Etat de l'art en compression multi-phrases pour la synthèse de documents

Kévin Espasa

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

État de l'art en compression multi-phrases pour la synthèse de documents

Kévin Espasa[1, 2]
(1) Syllabs, 35-37 rue Chanzy, 75011 Paris, France
(2) LS2N, 2 Chemin de la Houssinière, 44322 Nantes, France
espasa@syllabs.com, kevin.espasa@univ-nantes.fr

RÉSUMÉ

La compression multi-phrases est utilisée dans différentes tâches de résumé (microblogs, opinions, réunions ou articles de presse). Leur objectif est de proposer une reformulation compressée et grammaticalement correcte des phrases sources tout en gardant les faits principaux. Dans cet article, nous présentons l'état de l'art de la compression multi-phrases en mettant en avant les différents corpus et outils à disposition. Nous axons notre analyse principalement sur la qualité grammaticale et informative plus que sur le taux de compression.

ABSTRACT

State-of-the-art of multi-sentence compression for document summarization

Multi-sentence compression (MSC) is used in various summary tasks (microblog, opinion, meeting or news articles). The aim is to generate a grammatical and reduced compression from multiple source sentences while retains their main facts. In this article, we present the state of the art of MSC and the different corpora and tools available. We focus our analysis more on grammatical and informative quality than on compression rate.

MOTS-CLÉS : compression multi-phrases, état de l'art, jeu de données.

KEYWORDS: multi-sentence compression, state-of-the-art, datasets.

1 Introduction

La compression de phrases a pour objectif à partir d'une phrase en entrée d'en produire une nouvelle plus courte, grammaticalement correcte et tout aussi informative (Jing & McKeown, 2000). Principalement utilisées pour des tâches de résumé, ces méthodes peuvent se séparer en deux classes. Les méthodes par suppression (Filippova *et al.*, 2015; Wang *et al.*, 2017; Zhao *et al.*, 2018) cherchent à produire un résumé en supprimant les mots inutiles tandis que celles par abstraction (Choi *et al.*, 2019; Yu *et al.*, 2018) proposent une reformulation de la phrase en y ajoutant de nouveaux mots.

C'est à partir des travaux de Barzilay & McKeown (2005) que plusieurs phrases sont proposées en entrée d'un système de fusion de phrases. Le système doit permettre d'obtenir une phrase fluide et concise reflétant les faits communs à un ensemble de phrases partageant un même thème. Par la suite, Filippova & Strube (2008) proposent de ne plus se limiter aux faits communs mais d'utiliser la complémentarité des phrases partageant un même thème pour produire une phrase profitant de l'ensemble des faits. C'est à partir de Filippova (2010) que la tâche est nommée compression multi-

phrases. L'auteure considère que l'approche visant à produire une nouvelle phrase en gardant les faits importants présents dans un ensemble de phrases sources s'apparente plus à la tâche de compression de phrases qu'à une tâche de fusion de phrases.

source	Le cofondateur d'Apple nous a quitté mercredi 5 octobre à l'âge de 56 ans. Steve Jobs est mort mercredi, à la suite d'une longue maladie . Le fondateur d'Apple s'est éteint mercredi, à 56 ans, des suites d'un cancer du pancréas.
ref.	Steve Jobs, co-fondateur d'Apple, s'est éteint ce mercredi 5 octobre à 56 ans.
gen.	le co-fondateur d'apple steve jobs est mort le 5 octobre . apple steve jobs est mort le 5 octobre .

TABLE 1 – Exemple de phrases sources et de compressions issu de Boudin & Morin (2013)

La table 1 présente un exemple de phrases sources et d'une compression de référence (ref.) issu du corpus [1] Boudin & Morin (2013) ainsi que deux compressions générées automatiquement (gen.) par l'algorithme Takahe [2].

Dans cet article, nous présentons un état de l'art des méthodes de compression multi-phrases. Nous cherchons à évaluer les différentes méthodes, la facilité de reproductibilité et les différents corpus d'évaluation. Notre objectif par la suite est d'utiliser ces méthodes afin de produire une reformulation d'un ensemble de documents partageant un même thème dans un cadre industriel. Nous considérons ces approches comme pertinentes car elles permettraient de générer des textes plus ou moins compressés en fonction des besoins (de la brève à l'article détaillé).

Dans la suite de l'article, nous commençons par développer la problématique liée à la tâche en section 2. Puis nous discutons des différentes méthodes dans la section 3. La section 4 présente les différents jeux de données et les méthodes d'évaluation. La section 5 décrit les expérimentations que nous avons faites. Enfin nous concluons et présentons nos perspectives de recherches.

2 Compression multi-phrases

La compression multi-phrases cherche, à partir d'un regroupement de phrases similaires (i.e. partageant un même thème), à proposer une ou plusieurs reformulations respectant des contraintes d'information, de compression et de grammaticalité. La contrainte d'information a pour but de produire une reformulation la plus informative possible. Plus précisément, la méthode doit être capable d'identifier les faits les plus pertinents et de les restituer en sortie.

Considérons un ensemble de phrases S en entrée ayant en moyenne n termes. Afin de respecter la contrainte de compression la ou les phrases en sortie S' doivent avoir une moyenne de termes n' inférieure à n. Le respect de la grammaticalité est également important, les phrases générées doivent contenir le moins d'erreurs grammaticales.

La compression de phrases et la compression multi-phrases sont utilisées dans différentes tâches comme le résumé de microblogs (Sharifi *et al.*, 2010), d'opinions (Ganesan *et al.*, 2010), de réunions (Shang *et al.*, 2018) ou d'articles de presse (Nayeem *et al.*, 2018). Pour nos travaux, nous nous plaçons dans cette dernière tâche. Nous cherchons une méthode capable de résumer un ensemble

1. https://github.com/boudinfl/lina-msc/tree/master/src
2. https://github.com/boudinfl/takahe

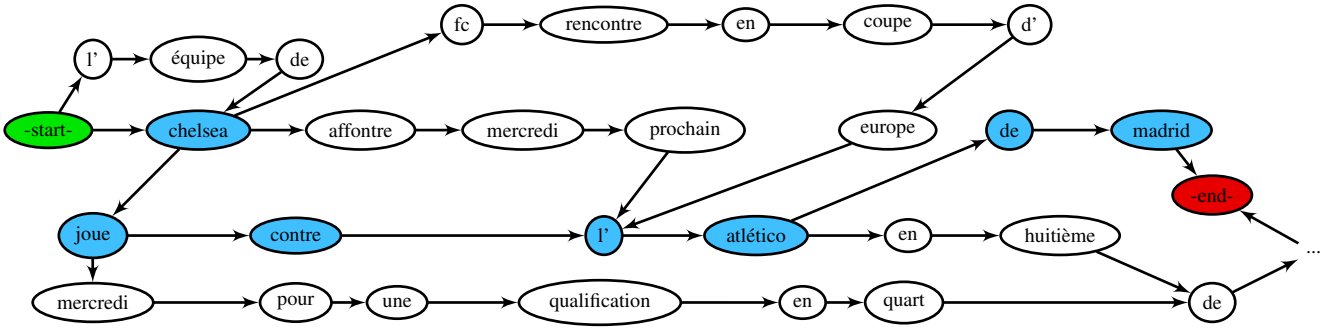

FIGURE 1 – Graphe représentant les phrases 1 à 4 et un chemin possible

de phrases pour produire une ou plusieurs phrases candidates. Étant dans un cadre industriel, nous mettons l'accent principalement sur la grammaticalité et l'informativité de la phrase. Il est également intéressant dans notre cadre de produire une reformulation des phrases sources. Plus précisément, nous souhaitons que les phrases générées possèdent le moins de mots en commun avec les sources.

3 Approches étudiées

La résolution de la tâche de compression multi-phrases utilise principalement une représentation des mots à l'aide d'un graphe (Filippova, 2010; Boudin & Morin, 2013; Linhares Pontes *et al.*, 2018; ShafieiBavani *et al.*, 2016). Les méthodes diffèrent cependant dans le regroupement entre les différents mots ainsi que lors du parcours du graphe.

3.1 Filippova (2010)

Filippova (2010) propose pour la compression multi-phrases une méthode basée sur la représentation des mots à l'aide de graphes. Un graphe $G = (N, A)$, avec N l'ensemble des nœuds et A l'ensemble des arêtes, est construit par l'ajout successif de mots des phrases d'un ensemble $S = s_1, ..., s_n$.

Les exemples de phrases suivantes servent à illustrer le fonctionnement de la méthode.

1. Chelsea joue contre l'Atlético en huitième de *finale de la Ligue des Champions*
2. Chelsea affontre mercredi prochain l'Atlético de Madrid
3. Chelsea FC rencontre en coupe d'Europe l'Atlético de Madrid
4. L'équipe de Chelsea joue mercredi pour une qualification en quart de *finale*

La Figure 1 représente le graphe construit en utilisant l'algorithme. En vert, le nœud de début et en rouge celui de fin de parcours. Les nœuds en bleu représentent un chemin possible lors du parcours. Afin d'améliorer la lisibilité du graphe, les parties en italique des phrases 1 et 4 ont été remplacées par des points de suspensions dans la Figure 1.

Chaque mot de la première phrase s_1 (la ponctuation étant exclue) est transformé en nœud n. Puis pour chaque phrase suivante, les mots sont ajoutés au graphe dans l'ordre suivant :

1. les mots grammaticaux n'ayant pas de nœuds candidats au regroupement ou pas d'ambiguïté possible sur le candidat,

2. les mot grammaticaux ayant plusieurs candidats au regroupement possible,
3. les mots vides.

Pour les mots du premier groupe, un mot est regroupé avec un nœud existant si ils sont similaires et ont la même étiquette morphosyntaxique et qu'aucun mot de la phrase s n'a déjà été regroupé avec le nœud n du graphe. Dans le cas où le mot ne peut être regroupé, un nouveau nœud est ajouté à G.

Pour les deux derniers groupes, en cas d'ambiguïté les mots suivant et précédent de chaque candidat sont comparés pour choisir le meilleur regroupement. Les mots vides sont regroupés seulement si leur contexte immédiat est similaire, sinon un nouveau nœud est créé. Le premier mot de chaque phrase est connecté avec un nœud de départ (-*start*- dans la Figure 1) tandis que le dernier est connecté avec un nœud de fin (-*end*- dans la Figure 1). Les nœuds sont reliés par des arêtes unidirectionnelles suivant leur ordre dans la phrase et un poids par défaut de 1 est ajouté aux arêtes.

Une fois le graphe obtenu, le poids de chaque arête est calculé en utilisant l'équation 1. $freq(i)$ et $freq(j)$ représentent respectivement la fréquence du mot i et du mot j. La fonction de cohésion (équation 2) calcule pour chaque i et j leur fréquence divisée par la distance entre les mots dans chaque phrase. Le but étant de privilégier les mots qui apparaissent le plus souvent ensemble.

Par la suite un algorithme de K-plus court chemin est utilisé pour parcourir le graphe. Le parcours vise à trouver un chemin de taille définie (8 dans l'article) tout en minimisant la somme des arêtes parcourues. Afin d'obtenir une phrase grammaticalement correcte en sortie, un nœud contenant un verbe doit être traversé. Les scores sont finalement normalisés en fonction de la taille de phrase générée puis réordonnés. Le chemin ayant le plus petit poids est alors la meilleure compression.

$$w(i,j) = \frac{cohesion(i,j)}{freq(i) \times freq(j)} \tag{1}$$

$$cohesion(i,j) = \frac{freq(i) + freq(j)}{\sum_{s \epsilon S} distance(s,i,j)^{-1}} \tag{2}$$

La méthode offre l'avantage de n'être dépendante que d'un outil d'étiquetage morphosyntaxique et une liste de mots outils. Pour l'anglais, l'auteure génère une liste de 600 mots vides spécifiques aux articles d'actualités pour l'anglais incluant certains verbes (*said*, *seems*) ainsi qu'une liste publique de 180 mots vides [3] pour l'espagnol. La liste générée pour l'anglais n'est cependant pas mise à disposition et la façon dont elle est créée n'est pas décrite pour reproduire les expériences.

3.2 Boudin & Morin (2013)

Boudin & Morin (2013) proposent une amélioration de la méthode de Filippova (2010) en y incluant la ponctuation ainsi qu'en utilisant une méthode d'extraction de termes clefs pour le calcul des poids. Les auteurs reprennent les trois étapes successives d'ajout des mots et en ajoutent une quatrième pour la ponctuation. En cas d'ambiguïté lors de l'ajout, le contexte immédiat (mot suivant et mot précédent) sont comparés.

Dans les résultats donnés par Filippova, l'information est restituée en totalité dans 52 % des cas en anglais (40 % en espagnol). Afin d'améliorer la conservation de l'information, une méthode pour réordonner les phrases générées en fonction des termes clefs qu'elles contiennent est mise en place.

3. https://www.ranks.nl/stopwords/spanish

La méthode se déroule en deux étapes. Tout d'abord, un graphe pondéré est construit pour chaque regroupement de phrases. Chaque nœud contient le mot et son étiquette morphosyntaxique. Les arêtes sont pondérées en utilisant la cooccurrence entre les mots présents dans des nœuds. Un poids d'importance du nœud est calculé (équation 3) en utilisant la méthode TextRank (Mihalcea & Tarau, 2004).

$$TextRank(A_i) = (1-d) + d \times \sum_{V_j \epsilon adj(V_i)} \frac{w_{ji}}{\sum_{V_k \epsilon adj(V_j)} w_{kj}} S(A_j) \tag{3}$$

Le score du nœud A_i est calculé à l'aide de l'équation 1, de A_j qui représente les nœuds en lien direct avec V_i et le facteur d défini à 0,85. La seconde étape consiste à extraire, pour chaque phrase générée, les expressions clefs ayant la forme suivante : $(ADJ) * (NPP|NC) + (ADJ)*$. Une fois extrait, le score d'une expression clef k est calculée avec l'équation 4. Puis le score général de la phrase est défini en utilisant la somme des scores des poids du chemin pour construire la phrase c divisée par sa longueur et par la somme des scores des expressions clefs (l'équation 5).

$$score(k) = \frac{\sum_{w \epsilon k} TexTrank(w)}{|k| + 1} \tag{4}$$

$$score(c) = \frac{\sum_{i,j \epsilon path(c)} w_{i,j}}{|c| + \sum_{k \epsilon c} score(k)} \tag{5}$$

D'après les résultats présents dans Boudin & Morin (2013), cette méthode permet une restitution totale de l'information de 62,5 % des cas contre 43,3 % pour Filippova (2010) sur un corpus français. À noter, qu'une différence de 8,7 % est présente dans la restitution totale de l'information entre l'approche de Filippova (2010) sur l'anglais et la reproduction de la méthode sur le français par Boudin & Morin (2013). Outre la langue, il est possible comme pour la comparaison entre l'espagnol et l'anglais que la taille de la liste des mots vides, 203 mots pour le français contre 600 pour Filippova (2010) avec l'anglais, ait un impact sur les performances. Notons également que l'amélioration de l'information entraîne une diminution 7,5 % de la grammaticalité des phrases générées.

Les auteurs ont mis en ligne le code [4] de leur algorithme ainsi qu'une implémentation de celui de Filippova (2010). Les listes de mots vides utilisées sont également disponibles.

3.3 ShafieiBavani *et al.* (2016)

En reprenant les travaux de Filippova (2010) et Boudin & Morin (2013), ShafieiBavani *et al.* (2016) proposent une méthode utilisant la détection d'expressions polylexicales et le remplacement de synonymes afin d'améliorer la grammaticalité et l'information dans les phrases générées. La méthode utilise l'approche de Filippova (2010) pour la construction d'un graphe à partir d'un ensemble de phrases. Deux modifications sont apportées afin de prendre en compte les expressions polylexicales et les synonymes.

La première consiste à détecter les expressions polylexicales avec l'outil jMWE (Kulkarni & Finlayson, 2011) puis à les remplacer par un synonyme composé d'un mot à l'aide de Wordnet. Par exemple

4. https://github.com/boudinfl/takahe

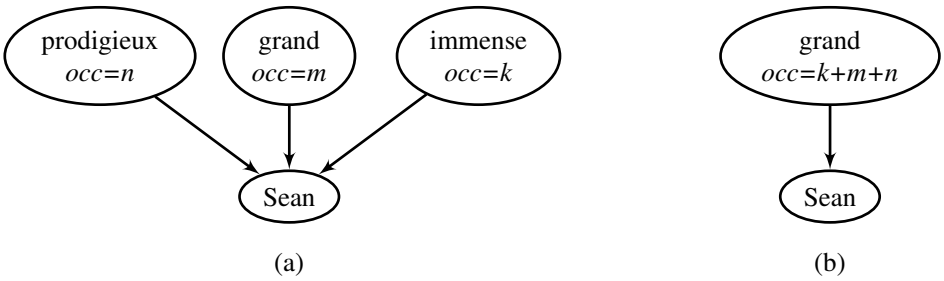

FIGURE 2 – Graphe représentant le regroupement de synonymes

l'expression *passer l'arme à gauche* dans *Sean Connery vient de passer l'arme à gauche* sera détectée comme une expression polylexicale puis convertie en *mourir* lors de son ajout dans le graphe.

La seconde modification consiste à regrouper les synonymes dans un même nœud afin de : (i) réduire l'ambiguïté lors du regroupement entre un mot et un nœud candidat et (ii) limiter le nombre de chemins et augmenter le score de cooccurrences entre un mot et un ensemble de mots synonymes. Pour les phrases suivantes, lors de la construction du graphe de la Figure 2 *grand*, *immense* et *prodigieux* seront regroupés dans un même nœud. Dans notre exemple, k, n et m ont un score de un (a), le score de cooccurrences total sera alors de trois pour le nœud *immense* (b) :

1. Le grand Sean Connery est mort
2. L'immense Sean Connery est mort
3. Le prodigieux Sean Connery est mort

Une fois le graphe pondéré créé, un parcours à l'aide de l'algorithme des K-plus courts chemins est effectué. Puis un algorithme pour réordonner la liste de candidats est utilisé. Les auteurs utilisent en plus de la méthode d'équation de Boudin & Morin (2013), un modèle de langue créé à partir d'étiquettes morphosyntaxiques afin d'augmenter la validité syntaxique de la phrase (équation 6). Le score final d'une phrase c est représenté par un facteur μ, le score calculé par l'équation 4 et le score du modèle de langue (équation 7).

$$score_{LM}(c) = 10^{\frac{logprob(c)}{|c|}} \tag{6}$$

$$score_{final}(c) = \mu \times score(c) + (1 - \mu) \times score_{LM}(c) \tag{7}$$

La méthode permet grâce à la création d'un graphe ayant moins de chemins possibles une amélioration de la compression et de la grammaticalité. Cependant, le remplacement des expressions polylexicales et de certains termes par leur synonyme limite la variabilité des phrases générées.

3.4 Linhares Pontes *et al.* (2020)

Linhares Pontes *et al.* (2020), eux, utilisent un graphe de nœuds étiquetés pour la compression de phrases combiné à un modèle d'optimisation linéaire en nombre entiers (OLNE). Un graphe $G = (N, A)$ est construit en utilisant la méthode d'ajout successif de mots de Filippova (2010). Chaque nœud N se voit associé d'un label compris dans $K = 0, ..., |K|$. L'objectif est de parcourir le graphe en passant dans le plus de nœud étiqueté tout en ne les traversant pas plus d'une fois. Dans le

contexte de l'article, le label est associé à un mot clef, si le nœud ne contient pas de mot clef le label sera zéro.

Trois méthodes sont mises en place pour la détection des mots clefs : une indexation sémantique latente (LSI), une allocation de Dirichlet latente (LDA) et l'algorithme TextRank. Des expériences sont faites en sélectionnant les 5 ou 10 mots clefs les plus pertinents de chaque méthode. Les mots clefs extraits à l'aide de la méthode LDA sont les plus présents dans les corpus français, portugais et espagnol de référence. Le LDA utilisant un seuil de 10 est sélectionné.

OLNE permet de définir une fonction d'optimisation ainsi que des contraintes lors du parcours du graphe. L'objectif est définie par l'équation 8 avec $x_{i,j}$ qui représente l'existence d'un arc entre les nœuds i,j, $w_{i,j}$ est l'équation 1 et b_k indique la présence du mot clef k dans la solution.

$$score_{opt}(s) = Minimize(\sum_{(i,j)\epsilon A} w_{i,j} \cdot x_{i,j} - c \cdot \sum_{k\epsilon K} b_k) \tag{8}$$

La liste de contraintes comporte : un minimum et un maximum dans la longueur de la phrase générée, la phrase doit contenir des mots clefs et le chemin ne doit pas traverser plusieurs fois le même nœud. Afin de prendre en compte la taille des phrases générées, le score est normalisé (équation 9).

$$score_{norm}(s) = \frac{e^{score_{opt}(s)}}{|c|} \tag{9}$$

3.5 Zhao *et al.* (2019)

Enfin, Zhao *et al.* (2019) proposent une méthode basée sur un bi-LSTM pour réécrire les compressions générées. Un corpus en phrases similaires est construit (voir section 4.1). La construction permet d'obtenir un corpus A de 140 000 regroupements de phrases composés chacun de 2 à 4 phrases.

La méthode de création du modèle se divise en 3 étapes. La première consiste à partir du corpus A de créer à l'aide de la méthode de Boudin & Morin (2013) un corpus de compression nommé B. Les expressions polylexicales, les verbes, les adjectifs et les noms compris dans chaque phrase du corpus B sont remplacés par leur plus petit synonyme possible à l'aide de Wordnet [5] et de PPDB 2.0 [6]. Le but étant d'obtenir un troisième corpus nommé C avec une compression maximale.

Lors de la seconde étape, un modèle bi-LSTM encodeur décodeur est entraîné avec en entrée le C et en sortie B. Ce dernier permet de générer à partir d'un corpus C' contenant 1 millions de phrases (le nombre de tokens moyen de C' équivaut à celui de C) un corpus B'. La dernière étape consiste à entraîner un modèle bi-LSTM avec en entrée les corpus B et B' et en sortie les corpus C et C'. L'objectif de ce modèle est de reformuler les compressions générées par d'autres méthodes en améliorant la grammaticalité et en y ajoutant de nouveaux mots (grâce à l'étape 2).

L'approche de Zhao *et al.* (2019) a pour but de produire une réécriture plus compressée et introduisant des mots non présents dans la compression d'origine. La réécriture reste cependant dépendante de l'approche de compression utilisée, la propagation d'erreurs informatives n'est pas à négliger.

5. https://wordnet.princeton.edu
6. http://paraphrase.org

3.6 Synthèse

Les méthodes utilisant des graphes proposent une compression des faits présents dans plusieurs phrases. Les différentes façons dont les scores sont calculés présentent l'avantage de ne pas pénaliser un regroupement de phrases légèrement bruité. En effet, l'information restituée en première sera celle présente le plus souvent. Ce qui est intéressant notamment lorsque le regroupement de phrases se fait de manière automatique. Nous nous plaçons dans une problématique de synthèse d'articles de presse, les méthodes de Filippova (2010); Boudin & Morin (2013); Linhares Pontes *et al.* (2018) ont le désavantage de ne pas ajouter de variantes lexicales aux phrases en sortie du système. Les mots en sortie correspondent obligatoirement aux entrées. Zhao *et al.* (2019); ShafieiBavani *et al.* (2016) tentent de proposer des méthodes pour remplacer les mots ou regrouper les synonymes. ShafieiBavani *et al.* (2016) s'appuient sur un outil d'extraction de mots polylexicaux seulement disponible en anglais et Zhao *et al.* (2019) ne précisent pas la façon dont les expressions polylexicales sont détectées. Le remplacement des expressions polylexicales n'est cependant pas sans risque, l'ambiguïté existe et n'est pas facilement détectable.

4 Jeux de données et mesures d'évaluation

Dans les travaux cités précédemment, les langues utilisées sont l'anglais, l'espagnol, le français, et le portugais. Des jeux d'évaluation ont été mis en ligne pour l'espagnol, le français et le portugais, mais il n'existe aucun jeu de référence, à notre connaissance, pour l'anglais. Notons quand même qu'une procédure standard de création des corpus d'évaluation existe. Nous décrirons ensuite les différentes mesures d'évaluation mises en œuvre dans le cadre de la compression multi-phrases.

4.1 Méthodologie de création de jeux de données

La méthode de Filippova (2010) consiste à collecter des regroupements d'articles de presse traitant d'un même événement à l'aide de Google News [7]. Ce dernier présente l'avantage d'avoir une classification et un regroupement d'articles à disposition. Les regroupements, manuellement extraits, contiennent plusieurs articles, entre 10 et 30 pour Filippova (2010) et au moins 20 pour Boudin & Morin (2013). La première phrase de chaque article est conservée (sauf en cas de duplicata où la phrase est retirée). Elle est considérée comme étant un bon résumé de l'article et est utilisée en référence dans la tâche de résumé (Dang, 2005).

Boudin & Morin (2013) ajoutent une compression manuelle pour le jeu de référence. La compression de référence consiste à demander à des locuteurs natifs de produire, à l'aide des phrases sources, la meilleure compression possible en utilisant le moins de nouveaux mots possibles.

Zhao *et al.* (2019) proposent une méthode de construction automatique d'un corpus de phrases similaires. Les auteurs appliquent une méthode de similarité des bigrammes sur le corpus English Gigaword. Une limite basse et une limite haute sont ajoutées afin d'éviter un regroupement de phrases pas assez ou trop similaires. Une évaluation humaine sur cinquante regroupements de phrases donne un résultat de 90 % de regroupement correct (Zhao *et al.*, 2019). Le corpus ainsi obtenu contient 140 572 groupements de phrases contenant entre 2 et 4 phrases chacun. Les auteurs créent un corpus

7. https://news.google.com

de référence en sélectionnant aléatoirement 150 phrases et en demandant à deux locuteurs natifs d'en produire une compression.

Article	Langue	Corpus	Disponible	Licence
(Filippova, 2010)	anglais	Google News	non	
	espagnol	Google News	non	
(Boudin & Morin, 2013)	français	Google News	oui [8]	MIT
(ShafieiBavani *et al.*, 2016)	anglais	Google News	non	
(Linhares Pontes *et al.*, 2020)	espagnol	Google News	oui [9]	GPL
	portugais	Google News	oui [10]	GPL
(Zhao *et al.*, 2019)	anglais	English Gigaword	oui [11]	LDC
	anglais	Fusion Corpus		

TABLE 2 – Caractéristiques des données utilisées dans les différents articles cités

La table 2 présente les caractéristiques des données utilisées dans les articles. Majoritairement, les auteurs utilisent Google News pour créer leur jeu de données mais seulement trois d'entre eux sont disponibles gratuitement. Notons qu'aucun corpus de référence n'existe pour l'anglais à ce jour et que le Fusion Corpus (McKeown *et al.*, 2010) utilisé comme second corpus d'évaluation par Zhao *et al.* (2019) n'est plus disponible à l'adresse indiqué dans leur article.

4.2 Description des corpus disponibles

Nous décrirons dans cette partie les corpus disponibles pour le français, le portugais et l'espagnol. Boudin & Morin (2013) ont mis à disposition un corpus libre en français contenant 618 phrases ainsi que les 120 phrases de références associées aux 40 clusters (3 phrases de références par cluster). Les longueurs moyennes en nombre de tokens pour la source et la référence sont respectivement de 32,7 et de 19,7. Ce qui implique un taux de compression manuel de 60 %.

Des corpus en espagnol et en portugais ont été créés par Linhares Pontes *et al.* (2020). Ces derniers sont également libres. Le corpus portugais contient 40 clusters composés de 544 phrases sources et 80 phrases de références. Le corpus espagnol se compose 800 phrases réparties dans 40 clusters et 4 phrases de références par cluster. Le taux de compression entre la source et la référence est en moyenne de 54 % pour le portugais et 61 % pour l'espagnol.

La table 3 récapitule les différentes informations sur les corpus : nombre de phrases, nombre minimum et maximum de phrases par cluster, nombre minimum, maximum et moyen de tokens par phrases ainsi que le taux de compression entre les phrases de références et les phrases sources.

4.3 Mesures d'évaluation

Plusieurs types de mesures existent pour évaluer la qualité d'une compression multi-phrases, ces mesures peuvent être séparées en deux familles : les automatiques et les manuelles. Les méthodes

8. https://github.com/boudinfl/lina-msc
9. http://juanmanuel.torres.free.fr/corpus/msf2/publications.html
10. http://juanmanuel.torres.free.fr/corpus/msf2/publications.html
11. https://catalog.ldc.upenn.edu/LDC2011T07

	(Boudin & Morin, 2013)		(Linhares Pontes *et al.*, 2020)			
Langue	Français		Portugais		Espagnol	
#clusters	40		40		40	
Type	**Source**	**Référence**	**Source**	**Référence**	**Source**	**Référence**
#phrases	618	120	544	80	800	160
min. phrases	7	3	9	2	20	4
max. phrases	36	3	22	2	20	4
#tokens	20 225	2 362	17 998	1 426	30 589	3 695
min. tokens	10		11	10	16	16
max. tokens	82		77	26	100	35
moy. tokens	32,7	19,7	33,1	17,8	38,2	23,1
taux de compression		60 %		54 %		61 %

TABLE 3 – Caractéristiques des corpus

automatiques d'évaluation sont devenues courantes dans les tâches d'évaluation des textes générés que ce soit pour la traduction automatique ou le résumé. Dans le cas de la compression multi-phrases, différentes mesures sont utilisées : BLEU, ROUGE ou encore METEOR afin de comparer la similarité entre les phrases de références et les phrases générées. Cependant, ces méthodes ne permettent pas de comparer la qualité grammaticale et la quantité d'informations pertinentes restituées.

Les méthodes manuelles pour la compression multi-phrases cherchent à évaluer ces aspects (cf. table 4. Barzilay & McKeown (2005) proposent une méthode de notation de la qualité grammaticale des phrases générées : *parfait* si la phrase est grammaticalement correcte (2 points), *presque* si la phrase ne requiert qu'une correction minime : une seule erreur (1 point) et *agrammaticale* si la phrase est incorrecte (0 point). Filippova (2010) reprend ce système pour noter cette fois la qualité de l'information restituée : *n/a* si le regroupement de phrases est trop bruité et ne peut donc pas produire de synthèse, *parfait* si la phrase contient les informations du thème principal (2 points), *en relation* si phrase contient une partie des informations du thème (1 point) et *sans relation* si la phrase n'a pas de rapport avec le thème.

Caractéristique	**Description**	**Point(s)**		
		2	1	0
Grammaticalité	grammaticalité parfaite	×		
	correction minime		×	
	agrammaticale			×
Information	parfaitement restituée	×		
	quasiment restituée		×	
	non restituée			×

TABLE 4 – Notation de l'information et de la grammaticalité

Le taux de compression est également évalué. Cela consiste à diviser le nombre de tokens de la phrase générée par le nombre de tokens moyens des phrases en entrée du système.

La table 5 récapitule les méthodes utilisées dans les différents articles cités. Les scores de grammaticalité, d'information et le taux de compression sont présents à chaque fois, ce qui montre leur importance dans la compression de phrase. Nous pouvons noter que Filippova (2010) n'utilise pas

d'évaluation automatique. Cela s'explique par le fait que son approche est évaluée sur une sous-partie de son corpus source et donc qu'il n'y a pas de corpus de référence construit manuellement.

Articles	Gram.	Inf.	Taux comp.	BLEU	ROUGE	METEOR
(Filippova, 2010)	×	×	×			
(Boudin & Morin, 2013)	×	×	×	×	×	
(ShafieiBavani *et al.*, 2016)	×	×	×	×	×	
(Linhares Pontes *et al.*, 2018)	×	×	×	×		
(Zhao *et al.*, 2019)	×	×	×			×

TABLE 5 – La liste des métriques utilisées dans les différents articles

5 Expérimentation des méthodes

Dans cette partie, nous décrivons les expériences que nous avons réalisées. Notre objectif pour commencer était de regarder le résultat des systèmes de compression multi-phrases sur des titres d'articles de presse. Le choix de se limiter aux titres se justifie par différentes raisons : dans le cadre de nos travaux sur la synthèse de documents, il est intéressant pour nous de pouvoir reformuler un titre. L'une des complexités des méthodes de compression multi-phrases est l'alignement automatique des phrases de différents documents regroupés entre eux par similarité sémantique. L'utilisation des titres permet, dans un premier temps, de tester les approches en utilisant un corpus de phrases peu bruité. Enfin, les titres présents dans un regroupement de documents ont pour avantage d'être relativement court et de traiter du même thème.

Nous avons expérimenté les approches de Filippova (2010) et Boudin & Morin (2013) en utilisant le code mis à disposition [12] par Boudin & Morin (2013). Les deux approches sont implémentées, même s'il est à noter que la ponctuation est présente pour le système de Filippova (2010). Le code est fonctionnel en l'état mais des modifications ont été apportées pour l'intégrer dans notre processus automatique de traitement [13]. La compression sur des titres donne de bons résultats que ce soit sur des regroupements avec peu de phrases (4 ou 5 phrases) ou sur des phrases bruitées (nom du média présent dans le titre par exemple). Nous pouvons mettre en avant deux problématiques rencontrées.

La première est la gestion des entités nommées dans la restitution de l'information. Les méthodes, actuellement, ne prennent pas en compte les entités nommées dans la création du graphe, ce qui peut produire un manque d'information ou une confusion lors de la restitution. Le premier cas apparaît notamment avec les entités nommées de type fonction. Par exemple :

— Le président de Nikola a démissionné
— Nikola : le président a démissionné

Le système produira *le président a démissionné*. Cette dernière phrase est correcte mais en l'état le titre n'est pas exploitable si nous souhaitons privilégier l'informativité à la compression. Il sera ici intéressant de regrouper dans le même nœud du graphe les mots : *le président de Nikola*.

Un autre cas apparaît lorsque deux entités nommées dans une même phrase comportent un mot en commun. Par exemple avec *le groupe Renault* et *la Renault Clio*, il est important de dissocier les deux

12. https://github.com/boudinfl/takahe
13. Nous avons mis à jour le code pour qu'il puisse fonctionner en Python3 et modifié les patrons d'extractions pour qu'il fonctionne avec les étiquettes morpho-syntaxiques produites par un modèle https://spacy.io/

mots *Renault* afin de ne pas avoir en sortie : le *Groupe Renault Clio*.

La seconde problématique, plus complexe, est la capacité de l'approche à restituer une information lorsque le sujet et l'objet sont réversibles. Prenons l'exemple d'un match nul entre deux équipes de football :

— Le PSG version Mauricio Pochettino débute par un nul à Saint-Etienne
— L1 : l' ASSE décroche le nul 1 - 1 face au PSG

L'une des propositions des systèmes, de part l'utilisation des chemins du graphe, produira comme phrase : *l' asse décroche le nul à saint-etienne*. La phrase est grammaticalement correcte mais la production d'un titre comme ce dernier perd en information.

Nous avons essayé de reproduire l'expérience de Zhao *et al.* (2018) en utilisant le code qu'il a mis à disposition [14]. Le code peut se diviser en deux grands objectifs : le premier, la création du corpus parallèle, et en deuxième, la méthode de création du modèle. Le code mis à disposition n'est pas fonctionnel en l'état mais il est facilement révisable. Pour la création du modèle, les étapes 1 et 2 de son approche ne sont pas présentes. Pour l'étape 1, la création du corpus B avec l'outil de Boudin & Morin (2013) est facile à mettre en place. Il est plus compliqué de passer du corpus B au corpus C, les auteurs ne précisant pas dans l'article la façon dont les expressions polylexicales sont extraites et rien dans le code ne le fait. Pour l'étape 2, les auteurs génèrent un million de phrases compressées B' à partir d'un million de phrases C'. Nous regretterons que le modèle ne soit pas disponible ni que l'origine des phrases C' soit donnée. Pour l'étape 3, le code est disponible mais ne fonctionne pas et le modèle entraîné n'est pas disponible. Nous avons tenté de le corriger mais le manque d'informations sur certaines parties comme ce qui sert à l'entraînement du Word2Vec a rendu la tâche impossible.

6 Conclusion et travaux futurs

Dans cet article, nous avons présenté les principales méthodes de compressions de phrases, les corpus utilisés et les différentes méthodes d'évaluation. Pour nos besoins, nous avons mis l'accent sur la qualité grammaticale et de l'information restituée. Le taux de compression est secondaire par rapport aux deux autres caractéristiques.

Les articles reposent tous sur l'utilisation d'un graphe pour représenter les phrases et pour les parcourir. Les différentes améliorations apportées depuis les travaux de Filippova (2010) concernent la façon dont mettre en avant certains termes et la manière d'associer un score à une phrase générée. Il est également intéressant de souligner que les méthodes n'ont besoin que de peu d'apport extérieur pour fonctionner correctement. Une liste de mots vides et un outil d'étiquetage morpho-syntaxique sont souvent suffisants.

Les expérimentations nous ont montré que nos objectifs étaient légèrement différents de ce qui se fait dans le domaine de la compression multi-phrases. Nous cherchons à obtenir avant tout une phrase grammaticalement correcte et informative. Nous voulons également être capable d'ajouter dans la phrase générée des mots non présents dans phrases sources.

Nos travaux futurs s'intéresseront à une meilleure prise en compte des entités nommées dans les phrases afin d'obtenir une restitution plus informative tout en ne délaissant pas la qualité grammaticale de la phrase générée. Nous chercherons également à développer une méthode ajoutant de la variété lexicale en prenant en compte les difficultés sur l'ambiguïté de certains mots.

14. https://github.com/code4ai

Références

BARZILAY R. & MCKEOWN K. R. (2005). Sentence fusion for multidocument news summarization. *Computational Linguistics*, **31**(3), 297–328.

BOUDIN F. & MORIN E. (2013). Keyphrase extraction for n-best reranking in multi-sentence compression. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'2013)*, p. 298–305, Atlanta, GA, USA.

CHOI S. J., JUNG I., PARK S. & PARK S.-B. (2019). Abstractive sentence compression with event attention. *Applied Sciences*, **9**(19).

DANG H. T. (2005). Overview of duc 2005. In *Proceedings of the Document Understanding Conference (DUC'2005)*, p. 1–12, Vancouver, Canada.

FILIPPOVA K. (2010). Multi-sentence compression : Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, p. 322–330, Beijing, China.

FILIPPOVA K., ALFONSECA E., COLMENARES C. A., KAISER L. & VINYALS O. (2015). Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'2015)*, p. 360–368, Lisbon, Portugal.

FILIPPOVA K. & STRUBE M. (2008). Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP'2008)*, p. 177–185, Honolulu, HI, USA.

GANESAN K., ZHAI C. & HAN J. (2010). Opinosis : A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'2010)*, p. 340–348, Beijing, China.

JING H. & MCKEOWN K. R. (2000). Cut and paste based text summarization. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference (NAACL'2000)*, p. 178–185, Seattle, WA, USA.

KULKARNI N. & FINLAYSON M. (2011). jMWE : A Java toolkit for detecting multi-word expressions. In *Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World (MWE'2011)*, p. 122–124, Portland, OR, USA.

LINHARES PONTES E., HUET S., TORRES-MORENO J.-M., GOUVEIA T. & LINHARES A. (2020). A multilingual study of multi-sentence compression using word vertex-labeled graphs and integer linear programming. *Computación y Sistemas*, **24**.

LINHARES PONTES E., TORRES-MORENO J.-M., HUET S. & LINHARES A. C. (2018). A new annotated Portuguese/Spanish corpus for the multi-sentence compression task. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC'2018)*, p. 3192–3196, Miyazaki, Japan.

MCKEOWN K., ROSENTHAL S., THADANI K. & MOORE C. (2010). Time-efficient creation of an accurate sentence fusion corpus. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL'2010)*, p. 317–320, Los Angeles, CA, USA.

MIHALCEA R. & TARAU P. (2004). TextRank : Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP'2004)*, p. 404–411, Barcelona, Spain.

NAYEEM M. T., FUAD T. A. & CHALI Y. (2018). Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING'2018)*, p. 1191–1204, Santa Fe, NM, USA.

SHAFIEIBAVANI E., EBRAHIMI M., WONG R. K. & CHEN F. (2016). An efficient approach for multi-sentence compression. In R. J. DURRANT & K.-E. KIM, Éds., *Proceedings of The 8th Asian Conference on Machine Learning (ACML'2016)*, p. 414–429, Hamilton, New Zealand.

SHANG G., DING W., ZHANG Z., TIXIER A., MELADIANOS P., VAZIRGIANNIS M. & LORRÉ J.-P. (2018). Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'2018)*, p. 664–674, Melbourne, Australia.

SHARIFI B., HUTTON M.-A. & KALITA J. (2010). Summarizing microblogs automatically. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL'2010)*, p. 685–688, Los Angeles, CA, USA.

WANG L., JIANG J., CHIEU H. L., ONG C. H., SONG D. & LIAO L. (2017). Can syntax help ? improving an LSTM-based sentence compression model for new domains. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'2017)*, p. 1385–1393, Vancouver, Canada.

YU N., ZHANG J., HUANG M. & ZHU X. (2018). An operation network for abstractive sentence compression. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING'2018)*, p. 1065–1076, Santa Fe, NM, USA.

ZHAO Y., LUO Z. & AIZAWA A. (2018). A language model based evaluator for sentence compression. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'2018)*, p. 170–175, Melbourne, Australia.

ZHAO Y., SHEN X., BI W. & AIZAWA A. (2019). Unsupervised rewriter for multi-sentence compression. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'2019)*, p. 2235–2240, Florence, Italy.