contenant le mot "Trump" très représenté dans le jeu (> 11 000 documents). Ensuite, nous n'avons gardé que les articles entre 100 et 3000 signes. Le but est de ne pas perdre l'attention de l'évaluateur dans des articles trop longs à lire ou qui présentent trop peu de contexte. Nous avons choisi 150 documents au hasard parmi ceux répondant à ces critères. Nous nommons nos jeux de données ainsi échantillonnés AMINER et HuffPost respectivement.

4.2 Protocole d'évaluation

Pour évaluer notre approche, nous avons procédé en quatre étapes. Dans un premier temps, nous avons créé des trajectoires à partir de nos jeux de données. Nous avons réparti les chaînes parmi nos quatre évaluateurs de telle sorte que chaque chaîne soit évaluée exactement deux fois. L'étape suivante est l'évaluation elle-même. Enfin, nous étudions nos résultats et procédons à des comparaisons.

Nous avons créé nos trajectoires à partir d'une similarité cosinus sur les vecteurs TFIDF des documents. Le TFIDF est calculé sur l'ensemble des mots du document auquel nous ajoutons les n-grammes de tailles 2 à 4. Nous avons construit six trajectoires en faisant varier la mesure d'attachement d'une part (la moyenne arithmétique ou le minimum) et le seuil d'admissibilité d'autre part (parmi les valeurs 0,1 ou 0,2 ou 0,5). Nous avons réuni ces trajectoires pour chaque jeu de données. Cela nous donne deux ensembles de chaînes à évaluer.

La démarche de l'évaluateur est la suivante :

1. Le jeu AMINER est disponible ici : `https://aminer.org/citation`

1. D'abord, l'évaluateur doit prendre connaissance du contexte de la chaîne : le jeu de données dont elle provient et la nature des documents.
2. Ensuite, il lit le premier document de la chaîne (le plus ancien).
3. Puis, chacun des documents suivant lui est proposé en succession. À partir de là, il doit pour chacun déterminer si :
   (a) Il y a un lien sémantique fort ou faible avec le document précédent.
   (b) Il est fortement/faiblement/non plausible que de l'information se soit propagée du premier document jusqu'à celui-ci (évaluation de l'attachement).

4.3 Résultats de l'évaluation humaine

L'évaluation humaine a deux buts. Le premier est de vérifier que des humains peuvent se mettre d'accord sur la cohérence d'une chaîne de document, voire sur son intensité. Le second est de vérifier si notre approche construit de bons chemins.

Nous donnons dans la Tab. 2 le ratio d'accord des participants pour l'évaluation des liens directs et le ratio d'accord des participants pour l'évaluation de l'attachement pour les chaînes d'au moins trois documents. Les participants avaient trois choix dans les deux cas : présence d'un lien fort, d'un lien faible ou absence de lien. Nous séparons les résultats en deux. D'un côté, on décide que les participants sont d'accord s'ils ont fait exactement le même choix. De l'autre, on décide que les participants sont d'accord s'ils ont tous les deux choisi qu'il y a un lien (qu'il soit fort ou faible) ou tous les deux choisi qu'il n'y en a pas.

Pour les liens directs, les évaluateurs sont d'accord dans au moins 70 % des cas sur les deux jeux de données. Il y a un écart notable entre l'évaluation qui prend en compte l'intensité ou non sur AMINER. Pour l'attachement, les résultats sont moins bons sur AMINER quand on prend en compte l'intensité. Dans tous les autres cas, les évaluateurs sont d'accord dans au moins 80 % des cas pour l'attachement. Cela renforce l'intuition que l'évaluation est plus facile quand le contexte est plus riche. Ces deux résultats montrent que les humains arrivent à évaluer la cohérence des chaînes de documents avec consistance. Ceci nous conforte dans l'idée que le problème que nous traitons est bien posé.

TAB. 2: Accord inter-évaluateurs

Objet évalué	propriété	AMINER	HuffPost
Lien avec le document précédent	nombre d'évaluations	81	149
	Cohérence Fort/Faible/Non	68.09%	77.27%
	Cohérence Lien/Non	76.60%	79.55%
Attachement avec la chaîne (pour chaîne de taille > 2)	nombre d'évaluation	66	107
	Cohérence Fort/Faible/Non	57.89%	83.70%
	Cohérence Lien/Non	80.70%	85.87%

À présent, nous choisissons de répartir nos évaluations (que ce soit pour un lien direct ou un attachement) en cinq catégories selon l'accord des évaluateurs :

1. La majorité a jugé qu'il y avait un lien fort.
2. La majorité a jugé qu'il y avait un lien faible.
3. La majorité a jugé qu'il y avait un lien sans trancher son intensité.
4. La majorité a jugé qu'il n'y avait pas de lien.

5. Absence de majorité. Un vote pour la présence d'un lien, un vote contre.

La répartition des évaluations pour les liens directs et les attachements est donnée en Tab. 3. Nous remarquons que les résultats sont très bons pour AMINER avec seulement 17 % de non-liens directs et 9 % de non-attachement (catégorie 4). A contrario, les chaînes sur le HuffPost sont globalement mauvaises à la fois pour le lien direct (64 %) et pour les attachements (75 %). Pour comprendre ce résultat, nous devons nous rappeler comment a été créé l'ensemble de chaînes que nous évaluons. Il s'agit de l'union de plusieurs trajectoires, parmi lesquelles deux trajectoires calculées avec un seuil d'admissibilité de 0,1. Nous montrons dans la section suivante que les mauvaises chaînes proviennent sûrement de ces trajectoires. Cette différence entre AMINER et HuffPost pose le problème du choix du seuil d'admissibilité. Le seuil idéal dépend du jeu de données, ici un seuil de 0,1 n'est pas problématique pour AMINER tandis qu'il renvoie une grande quantité de mauvais liens pour HuffPost.

TAB. 3: répartitions des évaluations par catégorie (en pourcentage)

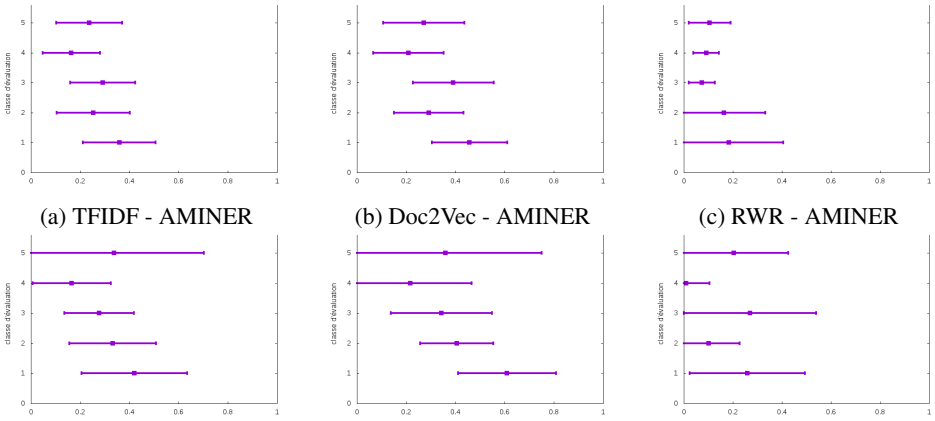
	Catégorie	1	2	3	4	5
Lien direct	AMINER	40.7	23.5	4.9	17.3	13.6
	HuffPost	18.8	10.7	1.3	63.8	5.4
Attachement	AMINER	34.8	19.7	19.7	9.1	16.7
	HuffPost	7.5	4.7	1.9	74.7	11.2

4.4 Comparaison de critères de cohérence

Puisque nous avons des chaînes annotées, nous diposons d'une vérité terrain qui nous permet de rechercher un critère de cohérence plus pertinent. Pour calculer nos chaînes, nous avons utilisé la similarité cosinus du TFIDF qui est une mesure standard de la similarité de textes. Nous proposons d'étudier qualitativement des fonctions d'attachement basées sur d'autres similarités. En plus du TFIDF, nous étudions : Une similarité basée sur Doc2Vec (Le et Mikolov, 2014) avec le produit cosinus. Nous la nommons Doc2Vec. Nous l'avons paramétrée avec un espace sémantique de taille 20. Enfin nous considérons une autre similarité, calculée par marche aléatoire avec retour, développée par Shahaf et Guestrin (2010) que nous nommons RWR. Nous l'avons paramétrée avec une probabilité de retour de 99 %.

Nous définissons une mesure d'attachement par moyenne arithmétique pour chacune de ces mesures de similarité. Pour chaque catégorie de chaînes annotées, nous calculons un intervalle centré sur la mesure d'attachement moyenne sur cette catégorie et d'amplitude deux fois l'écart-type de la mesure d'attachement. Nous estimons que la mesure d'attachement est bonne s'il y a une intersection faible ou inexistante entre l'intervalle des bonnes chaînes et l'intervalle des mauvaises chaînes. Les intervalles calculés sont présentés en Fig. 4.

Sur HuffPost, on remarque que l'ordre attendu pour le classement des catégories par attachement moyen est respecté pour les trois mesures. À savoir, dans l'ordre décroissant, la catégorie 1 (liens jugés forts), la catégorie 3 (liens jugés forts par un évaluateur, faibles par l'autre), la catégorie 2 (liens jugés faibles) et enfin la catégorie 4 (absence de lien). Les résultats sur le TFIDF confirment notre explication sur la proportion de mauvaises chaînes de HuffPost : elles ont en moyenne un attachement TFIDF en dessous de 0,2 ce qui suggère que les mauvaises chaînes proviennent de la trajectoire avec un seuil d'admissibilité de 0,1. Nous



(a) TFIDF - AMINER (b) Doc2Vec - AMINER (c) RWR - AMINER

(d) TFIDF - HuffPost (e) Doc2Vec - HuffPost (f) RWR - HuffPost

FIG. 4: Moyenne et écart-type des mesures d'attachement sur les chaînes par catégorie d'évaluation et jeu de données.

remarquons qualitativement que Doc2Vec semble être la mesure qui dissocie le mieux l'intervalle des chaînes fortement liées de celui des mauvaises chaînes.

Sur AMINER, on remarque que l'ordre attendu n'est pas toujours respecté mais que les chaînes fortement liées ont toujours un attachement moyen supérieur aux mauvaises chaînes. On note cependant qu'il y a très peu de mauvaises chaînes sur le jeu de donnée et que Doc2Vec semble être la mesure qui fait le mieux ressortir les chaînes fortement liées.

Ces mesures montrent qu'il est possible de capturer au moins en partie le jugement humain sur les chaînes avec des mesures bien connues. Si la section précédente montrait que la tâche est réalisable par l'humain, celle-ci renforce notre intuition que la tâche est également réalisable par une machine.

5 Conclusion

Calculer des approximations de la Trajectoire est un problème encore ouvert. Nous avons proposé un cadre formel pour le formuler. Ensuite, nous avons proposé un algorithme glouton qui calcule des chaînes de proche en proche. Dans le but de qualifier les chaînes que nous calculons, nous avons mené une campagne d'évaluation par l'humain. Pour ce faire, nous avons détaillé un protocole d'évaluation de chaînes que nous avons ensuite mis en pratique. Le bénéfice a été double : d'une part, nous avons vu que les évaluations humaines étaient consistantes entre elles, ce qui nous conforte dans l'idée que le problème est bien posé puisque la tâche est réalisable par l'humain. D'autre part, nous nous sommes servi de ces évaluations comme d'une vérité terrain pour tester différents critères de cohérence basés sur des mesures de similarités sémantiques. Nous avons vu que ces mesures réussissent à capturer les jugements humains sur la cohérence d'une chaîne de documents. Nous interprétons ce résultat comme une première preuve que la tâche est aussi réalisable de manière automatique.

Plusieurs axes d'améliorations sont envisagés. En particulier, Nous comptons former un critère de cohérence plus performant encore en explorant les similarités sémantiques prometteuses, par exemple Doc2Vec ou RWR. Pour cela, nous prévoyons une nouvelle campagne d'évaluation avec un nombre plus élevé de participants, ce qui aura aussi pour effet de consolider ou nuancer nos premiers résultats. Il faut aussi résoudre le problème du seuil d'admissibilité, un paramètre de notre critère de cohérence dont dépend la qualité des chaînes. Celui-ci dépend du jeu de données et est actuellement sélectionné empiriquement. Aussi, il serait intéressant de le déterminer automatiquement. Nous souhaitons également chercher de nouvelles façons de créer nos chaînes, par exemple en utilisant des méthodes probabilistes qui tireraient un ensemble de chaînes dont la cohérence serait élevée.

Une fois des trajectoires fiables calculées automatiquement, nous pouvons explorer leur utilisation dans plusieurs cas d'exploitation. Le but de la trajectoire est d'isoler les chaînes de propagation, aussi une volonté naturelle serait d'extraire la ou les informations qui se propagent le long de chaque chaîne. Ceci fait, nous pourrions étudier la manière dont les informations interagissent entre elles le long des chaînes. On peut aussi plonger les chaînes dans l'espace des auteurs afin d'étudier la manière dont ces derniers relaient l'information.

Enfin, nous nous posons la question du classement, de la synthèse et de la visualisation des chaînes elles-mêmes. Cela peut être la constitution d'un résumé de la propagation de l'information, une piste prometteuse en ce sens réside dans les travaux menés par Shahaf et al. (2013). Cela peut aussi être de trouver les informations communes entre les chaînes : c'est-à-dire de retrouver les informations principales du corpus. Ce sont les informations utilisées comme pré-requis dans les travaux connexes et dont nous nous passons pour calculer nos trajectoires.

Références

Farajtabar, M., M. Gomez-Rodriguez, M. Zamani, N. Du, H. Zha, et L. Song (2015). Back to the past : Source identification in diffusion networks from partially observed cascades. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*.

Gomez-Rodriguez, M., D. Balduzzi, et B. Schölkopf (2011). Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 561–568.

Kempe, D., J. Kleinberg, et É. Tardos (2003). Maximizing the Spread of Influence Through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, New York, NY, USA, pp. 137–146. ACM.

Le, Q. V. et T. Mikolov (2014). Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1188–1196.

Leskovec, J., L. Backstrom, et J. Kleinberg (2009). Meme-tracking and the Dynamics of the News Cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, New York, NY, USA, pp. 497–506. ACM.

Myers, S. A. et J. Leskovec (2012). Clash of the contagions : Cooperation and competition in information diffusion. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pp. 539–548.

Pinto, P. C., P. Thiran, et M. Vetterli (2012). Locating the source of diffusion in large-scale networks. *Physical review letters 109*(6), 068702.

Prakash, B. A., J. Vreeken, et C. Faloutsos (2012). Spotting culprits in epidemics : How many and which ones ? In *2012 IEEE 12th International Conference on Data Mining*, pp. 11–20. IEEE.

Shahaf, D. et C. Guestrin (2010). Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, New York, NY, USA, pp. 623–632. ACM.

Shahaf, D., C. Guestrin, et E. Horvitz (2013). "metro maps of information" by dafna shahaf, carlos guestrin and eric horvitz, with ching-man au yeung as coordinator. *SIGWEB Newsletter 2013*(Spring), 4 :1–4 :9.

Snowsill, T. M., N. Fyson, T. D. Bie, et N. Cristianini (2011). Refining causality : who copied from whom ? In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pp. 466–474.

Tang, J., J. Zhang, L. Yao, J. Li, L. Zhang, et Z. Su (2008). Arnetminer : Extraction and mining of academic social networks. In *KDD'08*, pp. 990–998.

Yang, S. et H. Zha (2013). Mixture of mutually exciting processes for viral diffusion. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pp. 1–9.

Zarezade, A., A. Khodadadi, M. Farajtabar, H. R. Rabiee, et H. Zha (2017). Correlated cascades : Compete or cooperate. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 238–244.

Zhao, Q., M. A. Erdogdu, H. Y. He, A. Rajaraman, et J. Leskovec (2015). SEISMIC : A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pp. 1513–1522.

Summary

Information spreads between people. On the Internet, information spreads particularly through textual documents. Many challenges arise from these spreads: identifying the targeted piece of information, tracking the changes over time, understanding the underlying mechanisms of those spreads, etc. Given a document among a huge corpus in which many pieces of information circulate, can we infer the paths information took in order to arrive to that document? We propose the notion of trajectory as the set of paths along which some information spread and we develop an algorithm for approximate it. We also propose a human evaluation protocol in order to annotate computed trajectories. We show that humans evaluations mostly match and that our algorithm finds good paths effectively.