

Kwaga : une chaîne UIMA d'analyse de contenu des mails

Proposition de démonstration

Philippe Laval, Frédéric Meunier, Gaëlle Recourcé, Sylvain Surcin,

Kwaga, 20 rue Raymond Marcheron 92170 Vanves
{laval, meunier, recource, surcin}@kwaga.com

1 Introduction

Kwaga a pour ambition de mettre en ligne une solution mettant en œuvre une chaîne de traitement des mails visant à faciliter la gestion du courrier électronique en affichant pour chaque message une icône reflétant les décisions à prendre par le destinataire (niveau d'urgence, demande d'action, invitation...) ainsi qu'à fournir un ensemble de raccourcis pour les actions associées (ajout de coordonnées dans le carnet d'adresses, intégration d'un rendez-vous dans le calendrier...).

Cette chaîne de traitement doit répondre aussi bien à des exigences de qualité (pertinence et couverture des résultats) qu'à des exigences purement techniques (performances). Nous montrerons en quoi le framework Apache UIMA est une infrastructure technique qui permet de couvrir ces deux axes d'exigences, souvent orthogonaux. Par ailleurs, nous décrirons comment nous avons contourné deux contraintes techniques introduites l'une par le standard UIMA lui-même, l'autre par le framework Apache UIMA.

2 Chaîne de traitement des mails

La chaîne de traitement des mails doit répondre à des exigences fonctionnelles et techniques pouvant s'avérer orthogonales. Nous avons choisi de décomposer la chaîne de manière suffisamment fine pour que l'assemblage des composants unitaires puisse couvrir l'ensemble de ces exigences, qui sont essentiellement focalisées sur la qualité de l'analyse du discours et l'ouverture nécessaire au multilinguisme.

Nous avons opté pour l'assemblage de composants de traitement du langage naturel, chacun se focalisant sur son périmètre d'analyse. Nous nous sommes orientés vers UIMA qui s'appuie sur une architecture pilotée par les données¹, permettant une interchangeabilité accrue. Ainsi, pour chaque langue nous pouvons établir une chaîne de traitement idoine tout en conservant l'architecture technique. Par ailleurs, l'implémentation par Apache du standard UIMA, apporte des garanties techniques qui

¹ *Model Driven Architecture*

2 Kwaga : une chaîne UIMA d'analyse de contenu des mails

couvrent nos exigences : la volumétrie, la disponibilité, l'intégrité des données et la sécurité. En effet, les chaînes de traitements Apache UIMA peuvent être déployées dans un conteneur Web, par exemple Tomcat avec un serveur Web Apache en frontal.

En amont de la chaîne de traitement des mails, nous avons le composant d'aspiration des mails (Importer), et, en aval, le composant offrant un service Web accessible par les clients de messagerie des utilisateurs de Kwaga. Ces deux composants s'appuient sur le framework AWS² d'Amazon, avec S3 pour la mise en persistance d'objets non structurés, SimpleDB pour la mise en persistance et la recherche d'objets structurés et SQS pour la gestion de la chaîne de traitement globale (Business Process Management).

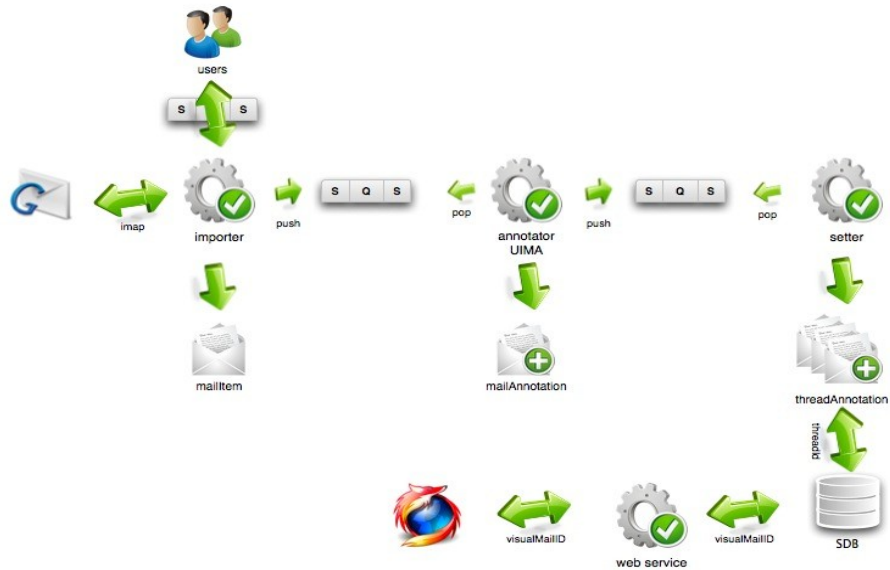


Fig. 1. Schéma général

3 Intégration des composants linguistiques via UIMA

3.1 Structuration des mails

La diversité des fournisseurs de messagerie et des standards fait du pré-traitement une opération délicate mais déterminante. Il faut tout d'abord analyser l'enveloppe du mail afin d'en extraire toute l'information structurée possible (expéditeur, destinataire(s), sujet, date d'envoi, route d'acheminement, nombre de parties, encodage des parties, identifiant du mail, etc.). Ces informations sont extraites par l'Importer et constituent l'entrée de la chaîne d'analyse.

² Amazon Web Services

L'étape suivante est l'analyse de la structuration interne des mails : dans le cas de mails résultant d'un « *reply* » ou d'un « *forward* », il est important de détecter le corps principal du mail, c'est-à-dire le texte réellement ajouté par le dernier expéditeur, que l'on appellera « Real Mail Body ». Ce composant permet aussi de repérer les zones ajoutées par les différents intermédiaires comme les footers (« *No virus found* » ou « *XXX Mailing List* »). Enfin, pour l'ensemble des composantes de la « discussion », il faut procéder à l'identification du format (texte ou HTML) et du codage des caractères (ISO-Latin, UTF-8, etc.), et convertir l'ensemble dans le codage que nous utiliserons systématiquement : l'UTF-8. La langue du message est par ailleurs déterminée (anglais ou français pour l'instant).

Le mail ainsi analysé est alors chargé dans un CAS (*Common Analysis System*) UIMA contenant d'une part l'ensemble des parties dédoublonnées du mail (hormis les attachements), l'enveloppe du mail sous forme de traits structurés et les différentes informations extraites sous forme d'annotations typées.

3.2 Analyse linguistique

L'étape suivante consiste à analyser le Real Mail Body. L'analyse linguistique se décompose en trois phases : une phase de pré-traitement par expressions régulières en amont de la tokenisation, une phase de génération dynamique de dictionnaire, puis l'analyse linguistique proprement dite par application d'automates à états finis (transducteurs) utilisant le composant open-source Unitex développé à l'université Marne-la-Vallée³.

Le pré-traitement repère les entités simples (dates machine, URLs, numéros de téléphone et adresses email) grâce à un analyseur spécialisé dans l'application d'expressions régulières. Un autre analyseur crée ensuite pour chaque entité une entrée de dictionnaire au format DELAF, en lui attribuant une information sémantique correspondant à sa classe.

L'analyse linguistique elle-même est réalisée par l'application de transducteurs Unitex, en faisant appel au dictionnaire créé à l'étape précédente, pour le repérage des indices linguistiques (indice d'action, de demande de rendez-vous...). Ces indices permettront d'amorcer l'identification des catégories de mails (Demande d'Action, Prise de rendez-vous...) dans la phase suivante.

Chaque transducteur est appliqué sur le texte découpé en phrases, tokenisé et étiqueté. Les éventuels chevauchements entre les sorties des transducteurs sont gérés au niveau de l'encapsulation UIMA. Des annotations spécialisées sont créées en fonction des types d'indices linguistiques détectés (par exemple pour interpréter des informations temporelles, une classe dérivée de celle des annotations d'indices linguistiques permet de stocker la représentation iCal d'un événement).

3.3 Interprétation en catégories

La dernière étape de cette chaîne d'analyse est un moteur d'interprétation qui a pour but d'associer à chaque mail une catégorie caractérisant son contenu en fonction d'une

³ http://www-igm.univ-mlv.fr/~unitex/why_unitex.html#resources

part des méta-données (courrier entrant ou sortant par exemple) et de l'autre des indices linguistiques identifiés dans l'étape précédente.

L'interprétation des indices linguistiques et des méta-données est réalisée en appliquant deux séries de règles. La première consiste en un ensemble d'heuristiques qui déterminent chacune zéro, une ou plusieurs catégories candidates possibles en fonction des indices linguistiques détectés et des méta-données du mail. Un second jeu d'heuristiques permet ensuite de sélectionner une seule catégorie pour le mail et/ou la conversation en fonction du client mail concerné. Ces heuristiques ne font plus référence aux informations linguistiques et travaillent uniquement à un niveau conceptuel « *métier* ». Les informations associées sont intégrées dans un formulaire propre à chaque catégorie (personnes, lieu et date pour un rendez-vous par exemple). Chaque catégorie contient en outre une liste chaînée de références aux annotations d'indices linguistiques afin de générer un texte comme résumé dans l'interface.

4 Organisation du Type System

4.1 CAS avec Sofa multiple

Le mail subit plusieurs transformations dans notre chaîne de traitement, par exemple, l'identification du « Real Mail Body » qui fournit en sortie les éléments de texte effectivement ajoutés par le rédacteur du mail. Ces éléments ne sont pas forcément contigus en particulier lors d'un « *reply* » où des réponses sont insérées dans le mail original. Nous avons choisi d'utiliser la possibilité de créer des vues (Sofa) multiples pour un même CAS.

Par ailleurs, Unitex modifie l'objet d'étude avant d'appliquer les transducteurs, en procédant à une normalisation du texte. Or en UIMA, il est impossible de modifier le contenu textuel d'un CAS une fois initialisé. Une solution est de créer une nouvelle vue et de l'insérer dans le CAS et de remplir cette vue du texte normalisé, en établissant un lien entre les Tokens de celle-ci vers les Tokens annotés dans la vue initiale⁴. Ceci nous permet en fin de traitement de repérer dans la vue initiale les parties du mail pertinentes à présenter à l'utilisateur.

Un autre atout de cette solution est la possibilité d'extension à de nouveaux formats : par exemple, nous ne travaillons actuellement que sur la Mime Part textuelle du mail, mais nous avons l'intention d'étendre la couverture à la Mime Part HTML (quand elle est disponible). Unitex ne supporte pas le HTML en natif, mais nous intégrerons un « déHTMLiseur » et les vues multiples avec référence entre Tokens permettront de surligner directement dans le mail HTML les portions de texte pertinentes extraites par Unitex.

4 cf. la section 6.6 de *UIMA Tutorial and Developers' Guides*, version 2.2.1-incubating.

4.2 Listes chaînées de Features

Nous avons été confrontés à une limitation du framework Apache UIMA, qui ne garantit pas un comportement stable d'un FSIterator dans le cas d'annotations « ambiguës ». En effet, si deux annotations ont le même début et la même fin, l'Iterator cherche à comparer les types des deux annotations. S'ils sont différents, il consulte dans le Type System Descriptor les priorités qui y sont précisées. Si ces deux types ne sont pas comparables ou égaux alors l'Iterator a un comportement aléatoire allant régulièrement vers une *OutOfMemoryError*⁵.

Pour éviter cette situation, nous utilisons la possibilité d'avoir des traits multivalués, ainsi une unique annotation peut décrire une ambiguïté non levée ou introduite par un analyseur. Cependant, les classes proposées par le framework Apache UIMA (*StringArray* et *FSArray*) sont lourdes à utiliser, car la taille de leurs instances est fixe et il est difficile d'y ajouter un élément au cours de l'analyse. Nous avons pour cette raison opté pour des listes chaînées non bornées pour les annotations avec des traits multivalués.

5 Conclusion

Nous avons décrit ici une application mettant en œuvre une chaîne de traitement des mails et de leur contenu. Le choix du standard UIMA et de son implémentation par Apache nous permet aujourd'hui d'isoler, d'une part, les composants d'analyse successifs, et, d'autre part, le code métier de l'application, tout en nous appuyant sur une infrastructure technique capable de répondre à nos exigences non fonctionnelles (performances, disponibilité, intégrité, sécurité). Les difficultés lors du développement (figement du contenu textuel d'un Sofa et traitement des ambiguïtés) ne se sont pas avérées bloquantes. Les contournements que nous avons mis en place ont été possibles grâce à l'ouverture du standard UIMA et de son implémentation par Apache UIMA. Ce type d'expérimentation positive permet également un gain important de productivité en termes de développement logiciel. Le framework Apache UIMA constitue en cela une excellente solution d'intégration robuste de composants TAL pour réaliser une chaîne de traitement de documents structurés et non-structurés. Cette expérimentation devrait encourager à multiplier le choix d'UIMA pour la mise à disposition de composants d'analyse linguistique.

⁵ cf. la documentation de l'API, pour la méthode *subiterator(AnnotationFS annot)* de la classe *AnnotationIndex*.