Synthèse résumé automatique

Introduction

L'objet d'étude de cette synthèse est le résumé automatique. Ce sujet n'est pas récent car plusieurs articles sur le sujet étaient déjà publiés dans les années 50.

Le principe général du résumé automatique est le suivant : Partant d'un document, on veut compresser l'information de ce document avec le moins de perte possible. On veut conserver le sens général du texte, mais éviter au maximum les redondances (entre autre, nous verrons différents critères) Il est important de noter que l'on ne parle pas ici de faire des plans d'articles de façon automatique ou même des synopsis, juste de compresser l'information contenu dans un article au mieux

Il existe pour cela plusieurs stratégies que nous détaillerons : Les méthodes dites extractives et les méthodes abstractives

Nous aborderons également d'autres types de résumé automatique, à savoir le résumé Multi-Documents qui, comme son nom l'indique doit résumé non plus un document, mais plusieurs. Et également le résumé mis à jour (Updated Summary en Anglais)

Le but de cette synthèse est de présenter les méthodes plus anciennes et actuelles de résumé automatique. Montrer quels sont les difficultés et les solutions proposées.

Nous traiterons également des différentes façons d'évaluer la qualité d'un résumé produit (méthodes automatiques et manuelles) et des pistes de recherches en résumé automatique

Définitions des termes

Sacs de mots (Bag of words) : Façon plus que compacte de traiter un texte. Au lieu de garder l'enchainement des mots, on represente le texte par un dictionnaire avec les mots présents (et leur nombre d'occurrence.)

Exemple :

```
Texte : Le chat ne mange pas le chat.
Sac de mots : 'Le' 2; 'chat' 2; 'ne' 1; 'pas' 1; 'mange' 1
```

Stop Words : Mots considérés comme non-pertinents car il ne contribue pas au sens du texte
Exemple :

```
Aussi, pas, ça, etc ...
```

Méthodes Extractives : On sélectionne plusieurs phrases du texte à résumer (les plus pertinentes) et on les concatène afin d'obtenir celui-ci. C'est la méthode la plus courante pour du résumé automatique.

Méthodes Abstratives : Dans cette approche, on génère les phrases qui composent le résumé (Nous détaillerons les différentes approches plus bas)

Résumé Multi-Documents : Au lieu d'appliquer le résumé à un seul document, le but étant de résumé plusieurs documents avec un seul texte

Résumé mis à jour : Forme dérivée du résumé Multi-Documents, on considère deux documents (ou deux corpus de documents) A et B qui sont dans le même thème. Le but du résumé mis à jour est de résumer le document B, sachant qu'on a déjà lu le document A. (c'est à dire faire une mise à jour de l'information de A grâce à B)

Difficultés

Les difficultés du résumé automatiques sont multiples.

- La première est la compréhension des différents thèmes, comment parvenir à capturer le sens et le sujet d'un ou de plusieurs documents ? Pour répondre à ce problème la communauté a proposée deux approches, la première, le résumé abstractif, propose l'utilisation des méthodes offerte par le NLP (Natural Language Processing) pour comprendre le sens d'un document, c'est une méthode délicate à mettre en place car elle est évidement dépendante de la langue étudié. La seconde, le résumé extractif, est plus communément utilisée. Elle repose sur des méthodes purement algorithmique, qui se concentrent uniquement sur la structure du document sans faire aucune interprétation.
- La seconde difficulté consiste à éviter la redondance, surtout dans les résumés extractifs, en effet il est possible, notamment dans les long documents, que deux phrases choisies par le résumeur comportent exactement la même information, ce qui est contraire à l'idée même du résumé.
- Dans la même veine, la cohérence globale du résumé peut être une difficulté, en effet si le texte initial ne suit pas une trame chronologique linéaire il peut être délicat de replacer les phrase dans l'ordre, ce problème est lié au problème plus général de la lisibilité globale du résumé final, des phrases porteuses de sens mises bout à bout ne seront pas toujours très compréhensibles.
- Enfin plus généralement nous nous pencherons sur l'évaluation des résumé automatique en fin de synthèse.

Le résumé automatique : Historique, Avancées et Évaluation

Historique et Articles fondateurs

Comme nous le disions, les débuts du résumé automatique remonte aux années 50. Dans un premier temps le but était de créer des abstracts automatiquement pour des articles scientifiques, les méthodes les plus primaires était basées sur de l'extraction aléatoire de phrases dans le texte, ou extraction à intervalles régulier. Nous allons voir comment améliorer cette approche naïve, avec des stratégies qui ont inspirés des méthodes plus récentes.

Méthodes Extractives

Luhn : The automatic creation of literature abstract 1958 Dans cet article, la problématique est la suivante "Comment bien sélectionner les phrases à extraire pour une tâche de résumé automatique ?"

Le principe général est le suivant : On attribue à chaque phrase un score de pertinence (0 ou 1 dans un premier temps). Le score va dépendre des mots qui composent la phrase, si les mots sont pertinents, la phrase va l'être. On ne tient pas compte du tout de l'enchainement des mots, juste de leur présence (ou non). Pour savoir si un mot est pertinent ou non, il faut se baser sur sa fréquence dans le texte, on va donc utiliser des sacs de mots (Ici, on ne tient pas compte des différentes formes que peuvent prendre un mot Ex : échantillon, échantillons, échantillonner représentent des mots différents) Seulement, il faut moduler un peu cette fréquence. En effet un mot qui apparait très peu de fois n'est pas pertinent et doit être vu comme du bruit. Au contraire un mot qui apparait dans quasiment toutes les phrases du texte n'est pas non plus pertinent (*Comme par exemple les stops words*)

Ainsi on va mettre le score de tous les mots qui ont une fréquence inférieur à C ou supérieur à D à zéro et mettre tous les autres à 1 :

$$score(\text{mot}) = \mathbb{1}_{\{C<\text{freq}_{\text{mot}}<D\}}$$

Le critère pour inclure une phrase dans l'abstract est : Si la phrase contient des mots pertinents séparés de moins de 4 mots non pertinents, alors la phrase est incluse dans l'abstract.

Ce critère peut être affiné en donnant un score plus important au phrases qui contiennent un rapport plus important de mots pertinents par rapport au nombre total de mots

Exemple :

```
4 mots pertinents sur 8 = 0.5
6 mots pertinents sur 10 = 0.6 => Meilleur phrase
```

La difficulté repose sur le fait qu'il faut bien sélectionner C et D. Les résumés produits sont plutôt pertinents, seulement ils sont souvent assez redondants et parfois l'enchainement des phrases n'est pas bon. Mais cette approche a servi de base et est encore utilisé par la plupart des méthodes extractives.

Edmunson : New Methods in Automatic Extracting 1969 Edmunson se penche en 1969 sur les résumés automatique, il reprend les idées de Luhn 1958 en les enrichissant avec de nouvelles heuristiques. Il est le premier à utiliser les mots composant le titre et le sous titre comme base de recherche dans le texte. Il conserve l'hypothèse principale de Luhn qui dit que l'on peut fournir un résumé décent composé uniquement de phrases du ou des documents concernés. Cependant il fait la différence entre deux types de résumés : les résumé indicatifs et les résumé informatifs. Les premiers vont présenter une version abrégé du document, signalant le ou les thème de ce document. et présentant la structure générale du texte. Le second remplace la lecture du texte primaire, il ne retient que l'information pertinente et élimine tout le superflu. Dans son article Edmunson se penche sur les résumés indicatifs.

Axe de Recherche Actuel

La recherche actuelle se focalise principalement sur les résumés automatiques de type extractifs, les résumés de type abstractifs étant très complexes, on cherche maintenant à trouver les heuristiques les plus pertinentes pour noter chacunes des phrases du ou des documents. On cherche ensuite à traiter la redondance et la temporalité dans les résumés uni ou multi-documents. Les outils utilisés actuellement sont :

- Maximum de vraissemblance
- Latent semantic analysis
- HMM (Chaînes de Markov Cachée (cf John M. Conroy et Dianne P.O'Leary))
- CRF (Conditional Random Field Nowshath K. Batcha)
- SVM (Support vecteur machine)
- programmation dynamique

Dans la suite d'Edmundson : Résumé automatique de texte avec un algorithme d'ordonnancement *Usunier et al.* La stratégie adoptée par Usunier est la même que Edmunson. Il se sert des mots du titre comme base de mots pertinents et de la fréquence des mots du texte.
Ce qu'il propose est d'enrichir la base du titre avec des mots synonymes, une

Analyse du Contexte Locale (LCA) et des co-occurrences de mots ("Les mots du titre apparaissent au côté de quels mots ?"), pour ainsi créer plusieurs score pour une même phrase (la pertinence par rapport au contexte, par rapport aux synonymes, par rapport à la co-occurrences)

Les nouveautés apportés portent aussi sur la combinaison des phrases, car maintenant il n'y a plus un seul mais plusieurs scores, ainsi il faut choisir les meilleures phrases mais selon plusieurs critères. Usunier utilise un algo d'ordonnancement pour classer ces phrases, et choisit donc les premières.

L'avantage de cette approche est de pouvoir déterminer quels sont les meilleurs critères quand on veut sélectionner une phrase ou non pour le résumé.

Après évaluation de son modèle, on retient que les critères pour qu'une phrase soit dans le résumé sont :

La présence dans la phrase : * des mots du titre enrichi par :
* L'analyse du contexte locale * L'analyse des co-occurrences entre mots

Et l'utilisation d'un algo d'ordonnancement accélère le processus de classification des meilleurs phrases.

Text Summarization via Hidden Markov Models 2001 John M. Conroy et Dianne P.O'Leary

Dans leur approche les auteurs calculent la probabilité à postériori qu'une phrase soit incluse dans le résumé final. Par opposition au naive Bayes classique ou la probabilité que chacune des phrases soit dans le résumé est indépendante du fait que ses voisines soit dedans, ce n'est pas le cas ici. Pour chaque phrase ils considèrent trois attributs, sa position, le logarithme du nombre de mots de cette phrase plus un et le logarithme de la probabilité de la présence des mots de cette phrase sachant les mots du document.

On les notera respectivement $p, o_1(i), o_2(i)$.

La chaîne de Markov proposé possède $ 2 s + 1 $ états, avec s états dans le résumé, et $s + 1$ en dehors.

Ici on a besoin de donnée d'entrainement (training data) pour créer la matrice de transition. Elle est formée à partir du maximum de vraisemblance de chacune des transition dans les données d'entrainement.

Chaque état i est associé à une fonction $b_i(O) = P_r(O|state_i), O$ représente le vecteur d'attribut, le maximum de vraisemblance sera utilisé pour déterminer pour estimer la fonction (sa moyenne ainsi que sa matrice de covariance), ils ont estimé les $2s + 1$ moyenne mais fait l'hypothèse que chacune des fonctions partageait la même matrice de covariance.

Pour chaque phrase on calcule la probabilité qu'elle appartienne à chacun des états i, si i est pair la probabilité correspond à la probabilité qu'elle soit dans le résumé, si i est impair, à la probabilité qu'elle n'y soit pas.

On somme ensuite sur les i pairs pour determiner la probabilité totale.

L'expérimentation a été ménée sur 1304 documents résumés par la même personne. Ils ont découpé chacun de ces documents en deux parties (train & test). Ils ont ensuite comparé les deux résumés, ceux produits par l'homme, et ceux produits par la machine à l'aide de la métrique suivante :

$$F_1 = 100 * \frac{2r}{k_h + k_m}$$

$k_h$la longueur du résumé humain et k_m la longueur de celui de la machine. Et r le nombre de phrase en commun entre les deux résumés.

La méthode HMM apparaît ici bien meilleur que la méthode DimSum.

Résumé automatique multi-document et indépendance de la langue : une première évaluation en français 2009 Florian Boudin et Juan-Manuel Torres-Moreno

On commence par construire le graphe après avoir fait un minimum de prépocessing sur les phrases : on enlèves les stopwords, les majuscules et la ponctuation (trois processus simples)

Ensuite on calcule la similarité à l'aide de la mesure LCS, la plus grande chaine de caractère en commun entre deux segments.

$$sim(S_1, S_2) = \alpha * \cos(S_1, S_2) + (1 - \alpha) * LCS^*(S_1, S_2); 0 < \alpha < 1$$

$$LCS^*(S_1, S_2) = \frac{1}{|S_1|} * \sum_{m_2 \in S_1} max_{m_2 \in S_2'} LCS(m_1, m_2)$$

avec S_2' l'ensemble de mots du segment S_2 dans lequel les mots m_2 qui ont déjà maximisé $LCS(m_1, m_2)$ durant les étapes précédentes du calcul, ont été enlevés. Le facteur $|S_2|^{-1}$ permet la normalisation du calcul.

La dernière partie est variable, c'est la pondération des segments, ici ils présentent trois méthodes :

Popularité : méthode naive, une phrase est populaire si elle a beaucoup de liens dans le graphe

$$popularit(s) = card\{adj[s]\}$$

avec $card\{adj[s]\}$ la cardinalité de l'ensemble de sommets reliés à s dans la matrice d'adjacence.

LexRank : basée sur le pageRank adaptée au texte, la grosse différence avec le pageRank habituel étant que le graphe n'est pas dirigé, les liens étant basés sur la similarité entre deux segments. Cette méthode utilise aussi la popularité calculée dans la première méthode.

$$p(s) = (i - d) + d * \sum_{v \in adj[s]} \frac{p(v)}{popularit(v)}$$

Avec $popularit(v)$ le nombre d'arêtes du sommet v et d un facteur d'amortissement (souvent 0.85).

TextRank : Variante de la méthode précédente où l'on utilise pas la popularité

Après avoir selectionnées les phrases on les assembles, on applique un seuil de similarité afin d'éviter la redondance et on positionne les phrases en fonction de la chronologie des documents.

$$p(s) = (i - d) + d * \sum_{v \in adj[s]} \frac{Sim(s, v)}{\sum_{z \in adj[v]} Sim(z, v)}$$

Résultats

En testant toutes ces méthodes c'est les méthodes à base de graphes qui sont les plus efficaces, ils ont calculé les scores à l'aide de la méthode ROUGE (Recall-Oriented Understudy for Gisting Evaluation cf partie évaluaion). Et entre LexRank et TextRank c'est TextRank qui prime

DualSum : Résumé Multi-Documents mais un but différent par *Delort et Alfonseca* Cet article est un bon exemple d'application légèrement différentes du résumé automatique Multi-Documents, car ici, l'article traite de résumé mis à jour.

L'approche la plus courante pour du résumé mis à jour est d'appliquer un algorithme de résumé sur le document B (*Pour rappel* : **A** : Document de base. **B** : Nouveau document qui met à jour l'info de **A**), puis de retirer des phrases qui sont redondantes par rapport aux documents A

Dans cet article l'approche est différente, car il est question de modèle Bayesien. Le principes est le suivant :

On ne considère plus des mots seuls mais des bi-grammes de mots et ces bi-grammes sont issus de plusieurs distribution de probas :

ϕ^G une distribution générale
ϕ^{cd} la distribution spécifique d'un document d dans la collection c
ϕ^{Ac} La distribution spécifique de A, qui correspond au thème principal
ϕ^{Bc} La distribution spécifique de B, qui correspond à la mise à jour des infos par rapport à A

Avec un a priori (sur la répartition des distributions dans chaque document) et un échantillonage de Gibbs, le modèle va être capable d'apprendre les différentes distributions. Après l'apprentissage, le modèle ainsi créé permettra de générer des résumés basées sur la distribution ϕ^G et ϕ^{Bc} (également un peu de ϕ^{Ac} pour avoir un background du thème principal)

Abstractive From Extractive to Abstractive Meeting Summaries: Can It Be Done by Sentence Compression? 2009 F.Liu et Y.Liu

Dans cet articles les auteurs cherchent à produire des résumés abstractifs à partir de résumé extractifs. Ils appliquent des algorithmes de compressions de phrases sur les résumés. Les documents qu'ils cherchent à résumer sont des comptes rendus de réunions, ces documents sont composés en grande partie de rapports de discours oraux, ce qui implique une certaine redondance qu'on cherche ici à supprimer. La compression pour faire le lien entre les deux types de résumés, ils utilisent deux algorithmes : integer programming (IP) & $SCFG$ puis comparaison avec une compression humaine (séparation des résumés en groupe de 10 phrases puis compression par groupe par plusieurs personnes, puis sélection du meilleur par un autre humain, il ordonne aussi les 4 meilleurs pour des tests ultérieurs) . Ils utilisent en amont des formulations Markovienne de règles de grammaire pour détecter et éliminer le bruit. Ils ont eux aussi utiliser ROUGE pour évaluer leurs résumés, les scores des résumés améliorés étaient meilleurs que ceux des résumés bruts (sans améliorations) mais il reste globalement bas et les auteurs considèrent qu'on peux encore améliorer cette méthode.

Evaluations dans le domaine

Evaluation Manuelle La première approche qui semble être la plus évidente lorsque l'on parle de résumé automatique est une évaluation manuelle (ou semi-manuelle), mais selon quels critères ?

Il y a deux façons de procéder, mais dans les deux cas, on a besoin d'au moins un résumé de référence par document.

1. Une première solution est de demander à une personne de sélectionner parmi plusieurs résumés proposés celui qui lui semble le meilleur selon plusieurs critères (Delort 2012)

   - Qualité globale du résumé
   - Le résumé contenant le moins de détails non pertinents
   - Cohérence du résumé et non redondant

   Si obtient les mêmes résultats entre les résumés de référence et les résumés créés (voir mieux) on peut en conclure que l'algorithme de résumé est bon.
2. Dans le cas des méthodes extractives, il suffit de comparer les phrases du résumé de référence avec les phrases de la méthode à évaluer. Deux critères proposés (Usunier 2009):

   - Rappel : $\frac{\#\text{ Phrases bonnes}}{\#\text{Phrases extraites}}$
     Ce critère permet de savoir si il n'est extrait pas trop de phrases superflues

- Précision : $\frac{\#\text{Phrases bonnes}}{\#\text{Phrases dans reference}}$
  Ce critère permet de savoir si sont bien extraites l'ensemble des phrases pertinentes

Cette méthode semi-manuelle (on a besoin quand même d'un résumé de référence, qui peut être humain) permet également de juger de la qualité d'un algorithme de résumé automatique seulement elle n'est applicable qu'aux méthodes extractives

Evaluation Automatique Pour l'évaluation automatique c'est Recall-Oriented Understudy for Gisting Evaluation plus communément appelé ROUGE qui prime. Ce set de métriques en comporte 5 :

- ROUGE-N : Basée sur les co-occurences statistiques des n-upplets.
- ROUGE-L : Basée sur le LCS vu plus haut, prend en compte l'organisation de la structure des phrases similaires et identifie le plus long n-gramme apparaissant plusieurs fois automatiquement.
- ROUGE-W : Repose sur $ROUGE - L$ en appliquant des poids sur les séquences afin de favoriser les LCS consécutives.
- ROUGE-S : S pour $skip - bigram$ comme la méthode $ROUGE - N$ mais sans prendre en compte les bigrams.
- ROUGE-SU : U pour $unigram$, comme $ROUGE - S$ mais sans prendre en compte les unigrams.

Pistes de recherche Nous constatons que les méthodes abstractives sont beaucoup moins présentes que leurs homologues extractives. Il reste encore de place pour développer ce type de méthode et enrichir par la même occasion les algorithmes de génération de texte écrit.

Il est à noter également qu'il ne s'agit plus uniquement de faire des abstracts d'articles scientifiques comme c'était le cas dans les années 50. Les types de résumé changent, on se borne plus aux articles scientifiques et des résumés de document simple, on parle maintenant de résumé Multi-Documents et de résumé mis à jour par exemple. On pourrait aussi penser à des résumés non-exhaustifs comme par exemple générer des synopsis de film ou de livre (Ainsi il faudrait ne pas dévoiler toutes les informations)

Les méthodes d'évaluations automatiques sont également un pan majeur de la recherche (de façon générale dans le TAL) car il est long de créer des résumés à la main, et tout autant de comparer plusieurs résumés de références avec les résumés tests. ROUGE existe déjà, mais il reste encore difficile à utiliser.

Conclusion

Nous pouvons constater qu'il existe plusieurs façons de créer des résumés automatiques. On différencie les méthodes extractives, qui cherchent à garder les phrases les plus représentatives du texte, des méthodes abstractives qui créent un résumé sans coller directement au texte. Les premières sont plus répandus et ont l'avantage d'être plus simple à mettre en place tandis que les secondes ont pour but de créer des résumés plus cohérents et lisibles mais sont beaucoup moins répandus car plus complexes.

On retrouve beaucoup de stratégies différentes : * Des représentations en sacs de mots * Utiliser les mots du titre (enrichis avec du contexte, des synonymes etc …) pour trouver les phrases pertinentes * Des modèles basé sur des chaines de Markov * Des modèles à base de graphes * Des modèles Bayesien génératif

Nous avons vu qu'il existe des méthodes d'évaluations à la fois manuelles (comparaison avec un résumé de référence), mais également automatiques (comme ROUGE).

On pourrait résumé notre synthèse automatiquement et le mettre ici …