Compression entropique de phrases contrôlée par un perceptron

Thierry Waszak[1], Juan-Manuel Torres-Moreno[1,2]

[1]Laboratoire Informatique d'Avignon – BP 1228 FR-84 911 cedex 09 Avignon – France

[2]Ecole Polytechnique de Montréal – Montréal (Québec) H3C 3P8 – Canada

Abstract

Sentence compression is a necessary component to the generation of abstracts. Previous studies focused mainly on the syntactic tree representation of the sentence. Our approach is a statistic approach, which does not use syntactic trees, which can be inaccurate in sentence analysis. At the core of our system is a language model based on lemma bigrams and part-of-speech tags (only a shallow parsing is performed) as well as an entropy computation over sentences to retrieve the best-compressed sentences. We also introduce the perceptron which is used to classify the compressed and non-compressed sentences and to indicate whether or not a sentence should be compressed.

Résumé

La compression automatique de phrases est un module nécessaire à la génération d'abstracts. Les études précédentes portant sur la compression de phrases sont pour beaucoup basées sur les arbres syntaxiques. Notre système se veut statistique mais n'utilisant pas ces arbres qui peuvent ne pas être suffisamment robustes dans l'analyse des phrases. Nous utilisons principalement un modèle de langage bigramme sur les lemmes et sur les étiquettes morpho-syntaxiques des mots (une simple analyse syntaxique de surface est nécessaire) ainsi qu'un calcul d'entropie pour retrouver la meilleure compression possible d'une phrase. Nous introduisons également un perceptron pour définir si une phrase doit ou non être compressée.

Mots-clés : compression de phrases, résumé automatique, perceptrons, méthodes statistiques.

1. Introduction

L'étude de la compression automatique des phrases que nous présentons est en rapport aux résumés de documents : le but est d'améliorer les systèmes de résumé existants qui sont basés principalement sur l'extraction des phrases des documents. Il s'agit donc d'extraits ou *extracts*. On veut se rapprocher du vrai résumé, c'est-à-dire générer un « *abstract* » qui correspond au résumé tel que l'aurait produit un humain. L'étude de Jing et McKeown (2000) sur les différentes actions effectuées par les résumeurs professionnels montre que la compression d'une phrase est un élément nécessaire à la génération de l'*abstract*. D'autres opérations comme la combinaison de deux phrases, la transformation syntaxique ou le tri des phrases sont aussi utilisées pour générer les *abstracts*. Nous nous sommes focalisés, dans cette étude, sur la compression de phrases, qui consiste à supprimer certains constituants d'une phrase. Cette suppression peut intervenir à différents niveaux de granularité : la suppression peut porter sur un mot, ou sur une proposition entière dans la phrase. Deux éléments sont à prendre en compte dans la compression des phrases pour avoir une bonne compression : la grammaticalité et la concision. La grammaticalité consiste à s'assurer qu'une phrase est grammaticalement correcte, c'est-à-dire qu'elle est bien formée et qu'elle respecte

la grammaire du langage. La concision, quant à elle, correspond au fait qu'une phrase compressée doit retenir l'information essentielle de la phrase originale. Ainsi, l'idée essentielle d'une phrase doit pouvoir se retrouver dans la phrase compressée correspondante. C'est à l'aide de ces deux mesures (non évidentes à réaliser) que l'on peut définir si une phrase est correctement compressée ou non. Il existe deux grandes approches pour la compression de phrases : l'approche linguistique qui consiste à définir des règles et l'approche statistique qui utilise un corpus approprié afin de détecter des régularités statistiques exploitables. Dans cette dernière approche, le corpus contient un ensemble aligné de phrases originales et de phrases compressées. L'avantage des méthodes statistiques est qu'elles extraient automatiquement les règles de compression. Toutefois il faut disposer d'un corpus assez large, ce qui n'est pas toujours évident. Notons que si une phrase est composée de n mots, il y a 2^n compressions possibles dans l'absolue ; mais n'étant pas toutes grammaticalement correctes. Le but est de retrouver, dans cet ensemble, les meilleures phrases candidates compressées. Nous définissons le « taux de compression » des phrases comme le pourcentage de mots (ponctuations comprises) gardés dans la phrase compressée (pour être en accord avec les précédentes études). Dans la section 2, nous avons réalisé un état de l'art afin de situer nos motivations pour le développement de notre système. Le système sera décrit dans la section 3, les corpus utilisés en section 4 et les résultats en section 5 avant de conclure et de donner quelques perspectives.

2. Etat de l'art

Plusieurs recherches ont été menées sur la problématique de la compression automatique de phrases. Une des premières et qui est utilisée comme référence par beaucoup d'autres, est celle effectuée par Knight et Marcu (2000), qui est basée sur des approches statistiques. Elle a donnée lieu à deux algorithmes :

Le modèle du canal bruité. L'hypothèse est qu'une phrase non compressée *l* était à l'origine une phrase compressée *c* et qu'il lui a été ajouté du texte optionnel. Le model est donc constitué de la source *P(c)* qui affecte une plus grande probabilité aux phrases bien formées ; du canal *P(l|c)* qui privilégie les phrases préservant l'information essentielle et du décodeur *P(c|l)* qui va rechercher la meilleure compression : la phrase *c* qui maximise *P(c|l)*. Ceci revient a maximiser $P(c)\times P(l|c)$ (règle de Bayes). Ces probabilités sont appliquées aux arbres syntaxiques représentant les phrases. Un modèle bigramme sur les mots des feuilles est appliqué pour assurer la grammaticalité. Ainsi *P(c)* se focalise sur l'information essentielle contenue dans les phrases. Ce modèle compresse peu les phrases (car il favorise les phrases longues : les probabilités sont pondérées avec la longueur de la compression).

Le modèle à base d'arbres de décision. Comme pour le modèle antérieur, il est basé sur les arbres syntaxiques. Le but est de réécrire l'arbre représentant la phrase *t* dans le plus petit arbre représentant la phrase compressée *s*. Il utilise une méthode à base de séquences d'actions *shift reduce drop* (issue du paradigme *shift-reduce parsing*). Ce modèle est plus flexible que le précédent car il permet de réécrire l'arbre de *t* en n'importe quel autre arbre représentant *s*, tant que les mots de *s* et *l* sont dans le même ordre (*t* peut se réécrire en s_1 ou s_2 / les tags POS – les catégories syntaxiques peuvent changer). Ce modèle est déterministe et produit donc une seule compression ; il a l'avantage d'être très rapide et d'avoir un taux de compression très proche de celui des humains.

D'autres approches ont été imaginées à partir des ces deux modèles. Certaines essaient d'améliorer la concision. Ceci est le cas notamment de (Nguyen et al., 2004a) et (Riezler et

al., 2003), qui ont rajouté à l'arbre syntaxique d'autres informations, respectivement, le contexte sémantique de chaque nœud et feuille de l'arbre et des informations fonctionnelles sur la syntaxe. Ces méthodes ont l'avantage d'augmenter le taux de mots importants gardés, mais ils compliquent l'analyse lexicale en la rendant plus aléatoire. En effet, l'analyse syntaxique n'est jamais à 100% correcte et le niveau supérieur (sémantique, fonctionnel) est encore moins efficace. Enfin un analyseur syntaxique n'est pas toujours disponible pour toutes les langues. On peut citer (McDonald, 2006) qui n'utilise l'arbre syntaxique que comme une source de vérification annexe, il se base plutôt sur un parseur de dépendance (qui permet de repérer les relations entre sujet, verbes, compléments…). Ou encore (Nguyen et al., 2004b) qui utilise une technique de *templates of translations* : une phrase non compressé est considérée comme étant écrite dans une langue source et on veut retrouver la phrase compressée, considérée comme écrite dans la langue cible. L'idée générale du fonctionnement de cette méthode est la suivante : à l'aide d'un corpus aligné de phrases/phrases compressées, on va générer des règles en matchant les éléments identiques de deux phrases et en considérant comme variables les différences entre ces deux phrases. Le processus de compression va essayer de trouver les meilleures variables qui se substitueront aux constituants originaux des phrases. Ce modèle a un temps de calcul exponentiel (dû à la grande quantité de combinaisons de règles possibles). Pour palier à cet inconvénient, on utilise un modèle de Markov caché où les états sont les règles lexicales et les symboles observés sont les phrases à compresser. Ces modèles donnent des résultats sensiblement équivalents à ceux du système à base d'arbres de décision. Il faut aussi évoquer l'approche de Vandeghinste et Pan (2004) qui se passe également d'arbre syntaxique. Elle est basée sur les *chunks* des phrases (i.e. groupes syntaxiques : groupe nominal, verbal…) ce qui nécessite uniquement une analyse lexicale de surface (POS tags). Les *chunks* semblent être intéressants pour la compression puisque si on supprime un nom on va forcement supprimer le déterminant qui le précède. D'autres approches existent, telle la compression basée sur une analyse rhétorique du discours (Sporleder et Lapata, 2005).

Le modèle à base de sources de connaissances multiples. Une approche intéressante (Jing, 2000) utilise des bases de connaissances multiples pour la compression en vue d'une intégration dans le résumé automatique ; à savoir la syntaxe, le contexte et une analyse statistique d'un corpus. L'idée est de supprimer les constituants de la phrase seulement s'ils ne sont pas relevants du sujet principal du document. La grammaticalité est assurée par l'utilisation d'un arbre syntaxique et d'une base de données indiquant quels sont les arguments obligatoires d'une phrase. Pour s'assurer de garder les mots importants des phrases, on utilise le contexte du document (les liens qu'a un mot avec les autres dans WordNet). A l'aide d'un corpus aligné, on calcule les probabilités de suppression, de non suppression et de réduction des constituants de la phrase. A partir de ces analyses, un constituant syntaxique est supprimé s'il n'est pas obligatoire pour la cohérence grammaticale, s'il n'est pas en rapport avec le sujet du document et s'il a une bonne probabilité d'être supprimé par un humain (analyse du corpus). Les mots importants sont ici assez bien identifiés et donc conservés.

Le modèle linguistique. Dans (Yousfi-Monod et Prince, 2006) les arbres syntaxiques sont transformés en utilisant des règles écrites manuellement. Nous sommes ici dans une démarche qualitative et linguistique. On travaille sur la fonction et la position dans l'arbre syntaxique des constituants. A différentes granularités dans la phrase (proposition, constituant nominal…) on va essayer de supprimer les éléments non gouverneurs (il s'agit en quelque sorte des satellites dans une analyse rhétorique). Cette méthode se montre plus efficace sur les

textes narratifs que sur les textes plus techniques où chaque constituant a un rôle important à jouer.

3. Le système du LIA

Nous avons décidé de développer un système principalement statistique, basé sur un corpus de phrases/phrases compressées. Dans la compression, l'ordre des mots ne sera pas modifié. Comme les analyseurs syntaxiques pour le français ne sont pas suffisamment robustes, nous avons décidé de nous en passer. La base de notre système est composée de modèles de langage bigramme et trigramme portant sur les mots. Nous pouvons ainsi, grâce au modèle de langage bigramme appris sur le corpus de phrases compressées, déterminer quels mots ont une forte probabilité d'être supprimés. Le modèle trigramme quant à lui nous assure de la bonne grammaticalité des phrases compressées. Enfin, pour améliorer la concision des phrases compressées, nous utilisons un calcul sur l'entropie des phrases. En effet, l'entropie va nous renseigner sur la quantité d'information contenue dans une phrase. Le principe de notre système est donc d'essayer de retirer un mot ou groupe de mots d'une phrase originale de sorte que l'entropie de la phrase soit diminuée au minimum.

3.1. Architecture globale du système

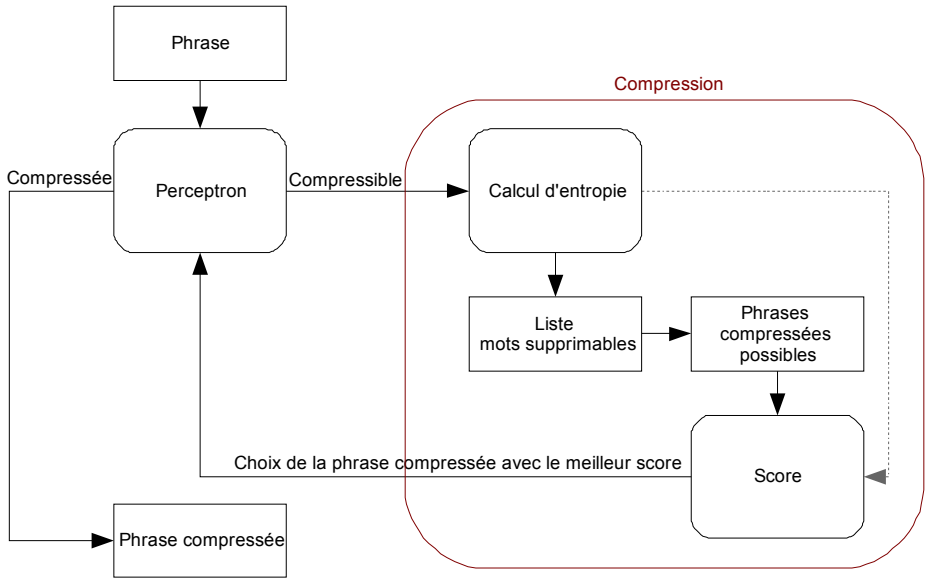

Figure 1. Architecture globale du système

Le système prend en entrée un fichier d'une phrase par ligne (les lemmes et les étiquettes morpho-syntaxiques – tags POS – sont associés aux mots grâce à une analyse syntaxique de surface conduite par TreeTagger[1]). Chaque phrase est alors analysée pour être compressée.

Le premier module de cette analyse est celui du Perceptron, dont le rôle est de dire si une phrase *s* appartient à l'ensemble des phrases compressées ou pas ; c'est-à-dire, décider si une phrase à besoin d'être compressée ou pas. Si oui le module de Compression est utilisé ; une phrase compressée est générée, et, tant que le Perceptron classe la phrase générée comme étant compressible, on répète l'opération. Lorsque le Perceptron classe la phrase générée

[1] Institute for Computational Linguistics of the University of Stuttgart. (http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/)

comme étant compressée, la compression de la phrase est (considérée) accomplie. Au final on génère les phrases compressées correspondant aux phrases originales du fichier texte.

3.2. Perceptrons

Le Perceptron est un algorithme d'apprentissage discriminant. En fonctions d'exemples : les phrases compressées et les phrases originales non compressées de notre corpus, on va classer les phrases en deux catégories : les phrases compressées (que l'on n'a donc pas besoin de compresser davantage) et les phrases non compressées (auxquelles on peut appliquer notre algorithme de compression). L'algorithme du Perceptron simule le fonctionnement des neurones. Un neurone peut être représenté par un automate caractérisé par un état interne, des signaux d'entrée x_i ($1 \leq i \leq n$) et une fonction de transition $y=g(x_1, x_2, \ldots, x_n)$. Des poids w_i sont attribués aux entrées du neurone. La sortie d'un neurone est discrète et peut avoir deux valeurs : inhibé (1) ou excité (0).

Nous utilisons la représentation précédente, en définissant les x_i comme étant des vecteurs des lemmes associés aux mots représentant une phrase s_i (présence=1 et absence=0). Les poids w_i sont calculés lors de la phase d'apprentissage du Perceptron. L'apprentissage étant arrivé à son terme, on en déduit que les ensembles des phrases compressées et des phrases non compressées sont linéairement séparables.

3.3. Algorithme de compression

Dans un premier temps il s'agit d'établir une liste de mots supprimables (cette liste privilégie les mots non fonctionnels). Ces mots sont classés en fonction de la perte d'entropie qu'ils engendreraient pour la phrase. En commençant par le premier mot dans la liste, on cherche la compression optimale de la phrase. Il s'agit d'obtenir un score de compression $S_{compression}$ en considérant la grammaticalité de la phrase $S_{trigramme}$ (obtenue à l'aide du modèle trigramme présenté dans la section 3.4), la probabilité de compression $S_{bigramme}$ (obtenue à l'aide du modèle bigramme présenté également dans la section 3.4), ainsi que le score de perte d'entropie $S_{entropie}$ que la suppression d'un (ou plusieurs) mot va entraîner. (La place du mot dans la liste des mots supprimables S_{indice} est également pris en compte.) On a donc la formule suivante (un $\lambda=0,3$ nous donne les meilleurs résultats) :

$$S_{compression} = (\lambda \times S_{trigramme} + (1- \lambda) \times S_{bigramme}) \times S_{indice} / S_{entropie}$$

La compression optimale peut être réalisée à trois niveaux :

- inter-groupe phrastique. Dans le cas d'une compression inter-groupe phrastique, on découpe la phrase en constituants séparés par les signes de ponctuations, on repère ensuite le constituant contenant les mots de la liste des mots supprimables, puis le groupe maximisant le score $S_{compression}$ est choisi.

- inter-groupe syntaxique. Après avoir obtenu les *chunks* des phrases grâce à TreeTagger (par exemple, la phrase suivante est découpée en *chunks* : *[les industries] [textiles,] [du jouet] [et] [de l'électronique] [occupent]…*), on va supprimer le *chunk*, contenant un mot de la liste des mots supprimables, maximisant le score $S_{compression}$. A ce niveau, la compression en utilisant les *chunks* est utile pour garantir une meilleure grammaticalité.

- intra-groupe syntaxique. On se place ici à l'intérieur des *chunks*. Il s'agit de supprimer simplement le mot de la liste des mots supprimables, ou un groupe de mots à l'intérieur du *chunk* tout en tenant compte du score $S_{compression}$. A ce niveau, la compression porte surtout sur les mots fonctionnels : par exemple, les mots *avant tout* seront ici supprimés.

La meilleure compression est choisie en fonction du score $S_{compression}$, elle peut être inter-groupe phrastique ou inter ou intra-groupe syntaxique.

3.4. Les modèles de langage

3.4.1. Le modèle bigramme

A l'aide de notre corpus de phrases/phrases compressées (voir section 4), nous construisons un modèle de langage bigramme qui pour un mot w_i donné, indique la probabilité que w_j suive w_i dans la compression : on a $P_{compression}(w_j|w_i)$. On se trouve ainsi dans la situation suivante : soit une phrase $s=w_1w_2w_3w_4w_5$, composée de w_i mots, si dans notre corpus d'apprentissage, nous avons rencontré une phrase $s_{compressée}=w_1w_2w_5$, on aura une forte probabilité $P_{compression}(w_5|w_2)$. On va donc décider de supprimer les mots w_4 et w_5.

Remarquons que $P_{compression}(w_j|w_i)$ est obtenue en construisant des modèles de langage sur les lemmes lem_i et les étiquettes morpho-syntaxiques pos_i associés aux mots w_i. En effet, le corpus de phrases/phrases compressées n'est pas suffisamment conséquent pour travailler avec un modèle de langage portant uniquement sur les mots. On a donc

$$P_{compression}(w_j|w_i) = \lambda \times P_{compression}(lem_j|lem_i) + (1-\lambda) \times P_{compression}(pos_j|pos_i)$$

(Après expérimentation, on fixe empiriquement $\lambda=0,5$.)

3.4.2. Le modèle trigramme

En remarquant que le modèle de langage bigramme est créé à partir d'un petit ensemble de phrases (le corpus de phrases/phrases compressées), nous avons eu l'idée de recourir à un corpus supplémentaire plus conséquent pour pouvoir juger de la bonne grammaticalité des phrases à l'aide d'un modèle de langage trigramme. Nous avons utilisé les articles scientifiques du corpus BAF[2] du RALI (Recherche appliquée en linguistique informatique) pour le modèle anglais (car le domaine d'application est l'informatique) et les documents narratifs de ce même corpus pour le modèle français. Ainsi, étant donné la suite de mots w_iw_j, on a la probabilité que w_k suive w_iw_j : $P_{grammaticalité}(w_k|w_iw_j)$.

3.4.3. Maximisation du score de l'entropie des phrases

La compression d'une phrase doit conserver les mots porteurs d'information essentielle. L'entropie calculant la quantité d'information présente dans une phrase, on peut considérer que supprimer un mot qui n'est pas porteur d'information diminuera moins le score de l'entropie que si l'on supprime un mot relevant. Nous parlons exclusivement des mots non fonctionnels, car supprimer un mot fonctionnel (un déterminent par exemple), n'apporte rien à la compression, de plus supprimer un mot fonctionnel est l'action qui générera le moins de perte d'entropie. L'entropie est calculée en utilisant le modèle de langage trigramme.

4. Les corpus utilisés

Notons tout d'abord que notre étude porte sur deux langues : le français et l'anglais. Nous essayons cependant de construire un système qui se veut suffisamment général pour pouvoir s'appliquer à d'autres langues. Les corpus de phrases/phrases compressées dont nous disposons traitent de deux domaines différents. Pour le français, nous disposons de 219

[2] http://rali.iro.umontreal.ca/Ressources/BAF/

phrases et de leurs compressions effectuées manuellement. Ces phrases sont extraites de textes généraux plutôt narratifs. Dans la suite, nous faisons référence à ce corpus sous le nom de corpus « Myriam ». Pour l'anglais, nous disposons du corpus utilisé dans (Knight et Marcu, 2000) de 1 087 phrases, accompagnées de leurs compressions. Comme eux, nous utilisons les mêmes 1 055 phrases pour l'apprentissage du modèle bigramme et du Perceptron et nous gardons 32 phrases pour le test. Ceci nous permettra de comparer notre système avec les systèmes de l'état de l'art. Ce corpus, extrait de revues informatiques, sera référencé sous le nom de corpus « Ziff-Davis ». On obtient pour ces corpus les statistiques rapportées au tableau 1.

	Myriam	Ziff-Davis
Entropie moyenne des phrases compressées	0,92	0,47
Entropie moyenne des phrases originales	1,56	0,88
Longueur moyenne des phrases compressées	0,19	0,12
Longueur moyenne des phrases originales	0,32	0,22

Tableau 1. Statistiques des corpus

Le taux de compression des phrases est de 57% pour le corpus Myriam et de 53% pour le corpus Ziff-Davis. Nous avons calculé le rapport entre l'entropie des phrases compressées et originales, respectivement de 59% pour le corpus Myriam et de 53% pour le corpus Ziff-Davis. La plus faible entropie correspond aux phrases du corpus Ziff-Davis. Cela peut s'expliquer par le fait que le corpus Myriam est constitué de quelques phrases très longues. Ce rapport d'entropie dans les deux corpus varie énormément d'une phrase à l'autre, ce qui nous fait penser que le taux moyen de compression du point de vue de l'entropie n'est pas utilisable. Nous avons effectué la même étude sur le corpus Myriam en ne tenant compte que des mots non fonctionnels. Le taux de compression est de 61% et le rapport d'entropie entre les phrases compressées et les phrases originales est de 53%. Il est logique de garder plus de mots, dans ce cas, car les mots non fonctionnels sont porteurs du sens. Les résultats sont présentés dans le tableau 2.

	Myriam
Entropie moyenne des phrases compressées	0,04
Entropie moyenne des phrases originales	0,08
Longueur moyenne des phrases compressées	0,08
Longueur moyenne des phrases originales	0,14

Tableau 2. Statistiques du corpus Myriam ne tenant compte que des mots non fonctionnels

5. Evaluation

Nous avons gardé 3% des phrases de chaque corpus pour effectuer l'évaluation des performances. En effet, nous avons voulu tester notre système avec le même nombre de, et les mêmes, phrases que l'expérimentation de Knight et Marcu (2000). Pour le corpus Myriam, nous avons aléatoirement choisi 5 phrases (équivalent à 3% du corpus). Pour mesurer la qualité de notre système, nous utilisons les mesures BLEU d'abord développées dans le cadre de traduction automatique mais maintenant utilisées dans la compression de phrases. Notons que plus le score BLEU s'approche de 1, plus le système est performant.

Corpus en français. Pour cette évaluation, nous comparons notre système (Entropie) à une baseline (suppression aléatoire des mots) et à notre premier système (Entropie sans *chunks*) qui ne considérait pas la compression inter-goupe syntaxique (pas de *chunks*) ni la

compression inter-groupe phrastique. Nous indiquons aussi les scores BLEU obtenus par l'approche linguistique[3] développée par Yousfi-Monod et Prince (2006).

	Baseline	Yousfi et Prince	Entropie	Entropie (sans *chunks*)
BLEU	0,2123	0,5246	0,5941	0,4224

Tableau 3. Score BLEU en français (corpus Myriam)

Exemple de phrases compressées par notre système. Dans le tableau 4 suivant, la première phrase est la phrase originale, la deuxième la phrase compressée manuellement et la troisième la phrase générée par notre système (Entropie) :

Phrase originale	Compression manuelle	Entropie
Construire un logement est un acte économique qui en tant que tel s'inscrit dans une logique de marché.	Construire un logement est un acte économique qui s'inscrit dans une logique de marché.	***Construire un logement est un acte dans une logique de marché.***
Les industries textile, du jouet, de l'horlogerie et de l'électronique occupent une part de plus en plus réduite.	Les industries textile, du jouet, de l'horlogerie et de l'électronique occupent une part plus réduite.	***Les industries textile, du jouet, de l'horlogerie et de l'électronique occupent une part réduite.***
Et les Japonais, en attaquant Pearl Harbour, souhaitaient avant tout protéger leur flanc au moment où ils s'emparaient des ressources pétrolières en Indonésie.	les Japonais, en attaquant Pearl Harbour, souhaitaient protéger leur flanc au moment où ils s'emparaient des ressources pétrolières en Indonésie.	***les Japonais, en attaquant Pearl Harbour, souhaitaient protéger leur flanc en Indonésie.***
FUSION DE MATRA ET D'HACHETTE entérinée par les assemblées générales extraordinaires des deux entités, le 29 décembre, l'opération prend forme d'une absorption de Matra par Hachette, rétroactive au 1 janvier 1992, le bénéfice net de l'ensemble pour 1992 est compris entre 350 et 400 millions de francs (le Monde du 31 décembre).	FUSION DE MATRA ET D'HACHETTE entérinée le 29 décembre, l'opération prend forme d'une absorption de Matra par Hachette.	***l'opération prend forme d'une absorption de Matra par Hachette, rétroactive au 1 janvier 1992, le bénéfice net de l'ensemble pour 1992 est compris entre 350 et 400 millions de francs.***

Tableau 4. Exemple de phrases en français compressées par notre système

Remarque : La dernière phrase est grammaticalement correcte, contrairement à la phrase produite par notre premier système qui était :

fusion de matra de matra par hachette rétroactive 1 janvier ensemble 1992 est compris millions de francs le monde 31 décembre

L'introduction des *chunks* et de la compression inter-groupe syntaxique et inter-groupe phrastique prouve ici tout son intérêt (Sporleder et Lapata, 2005 ; Yousfi-Monod et Prince,

[3] http://www.lirmm.fr/~yousfi/Compression.php

2006). Sur les quelques phrases testées ci-dessus, on remarque aussi que la compression de notre système est plus agressive que la compression manuelle. Toutefois, les mots portant l'information essentielle semblent bien être retenus. La grammaticalité, elle aussi, est bonne. Pour les phrases générées par le système de Yousfi-Monod et Prince, la grammaticalité trouvée n'est pas aussi bonne. De plus ce dernier système à un taux de compression de 80%, alors que notre système à un taux de 61% bien plus proche du taux de compression obtenu lors de la compression manuelle (57%).

Corpus en anglais. Pour cette évaluation, nous disposons des phrases générées par l'approche de Knight et Marcu (2000) des arbres de décision et du canal bruité, nous allons donc pouvoir comparer nos compressions de phrases (Entropie) avec les modèles précédemment cités.

	Baseline	Arbres de décision	Canal bruité	Entropie
BLEU	0,3083	0,5919	0,4544	0,5142

Tableau 5. Score BLEU en anglais (corpus Ziff-Davis)

On remarque que notre système semble être plus performant que celui du canal bruité : les phrases générées par notre approche semblent avoir une meilleure grammaticalité que celle du canal bruité. En effet, le canal bruité utilise seulement un modèle de langage bigramme. Or notre approche valide la grammaticalité des phrases à l'aide d'un modèle de langage trigramme ainsi qu'avec l'utilisation des *chunks* (compression inter-groupe syntaxique).

Comparaison avec une autre implémentation. Nous avons récupéré une implémentation d'un système existant (pour l'anglais) et l'avons testé. Le système en question fut développé par Jacob Balazer[4]. Cette approche est semblable à l'approche à base d'arbres de décisions présentée dans (Knight et Marcu, 2000). Le système utilise un analyseur syntaxique : le parseur de Charniak[5] qui génère des arbres syntaxiques et des règles de transformation d'arbres sont apprises et exécutées pour la compression. Le système s'avère être très lent, à cause principalement de l'analyseur syntaxique. Le temps d'exécution de la compression d'une phrase est de l'ordre de quelques minutes alors que notre système est en dessous de la seconde. Les deux systèmes sont globalement grammaticalement corrects, mais il semble que le nôtre ait une meilleure concision (l'entropie permet un meilleur choix des mots à supprimer). En utilisant les mesures BLEU, on obtient les résultats suivants :

	Baseline	Balazer	Entropie
BLEU	0,2197	0,6968	0,7782

Tableau 6. Score BLEU en anglais comparant le système de Balazer avec le nôtre

Le tableau 7 présente des exemples de phrases générées par les différents systèmes. La première phrase est la phrase originale, la deuxième la phrase générée par le système de Balazer et la troisième la phrase générée par notre système (Entropie) :

[4] http://www-personal.umich.edu/~balazer/sc/

[5] http://www.cs.brown.edu/people/ec/#software

Phrase originale	Balazer	Entropie
The VIO subsystem supports the full range of PC-based video adapters and displays, including monochrome, CGA, EGA, and VGA.	The subsystem supports the full range, including monochrome, CGA, EGA, and VGA.	The VIO subsystem supports the full range of PC-based video adapters and displays, including monochrome.
Satori Software last month began shipping the first stand-alone module in a new accounting software series that features object-oriented programming and an integrated desktop publishing environment.	Satori Software began shipping the first stand-alone module in a new accounting software series that features an integrated desktop publishing environment.	Satori Software month in a new accounting software series features object-oriented programming and an desktop publishing environment.

Tableau 7. Exemple de phrases en anglais compressées par notre système comparées avec les phrases compressées par le système de Balazer

6. Conclusions et perspectives

Nous avons développé un système de compression automatique de phrases basé sur un calcul d'entropie et piloté par un perceptron. Ce système s'avère être plus rapide qu'un système nécessitant une analyse syntaxique poussée. Nous utilisons simplement une analyse syntaxique de surface, qui améliore le système dont les bigrammes et trigrammes de mots sont la base. Les difficultés initiales rencontrées par notre système sur les phrases longues ont été résolues en utilisant une compression inter-groupe syntaxique et inter-groupe phrastique. Notre système se trouve fonctionner pour l'anglais mais aussi pour le français. Le seul autre système compressant les phrases françaises (à notre connaissance) est celui présenté dans (Yousfi-Monod et Prince, 2006). Ce système utilise une approche linguistique alors que notre approche est statistique. De plus il a un taux de compression élevé par rapport au nôtre qui est très proche du taux de compression moyen des phrases compressées manuellement. L'idée de Jing (2000) de ne pas supprimer les mots en rapport avec le sujet du document nous semble intéressante pour améliorer notre système. Nous pensons aussi que l'idée d'utiliser un perceptron comme pré-compression et un couplage d'autres algorithmes (basés sur des approches linguistiques comme celui de Yousfi-Monod et Prince, 2006) serait prometteur. Enfin, l'utilisation d'un perceptron à séparation optimale comme proposé par Torres-Moreno et al. (2002) pourrait améliorer les résultats.

Remerciements

Nous voulons remercier l'ANR qui a financé partiellement ces travaux de T. Waszak avec une subvention du projet PIITHIE (LIA-Avignon) sous la responsabilité de Patrice Bellot.

Références

Clarke J. and Lapata M. (2006). Models for sentence compression: a comparison across domain, training requirements and evaluation measures. In *Proc. of 21st ACL*, pages 377-384.

Hori C. and Sadaoki F. (2004). Speech summarization: an approach through word extraction and a method for evaluation. *IEICE, Transactions on Information and Systems*, Vol(E87-D1): 15-25.

Jing H. and McKeown K. R. (2000). Cut and paste based text summarization. In *Proc. of 1st ACL* pages 178-185.

Jing H. (2000). Sentence reduction for automatic text summarization. In *Proc. of 6th ANLP*, pages 310-315.

Knight K. and Marcu D. (2000). Statistics-based summarization - step one: sentence compression. In *Proc. of 17th NCAI and 12th CIAAI*, pages 703-710.

Lin C.-Y. (2003). Improving summarization performance by sentence compression - a pilot study. *Annual meeting of the ACL, Proc. of 6th IRAL'03*.

McDonald R. (2006). Discriminative sentence compression with soft syntactic constraints. In *Proc. of 11th EACL*, pages 297-304.

Nguyen M. L. and Horiguchi S. (2003). A sentence reduction using syntax control. *Annual meeting of the ACL, Proc. of IRAL 2003*, Vol.11, pages 146-152.

Nguyen M. L., Shimazu A., Horiguchi S., Ho T. B. and Fukushi M. (2004a). Probabilistic sentence reduction using support vector machines. In *Proc. of COLING*, pages 743-749.

Nguyen M. L., Horiguchi S., Shimazu A. and Ho B. T. (2004b). Example-based sentence reduction using the hidden Markov model. *ACM, Transactions on Asian language Information processing*, Vol.3(2), pages 146-158.

Riezler S., King T. H., Crouch R. and Zaenen A. (2003). Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar. In *Proc. of HLT/NAACL*, pages 118-125.

Sporleder C. and Lapata M. (2005). Discourse chunking and its application to sentence compression. In *Proc. of HLT/EMNLP*, pages 257-264.

Torres-Moreno J. M., Aguilar J. C. and Gordon M. B. (2002). Finding the number minimum of errors in N-dimensional parity problem with a linear perceptron. *Neural Processing Letter,* 16(3): 201-210.

Turner J. and Charniak E. (2005). Supervised and unsupervised learning for sentence compression. In *Proc. of the 43rd ACL*, pages 290-297.

Unno Y., Ninomiya T., Miyao Y. and Tsujii J. (2006). Trimming CFG parse trees for sentence compression using machine learning approaches. In *Proc. of COLING/ACL*, pages 850-857.

Vandeghinste V. and Pan Y. (2004). Sentence compression for automated subtitling: a hybrid approach. In *Proc. of ACL, Workshop on text summarization*.

Yousfi-Monod M. and Prince V. (2006). Compression de phrases par élagage de leur arbre morpho-syntaxique. Une première application sur les phrases narratives. *RSTI-TSI*, pages 437-468.