Reconnaissances des divisions logiques complexes par la classification des lignes de corpus

Hassan Saneifar*,** Stéphane Bonniol **
Pascal Poncelet*, Mathieu Roche*

*LIRMM, Université Montpellier 2
**Satin Technologies

1 Introduction

Plusieurs domaines d'application comme la Recherche d'Information (RI) ou la traduction automatique utilisent des méthodes de segmentation de textes. En effet, la segmentation de texte est le découpage automatique d'un texte en unités plus petites. Il existe trois catégories principales de méthodes de segmentation : thématique, fenêtre, et discours. La segmentation thématique consiste à identifier différents thèmes véhiculés par le texte, pour le segmenter en des blocs thématiques (Tarek, 2003). Dans la deuxième catégorie, la segmentation s'effectue selon des fenêtres (des lignes ou des phrases) de taille fixe ou variable. Avec l'approche appelée "passage de discours", la segmentation s'effectue sur la base de la structure logique (unités de discours) de documents comme les paragraphes ou les sections (Kaszkiel et Zobel, 2001).

Le choix du type de segmentation dépend des objectifs et du domaine d'application. Nous traitons ici des données textuelles complexes, en particulier des fichiers logs. Ces données, issues du monde industriel, représentant une source principale d'information sont utilisés dans les systèmes RI. Les caractéristiques de ces données, tel que l'aspect multi-sources, multi-vocabulaire et multi-structures les différencient de documents classiques écrits en langue naturel. Considérant ces caractéristiques, nous nous intéressons aux méthodes de "passages de discours" qui identifient les structures logiques comme des segments. Il existe des solutions afin d'identifier les structures logiques classiques telles que les paragraphes. Nous pouvons également exploiter les éléments marquant les structures logiques, comme les lignes blanches ou les alinéas. Ces éléments sont appelés les divisions logiques. Or, dans les fichiers logs, n'existant pas de notions telles que paragraphe ou section, les structures logiques classiques ne sont pas significatives. Pourtant Il existe, par exemple, des structures comme des tableaux, des blocs de données et des chaînes de caractères particulières marquant le début de nouvelles informations. Ces structures textuelles, plus complexes que des structures classiques, sont utilisées afin de regrouper des idées et des informations. Ainsi, nous les considérons comme des Structures Logiques (SL) des fichiers logs.

Notre objectif est donc de proposer une approche de reconnaissance des Divisions Logiques (DL) dans les données textuelles complexes. Ainsi, nous cherchons à mettre en place un classifieur qui associe les lignes d'un corpus non expertisé à une classe : classe des lignes représentant une division logique (positive) et classe des lignes non associées à une division

logique (négative). Contrairement aux tâches de classification de textes qui sont fondées sur les contenus (voir (Pessiot et al., 2004)), la reconnaissance d'une division logique doit s'effectuer selon la structure et la mise en page des documents. Ainsi, le challenge est de définir un ensemble de descripteurs (features) pertinents qui caractérisent les structures logiques des documents, ici les fichiers logs. Bien que dans cet article nous nous appuyons sur des données de type "fichier logs", notre approche est applicable sur tous les types de données textuelles qui ont des structures logiques complexes.

La section 2 présente notre approche de reconnaissance des DL fondée sur un système d'apprentissage supervisée. Nous développons dans les sections 3 la méthode automatique d'acquisition des descripteurs. La section 4 est consacrée aux expérimentations.

2 Classification pour la reconnaissance des divisions logiques

Dans les textes classiques, nous pouvons, par exemple, considérer un "*alinéa*" comme une des caractéristiques permettant d'identifier la structure logique de paragraphe. Afin de pourvoir modéliser les structures logiques complexes, nous identifions donc leurs *caractéristiques syntaxiques*. Celles-ci permettent de distinguer le début d'une structure logique (c.-à-d. une division logique) des autres lignes dans les documents. Nous appelons ces caractéristiques syntaxiques les *descripteurs*. Autrement dit, nous caractérisons donc les structures logiques des fichiers logs en définissant un certain nombre de "descripteurs". Dans ce but, nous avons opté pour une méthode automatique d'acquisition des descripteurs dans un corpus expertisé. Nous développons cette méthode dans la section 3.
Une fois l'ensemble des descripteurs déterminé, nous considérons chaque ligne du corpus expertisé comme une instance positive ou négative. Chaque ligne est représentée sous forme d'un vecteur booléen dont chaque élément identifie la présence ou l'absence d'un des descripteurs autour de la ligne. Ainsi, nous obtenons un jeu de données d'apprentissage en fonction des descripteurs. Ensuite, un processus d'apprentissage automatique supervisé fondé sur une méthode de classification peut être appliqué. Cela permet de créer un modèle de classification des lignes de corpus en deux classes (début de segment / non début de segment). Ce modèle de classification sera ultérieurement utilisé afin d'associer les lignes d'un nouveau corpus non expertisé à une des classes positive ou négative.

3 Acquisition des descripteurs

Nous cherchons à identifier des patrons syntaxiques qui différencient une ligne marquant le début d'une structure logique des autres lignes. La figure 1 représente un exemple de deux structures logiques dans les fichiers logs. Les lignes surlignées représentent le début de deux structures. De manière simple, nous pouvons caractériser le début de cette structure logique par un patron tel que "<- - -><string><fin :>" qui signifie *une ligne commençant par une série de "-", suivi par une chaîne de caractères fini par un " :"*. Nous considérons ce patron comme un *descripteur*. Pourtant, nous avons besoin d'un ensemble de descripteurs car un seul ne suffit pas pour caractériser pertinemment une division logique. Nous cherchons également à concevoir une méthode *automatique* d'acquisition des descripteurs. Ainsi, nous avons décidé d'utiliser les n-grammes pour caractériser le début des structures logiques. Dans le domaine du TAL

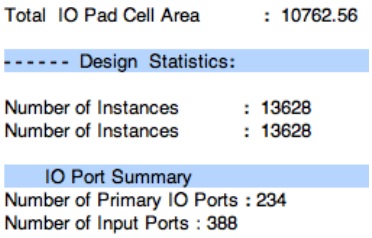

FIG. 1 – *Deux structures logiques dans un fichier logs.*

(Traitement Automatique du Langue), un n-grammes correspond à une série d'items dans un texte où les items peuvent être des lettres ou des mots. Les n-grammes sont souvent utilisés comme des descripteurs dans des tâches de classification de documents textuels Tan et al. (2002). Les n-grammes permettent de modéliser le *contenu* ainsi que *l'enchainement des mots* dans un document. Par exemple, en extrayant les tri-grammes (des lettres) dans la première ligne surlignée de la figure 1, nous obtenons "`---`", "`---`", "`␣De`", "`sig`", "`n␣S`", "`tat`", "`ist`", "`ics`", "`:`". *Or*, dans notre contexte, nous ne nous intéressons qu'à la *structure* des documents. Cela signifie que nous ne cherchons pas à identifier les structures logiques selon leur contenu (les mots ou les lettres) *mais* selon leurs structures textuelles (*les ponctuations, les symboles, les mises en page, etc.*). Cette nécessité nous a conduit à définir et proposer un *nouveau type original* de grammes que nous appelons *vs-grammes généralisés*. Ainsi, un vs-grammes est une série de caractères alphanumériques et non-alphanumériques défini de la manière suivante :

- si le gramme contient une série de caractères alphanumériques, il se termine par un caractère non-alphanumérique [1]. Le gramme suivant commence par le caractère non-alphanumérique.
- si le gramme commence par une série de caractères non-alphanumérique, il se termine par un caractère alphanumérique. Le gramme suivant commence par le caractère alphanumérique.

En prenant l'exemple précédent (la figure 1), nous obtenons les vs-grammes suivants sur le premier segment : "`------ D`", "`Design Statistics :`". Contrairement aux tri-grammes précédemment extraits, les vs-grammes modélisent bien la composition des caractères dans cette ligne. Autrement dit, ces vs-grammes expriment une composition de caractères telle qu'une chaîne de "-" suivie par une lettre et une chaîne de lettres finie par un " **:**", ce qui marque le début d'une structure logique dans les fichiers logs. Les vs-grammes sont toujours sensibles au contenu des textes. Par exemple, ici, le deuxième vs-gramme extrait, présente une série de lettres *composée* des mots "`Design`" et "`Statistics`". Or, la connaissance essentielle à prendre en compte est la présence d'une chaîne de lettres. De la même manière, le nombre de "-" dans l'autre vs-grammes n'est pas informatif. C'est la raison pour laquelle nous généralisons les vs-grammes en remplaçant les suites de lettres et de symboles par leurs équivalents en expression régulière. Ainsi, sur cet exemple, nous obtenons les vs-grammes généralisés suivants : "\-+ \w+", "\w+ **:**".

1. Les espaces s'ajoutent systématiquement aux grammes.

Nous constituons l'ensemble des descripteurs en extrayant des vs-grammes généralisés dans une fenêtre de lignes autour des lignes marquant le début des structures logiques. En prenant le premier segment de la figure 1, nous obtenons les descripteurs suivants par l'extraction des vs-grammes généralisés : D_1(\-+ \w+, 0), D_2(\w+ **:**, 0), D_3(\w+ \s+ **:**, -2) et D_4(**:** \w+, -2). Pour chaque descripteur, le chiffre après le patron représente le numéro de ligne dans la fenêtre. Le zéro correspond à la ligne même du début de segment. Nous procédons de la même manière pour les autres lignes marquant les DL afin de créer l'ensemble des descripteurs du corpus.

4 Expérimentations

Nous avons calculé la performance de notre approche en terme de précision et de rappel du modèle de classification. Le corpus d"apprentissage est constitué de 19 fichiers logs différents issus du monde industriel. Les fichiers logs contiennent des données réelles et sont différents en terme de contenu et surtout de structure. Le corpus d'apprentissage est de taille 1.1 Mo et contient au total 19638 lignes. Nous présentons ici les résultats obtenus en utilisant les algorithmes de classification qui ont donné les meilleurs résultats : l'arbre de décision C4.5 et KPPV. Pour appliquer les algorithmes, nous utilisons les implémentations intégrées au logiciel WEKA. Afin d'évaluer la performance de classification, nous utilisons la méthode de validation croisée (10 niveaux). Le tableau 1 présente la performance des classifieurs. Avec les

Classe	Précision	Rappel	F-Score
Pos	0.92	0.74	**0.82**
Neg	0.96	0.98	**0.97**

Classe	Précision	Rappel	F-Score
Pos	0.94	0.75	**0.84**
Neg	0.97	0.98	**0.97**

TAB. 1 – *Performance de classification en fonction de chaque classe - C4.5 (Gauche) et KPPV (Droite)*

KPPV, nous obtenons une précision égale à 0.94 dans la classe positive et égale à 0.97 dans la classe négative. Selon les résultats obtenus, nous argumentons que les vs-grammes généralisés représentent bien les caractéristiques syntaxiques des structures logiques d'un document.

5 Conclusions

Nous avons présenté une approche de segmentation des fichiers logs (les textes ayant des structures logiques complexes) fondée sur l'apprentissage supervisée. Dans notre approche nous avons constitué un ensemble de descripteurs où chacun présente une des caractéristiques syntaxiques des structures logiques. Afin de créer l'ensemble des descripteurs, nous avons proposé une méthode d'acquisition automatique des descripteurs qui utilise les vs-grammes généralisés présentés dans cet article. Nous avons réussi à construire un modèle de classification qui permet de distinguer les structures logiques dans les fichiers logs avec un F-Score égal à 0.99. Ces résultats ont conforté les experts à utiliser cette approche pour segmenter les fichiers logs.

Références

Callan, J. P. (1994). Passage-level evidence in document retrieval. In *Proceedings of SIGIR '94*, pp. 302–310. Springer-Verlag New York, Inc.

Kaszkiel, M. et J. Zobel (2001). Effective ranking with arbitrary passages. *J. Am. Soc. Inf. Sci. Technol. 52*, 344–364.

Pessiot, J.-F., M. Caillet, M.-R. Amini, et P. Gallinari (2004). Apprentissage non-supervisé pour la segmentation automatique de textes. In *CORIA*, pp. 213–228.

Tan, C.-M., Y.-F. Wang, et C.-D. Lee (2002). The use of bigrams to enhance text categorization. *Inf. Process. Manage. 38*, 529–546.

Tarek, O. (2003). La segmentation des documents techniques en amont de l'indexation : définition d'un modèle. *Revue d'Information Scientifique et Technique (RIST) vol. 13*(no1), 79–94.