Un outil de segmentation de courriels imbriqués en courriels individuels et en phrases

Ruslan Kalitvianski, Valérie Bellynck, Christian Boitet

Un outil de segmentation de courriels imbriqués en courriels individuels et en phrases

Ruslan Kalitvianski[*,**], Valérie Bellynck[*]
Christian Boitet[*]

[*]LIG-GETALP, bâtiment IMAG, 700 avenue Centrale, 30841 Grenoble cedex 9
prénom.nom@imag.fr
[**]Viseo Technologies, 4 avenue du Doyen Louis Weil, 38000 Grenoble

Résumé. Nous décrivons le problème de la segmentation de courriels représentant des conversations, c'est-à-dire contenant des courriels cités. Nous présentons un outil, SegDoc, conçu pour segmenter de telles conversations en courriels individuels, puis en extraire les phrases. La méthode consiste à repérer les entêtes générés par les outils de messagerie, qui marquent les frontières entre les messages. Nous décrivons les difficultés liées au repérage de ces en-têtes, dont la forme et les langues présentent une variété considérable. Une solution fondée sur des heuristiques indépendantes de la langue est proposée et évaluée. La tâche de segmentation en phrases est également décrite et évaluée. SegDoc produit une sortie XML contenant la conversation ainsi segmentée et préparée pour des traitements automatiques subséquents.

Introduction

Le problème qu'on cherche ici à résoudre est de segmenter des courriels en fragments de différents niveaux de granularité en vue de les préparer à de l'extraction d'informations à partir du texte. En particulier, nous visons à identifier les tâches mentionnées dans les énoncés constituant ces courriels, ainsi que leurs caractéristiques et leurs relations, ainsi que le type pragmatique de ces énoncés (annonce, demande, rappel, etc.). Dans la plupart des outils de messagerie, lorsqu'on répond à un courriel, ce courriel est placé en citation en dessous de la réponse qu'on rédige. Nous devons traiter ce type de suites de messages, et cherchons à les découper en messages individuels, puis en phrases.

Cet article décrit précisément la problématique de la segmentation de courriels, et présente un outil de segmentation multiniveau et multilingue de courriels, capable d'extraire les messages individuels à partir d'un seul fichier de conversation, puis d'extraire les phrases de ces messages et reconstituer le flux de la conversation.

Nous commençons par une présentation détaillée du problème, ensuite nous proposons une solution dans la partie 2, et l'évaluation dans la partie 3.

1 Les courriels et leur structure

Nous traitons des conversations sous forme de documents texte contenant des messages, en vue d'en extraire des informations relatives à des tâches auxquelles participent leurs auteurs. Ces messages sont souvent de la forme d'une séquence de courriels, donnée par ordre chronologique, chaque courriel en citant un autre, le précédant.

Bien que la plupart des outils de messagerie produisent des courriels en HTML, les documents qu'on traite sont purement textuels et ne contiennent pas de balises de structuration. L'exemple ci-dessous illustre le type de documents que nous voulons traiter :

```
Sujet : Re : Dossier
De : Michel Lefèvre
Date : 16/01/2011 23 :04
Pour : Yohann Treuillot <ytreuillot@gmail.fr>

Bonjour,
C'est fait.
Cordialement,
Michel

On 16 January 2016 at 13 :27, Yohann Treuillot <ytreuillot@gmail.fr> wrote :
> Bonjour, pourriez-vous faire suivre le dossier aux personnes concernées ?
> Merci !
> Cdlt,
> Yohann Treuillot

Michel Lefèvre - PhD
Assistant Professor, HDR
Laboratoire ABCD, Equipe EFGH - Tel : (+33) 4 00 00 00 00
```

TAB. 1 – *Une conversation à deux messages, sous forme de document textuel.*

Dans cette conversation exemple nous voyons :
- un en-tête récapitulatif sur plusieurs lignes qui commence le message ;
- un message cité, précédé par un en-tête sur une ligne, en anglais ;
- une signature qui suit le message cité.

Le message de Yohann est imbriqué dans le message de Michel, c'est-à-dire qu'il est précédé et suivi par du contenu d'un message d'un niveau hiérarchique supérieur. On le repère par la présence de caractères de citation '>' au début de chaque ligne.

A partir d'un telle conversation, nous voulons produire un fichier XML qui contient :
- la conversation telle quelle,
- les messages individuels extraits à partir de cette conversation, ordonnés de la même manière que dans la conversation, démêlés s'il y a entremêlage de segments ou imbrication d'un message dans un autre,

— pour chaque message, son en-tête, ainsi que les segments-phrases extraits étiquetés par leur langue et par un identifiant unique,
— dans les messages qui contiennent des fragments d'autres messages, ces fragments étiquetés par une référence vers le message extrait correspondant.

1.1 Les en-têtes

Pour segmenter une conversation en messages individuels nous devons nous baser sur des marqueurs de séparation entre les messages. Un de ces marqueurs est le caractère de citation '>'. Un autre est l'en-tête (sur plusieurs lignes).

Le rôle d'un en-tête est, au minimum, de rappeler l'expéditeur du message qui le suit. Dans la plupart des cas, les en-têtes indiquent aussi la date du message, et, dans le cas des en-têtes multiligne, l'objet et les destinataires.

L'en-tête est un marqueur important de séparation entre messages car, d'une part, le symbole de citation '>' n'est pas toujours présent dans les conversations, et d'autre part, des informations importantes et utiles pour la suite des traitements peuvent être extraites des en-têtes.

Nous distinguons deux types d'en-têtes : multiligne et monoligne. Dans l'exemple 1 ci-dessus nous avons un aperçu de chaque. L'exemple 2 ci-dessous montre 4 en-têtes multiligne (anonymisés) pour en illustrer la variété linguistique et structurelle.

Da : FR BORDEAUX Online [mailto :donotreply@carlsonwagonlit.com] Inviato : Martedì 12 gennaio 2016 09 :15 A : Jean Dupont <jean.dupont@mail.fr> Cc : service@carlsonwagonlit.com Oggetto : Confirmation/Invoice Réservation Hôtel Importanza : Grande
Sender : messages-noreply@bounce.linkedin.com From : John Doe via LinkedIn <member@linkedin.com> Reply-To : John Doe <John.Doe@gmail.com> To : My Name <My.Name@gmail.com>
Von : Pierre Dupond [mailto :Pierre.Dupond@mail.de] Im Auftrag von Jean Dupont Gesendet : Mittwoch, 12. Oktober 2016 20 :55 An : Int'l E-learning Association <elearning-list@ielas.com> ; elearning@ehub.no Betreff : Shared task proposition
Sujet : Acte2i : votre état des lieux (mission n°28544) Date de renvoi : Tue, 2 Jun 2015 09 :58 :38 -0700 De (renvoi) : jdupont@gmail.fr Pour (renvoi) : jean.dupont@mail.fr Date : Tue, 02 Jun 2015 18 :58 :34 +0200 De : Acte2i <nepasrepondre@acte2i.com> Pour : jdupont@gmail.fr

TAB. 2 – *4 en-têtes de structures et de langues différentes.*

Souvent, des champs sémantiquement identiques possèdent différents libellés dans une langue donnée, selon le client de courriel qui les génère. Par exemple, le champ ‘Date’ peut aussi apparaître comme ‘Envoyé’, ou bien ‘Sujet’ peut être ‘Objet’. L’ordre et le nombre des champs varie considérablement : rien que pour le français nous avons identifié 17 structures d’en-têtes différentes. Le format des dates présente aussi une certaine variabilité.

Le Tableau 3 ci-dessous présente des exemples d’en-têtes monoligne observés dans notre corpus. Ici aussi nous observons la variété linguistique et structurelle de ces en-têtes : présence ou non d’une « adressielle » [1] , d’un format de date (lorsqu’elle est présente), etc.

Il giorno 21 ott 2015, alle ore 14 :46, Jean Dupont <jean@mail.fr> ha scritto :
Вторник, 19 января 2016, 12 :03 +01 :00 от Jean D. <jean@mail.fr> :
Am 22.02.2016 um 13 :44 schrieb Jean Dupont :
10/26/2015 12 :18 PM(e)an, Jean Dupont igorleak idatzi zuen :
Le 06.03.2008 09 :38, le perspicace Jean DUPONT s’exprimait en ces termes :
Jean Dupont wrote :

TAB. 3 – *Exemples d’en-têtes monoligne.*

1.2 L’entremêlement de messages

L’Exemple 1 illustre une autre caractéristique des correspondances par courriel, l’entremêlement de messages. A cause de cela, si l’on veut segmenter le texte d’un courriel en messages individuels, on ne peut pas se contenter de le couper en deux fragments au niveau de l’en-tête monoligne. Un outil de segmentation doit tenir compte de cet aspect pour pouvoir reconstruire les messages aussi correctement que possible. En pratique, lorsqu’il y a entremêlement nous pouvons identifier les messages de différents niveaux par la présence de caractères de citation.

Dans certaines conversations de notre corpus, ces caractères ne sont pas présents, mais en général ces conversations ne sont que des suites de messages sans imbrication, qui peuvent donc être découpés uniquement à l’aide d’en-têtes.

1.3 Le multilinguisme

Les conversations sont souvent multilingues : parfois des messages contiennent du texte dans des langues différentes, parfois les messages entiers d’une conversation diffèrent par leurs langues. Pour produire une segmentation correcte nous devons donc identifier la langue du texte, car les conventions de ponctuation et d’abréviation, qui influent sur la segmentation en phrases, varient d’une langue à l’autre.

1. Nous utilisons ce néologisme bien formé, de préférence à « adresse de courriel », trop lourd, et à l’horriblement mal formé « adresse de mél » — en effet, la phonétique et l’orthographe françaises interdisent absolument une terminaison en « -él ». C’est d’ailleurs pour cela qu’on a les termes « péritel », « minitel », et pas *péritél, *minitél.

2 Travaux antérieurs

Nous avons identifié dans la littérature un travail antérieur, *Zebra* de Lampert et al. (2009), qui vise à identifier dans les messages les « zones fonctionnelles ». Le système utilise un ensemble de traits structurels de surface pour apprendre à classifier les lignes ou des fragments des courriels en neuf catégories : *Author, Signature, Disclaimer, Advertising, Greeting, Signoff, Reply, Forward, Attachment*. Les catégories *Reply* et *Forward* sont d'intérêt pour nous, puisqu'elles correspondent aux fragments cités, et permettent de distinguer le message le plus récent des précédents.

Zebra ne vise pas à identifier les en-têtes, utiles pour nous car ils contiennent des informations nécessaires pour les traitements du message segmenté subséquents. De même, il n'y a pas d'analyse de la structure de zones étiquetées comme *Reply* ou *Forward*, et donc pas de décomposition de ces zones en messages individuels, même si leurs résultats d'identification de lignes correspondants à ces catégories sont très bons (F-mesures de 91% et 89% respectivement).

3 Proposition de solution

3.1 Segmentation en messages individuels

La segmentation en messages individuels est guidée par le repérage des en-têtes des messages et par les caractères d'imbrication. Dans les lignes du texte qui commencent par des caractères d'imbrication, nous associons un niveau à chaque ligne correspondant au nombre de ces caractères d'imbrication en début de ligne. On part de l'hypothèse qu'entre deux en-têtes toutes les lignes d'un même niveau appartiennent au même message.

Pour détecter les en-têtes des messages quelles que soient leurs langues et leurs structures. Pour cela, nous devons identifier des traits discriminants linguistiquement invariants.

Nous repérons les en-têtes à l'aide d'expressions régulières couplées à une recherche de de traits. Les expressions en elles-mêmes sont relativement peu discriminantes, et servent seulement à trouver des zones candidates.

Pour les en-têtes multiligne on cherche des blocs de trois à sept lignes, chaque ligne commençant par un à sept mots, suivis du caractère ' :', suivi lui-même par d'autres mots et enfin par un passage à la ligne.

Nous associons un score, valant 0 au départ, à chacun de ces blocs. Ce score est calculé en cherchant la présence :

— d'une année ou d'une heure (les mois étant souvent écrits avec des mots, cette information dépend donc de la langue) ;
— d'une adressielle ;
— d'un changement de niveau d'imbrication (nombre de caractères '>' commençant une ligne) entre la ligne qui précède le bloc et la ligne qui le suit ;
— de possibles noms de personnes, en détectant des suites de mots commençant par des majuscules ;
— de chaînes « re : » ou « fwd : », indicatrices d'un champ « objet » ;
— d'une ligne immédiatement précédente qui contiendrait une séquence de '—', comme dans «——— Message original ———».

Pour chaque trait présent nous incrémentons le score de 1. Si le score atteint un certain seuil (2 ou plus), nous considérons que la zone est bien un en-tête. Nous présentons dans la partie 3 une évaluation quantitative de cette approche pour le repérage.

Il est difficile de s'abstraire entièrement de la langue, car les deux derniers traits ne sont pas tout à fait indépendants de la langue : il n'y a pas de distinction entre majuscules et minuscules dans les scripts coréen, japonais et chinois, et nous ne savons pas si « re : » et « fwd : » sont des préfixes universels.

Pour les en-têtes monoligne nous cherchons des lignes contenant une année ou une heure, et qui se terminent par un ' :'.

3.2 Segmentation en phrases

Une fois que les messages ont été identifiés et « démêlés » si nécessaire, nous effectuons la segmentation en phrases, en trois étapes. D'abord, nous identifions les paragraphes dans le texte, séparés par des passages à la ligne. Ces paragraphes sont soumis à l'identification de la langue par l'outil Apache Tika [2]. Cela nous permet de choisir le bon jeu de règles de segmentation en phrases.

Avant d'appliquer les règles, nous passons par une étape de normalisation afin de minimiser les erreurs de segmentation. A cette étape on remplace les émoticônes (ou *smileys*) par des hors-textes spéciaux, car la ponctuation contenue dans les émoticônes peut causer une fausse segmentation. On détecte aussi des phrases tronquées par des passages à la ligne : certains messages peuvent avoir été encodés en *quoted-printable* [3] ou en MIME [4], des encodages qui limitent la longueur des lignes à 76 ou 78 caractères, insérant des passages à la ligne en cas de dépassement. Le passage à la ligne étant un marqueur potentiel de fin de phrase, nous voulons éliminer ceux qui n'en sont pas. Par conséquent si une ligne est d'une longueur entre 75 et 78 (sans compter les caractères de citation) et ne se termine pas par une ponctuation de fin de phrase, alors nous remplaçons le caractère de passage à la ligne par un espace.

Chaque paragraphe est segmenté en phrases en utilisant des règles SRX [5]. Les SRX sont des règles de segmentation en phrases, basées sur des expressions régulières qui décrivent les conditions sur le contexte d'une position dans le texte pour la classer ou non comme frontière de phrases. Lors de la segmentation, les règles sont appliquées en cascade à chaque position dans le texte, jusqu'à ce qu'une des expressions concorde. L'exemple ci-dessous illustre le principe et la syntaxe :

Les règles SRX ont l'avantage d'être interopérables et faciles à produire, contrairement à des modèles appris par apprentissage automatique. Il existe déjà de nombreux jeux de règles pour différentes langues, produites par la communauté (Miłkowski et Lipski (2009)). Nous disposons de règles SRX pour 20 langues, ce qui suffit largement pour nos applications.

2. https://tika.apache.org/
3. https://tools.ietf.org/html/rfc2045
4. https://tools.ietf.org/html/rfc2822
5. *Segmentation Rules eXchange :* https://www.gala-global.org/srx-20-april-7-2008

<rule break="no"> <before-break>\b[Mm]lle\.</beforebreak> <afterbreak>\s</afterbreak> </rule>	Cette règle indique que si l'analyse se situe à un endroit dans le texte qui est précédé par « Mlle. » ou « mlle. » et suivi par un espace, alors ce n'est pas une frontière de phrases.

TAB. 4 – *Règle de segmentation pour « Mlle. » ou « mlle. »*

4 Évaluation et résultats

Nous avons développé antérieurement l'outil SegDoc pour segmenter des documents de divers formats, en unités pouvant varier par la taille et la structure. Dans le cadre de ce travail, nous l'avons adapté pour segmenter des courriels en messages individuels, en phrases, puis pour produire une sortie XML qui contient la conversation, les messages individuels (démêlés lorsqu'il y a eu entremêlement), leurs en-têtes, et les segments-phrases associés à chaque message.

Nous évaluons sa performance d'abord sur un corpus de courriels majoritairement en français, mais contenant environ 4% de texte anglais, et ensuite sur une portion du corpus EnronSent [6] Styler (2011).

4.1 Évaluation du repérage d'en-têtes

Nous avons constitué un corpus de correspondances dans lequel nous avons manuellement identifié 495 en-têtes. En appliquant nos règles de repérage, sur 495 en-têtes annotées, 453 ont été correctement repérées. 25 fragments de texte ont été incorrectement classés comme en-têtes.

	Précision	**Rappel**	**F-mesure**
Repérage d'en-têtes	95%	91%	93%

TAB. 5 – *Résultats pour le repérage des en-têtes dans notre corpus.*

La plupart des faux négatifs pour les en-têtes multiligne étaient dus à des déformations induites par le passage du message par plusieurs clients de courriel. Cela a parfois pour effet d'ajouter des passages à la ligne, et parfois, au contraire, d'en supprimer. Dans tous les cas, le formatage sur lequel se basent nos heuristiques de repérage de blocs avait été altéré.

4.2 Évaluation du démêlage

Parmi les 201 conversations dans le corpus annoté, nous avons identifié 78 conversations contenant un entremêlement. SegDoc a correctement démêlé 59 de ces conversations. Dans

6. http://savethevowels.org/enronsent/

trois cas, SegDoc n'a pas détecté l'entremêlement car les caractères de citation n'étaient pas '>' mais une suite d'espaces. Dans les cas restants, SegDoc n'a pas rattaché les fragments aux bons en-têtes. De même, pour 5 conversations qui ne contenaient pas d'entremêlement, mais dont des caractères de citation avaient été localement altérés par des reformatages induits par des outils de messagerie, SegDoc a effectué un démêlage inutile qui a rattaché des fragments à de mauvais en-têtes.

Il nous paraît difficile d'améliorer ces résultats grandement, puisque cette opération est guidée par les caractères de citation, qui sont parfois appliqués de manière inconsistante, surtout lorsque la conversation passe par plusieurs clients de messagerie différents. Les erreurs de démêlage ne rajoutent cependant pas d'erreurs à la segmentation en phrases.

4.3 Évaluation de la segmentation en phrases

Nous évaluons la segmentation en phrases sur un corpus de 6994 segments issus de notre corpus, majoritairement en français (279 segments en anglais), et sur un corpus contenant 1000 segments issu du corpus EnronSent, tous en anglais. Nous mesurons le nombre de phrases correctement identifiées.

Nous évaluons notre outil contre NLTK (Bird et al. (2009)) et LingPipe (Alias-I (2008)). Le texte qu'on utilise est celui des messages, sans les caractères de citation, sans autre normalisation (chaque outil est supposé faire ensuite sa propre normalisation).

	Corpus-FR			**EnronSent**		
	Précision	**Rappel**	**F-mesure**	**Précision**	**Rappel**	**F-mesure**
SegDoc	85%	83%	84%	83%	81%	82%
NLTK	83%	81%	82%	87%	84%	85%
LingPipe	81%	77%	79%	88%	80%	84%

TAB. 6 – *Résultats pour le repérage de phrases dans des courriels*

Nous voyons dans le tableau 6 que les trois systèmes produisent des performances comparables, avec un léger avantage pour SegDoc pour le français, pour NLTK pour l'anglais.

Sur les messages en anglais de notre corpus, SegDoc obtient une F-mesure de 86%, contre 88% pour NLTK et 86% pour LingPipe.

Les règles de segmentation de SegDoc ayant été développées pour traiter des pages Web (pour de la traduction automatique), ces scores sont plus bas que ceux qu'on obtient sur les pages Web. Dans une évaluation en interne faite précédemment sur un ensemble de 20 pages, contenant 788 segments en anglais, SegDoc obtenait une F-mesure de 91%, dépassant la segmentation de Google Translate (89%) que nous utilisions. Dans le cas des courriels le texte est bruité de manière intrinsèque (mauvaises ponctuations et capitalisations, présence d'éléments non-textuels) et extrinsèque (formatages induits par les clients de messagerie).

Dans leur évaluation de neuf systèmes de segmentation de texte anglais non balisé (Read et al. (2012)) ont montré que, d'une part, aucun outil de segmentation n'est fiable à 100 %, et que, d'autre part, les performances relatives des systèmes de segmentation sur un corpus donné n'étaient pas toujours constatées sur un corpus différent, y compris après normalisation des corpus et adaptation des segmenteurs.

Conclusion et perspectives

Nous avons abordé le problème de la segmentation de courriels en vue de les préparer à de l'extraction d'informations. Nous avons proposé une méthode pour identifier et extraire les messages cités dans d'autres messages, et en extraire des phrases. Notre outil SegDoc développé antérieurement a été étendu et adapté pour effectuer ce traitement et produire une sortie XML qui contient la conversation, les messages individuels, leurs en-têtes, et les phrases associés à chaque message. L'outil continue d'être développé et sera prochainement publié en source ouvert.

Bien qu'aujourd'hui SegDoc utilise des SRX comme outil principal de segmentation en phrases, il est conçu pour pouvoir facilement intégrer d'autres outils de segmentation. Une amélioration envisageable serait de faire appel à plusieurs systèmes de segmentation pour produire un graphe de segmentation, puis d'utiliser un modèle de langue pour pondérer les trajectoires dans le graphe et retenir celle qui a le plus grand poids.

Nous prévoyons d'améliorer le repérage d'en-têtes en utilisant de l'apprentissage automatique et d'apprendre à détecter les zones inutiles pour notre traitement, comme les signatures, avec la méthode de segmentation en zones fonctionnelles, telle que décrite dans Lampert et al. (2009).

Références

Alias-I (2008). Lingpipe 4.1.0. `http://alias-i.com/lingpipe/`.

Bird, S., E. Loper, et E. Klein (2009). *Natural Language Processing with Python*. San Mateo: O'Reilly Media Inc.

Lampert, A., R. Dale, et C. Paris (2009). Segmenting email message text into zones. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, Stroudsburg, PA, USA, pp. 919–928. Association for Computational Linguistics.

Miłkowski, M. et J. Lipski (2009). Using srx standard for sentence segmentation in languagetool. In Z. Vetulani (Ed.), *Human Language Technologies as a Challenge for Computer Science and Linguistics*, pp. 556–560. Wydawnictwo Poznańskie, Fundacja Uniwersytetu im. A. Mickiewicza.

Read, J., R. Dridan, S. Oepen, et L. J. Solberg (2012). Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, Mumbai, India, pp. 985–994. The COLING 2012 Organizing Committee.

Styler, W. (2011). The enronsent corpus. Technical report 01-2011, University of Colorado at Boulder Institute of Cognitive Science, Boulder, CO.

Summary

We describe the problem of segmentation of email messages that represent conversations, that is containing cited messages. We present SegDoc, a tool designed to segment such conversations in individual messages, and then extract sentences. The method consists in detecting

headers generated by messaging tools, that indicate boundaries between messages. We describe difficulties related to the detection of these headers, the form and language of which vary considerably. A solution based on language-independent heurisitics is proposed and evaluated. The sentence segmentation task is also described and evaluated. SegDoc produces an XML output containing the segmented conversation, prepared for subsequent machine processing.