NaviQuest : un outil pour naviguer dans une base de questions posées à un Agent Conversationnel

Ph. Suignard[1]
Philippe.Suignard@edf.fr

[1]EDF R&D
1, avenue du Général de Gaulle
92 141 Clamart Cédex – FRANCE

Résumé :
Cet article présente NaviQuest un outil mixant des techniques de visualisation et d'analyse textuelle pour naviguer dans une base de questions posées par les internautes à un agent conversationnel.
Mots-clés : Agent Conversationnel, Text mining, Visualisation

Abstract :
This paper presents a tool called NaviQuest, mixing visualization techniques and textual analysis, to browse a database of questions from users to a conversational agent.
Keywords: Conversational Agent, Text Mining, Vizualisation

1 Introduction

Présente depuis plusieurs années sur le site Bleu Ciel d'EDF[1], Laura est un agent-conversationnel qui répond aux questions posées par les internautes. Laura est réalisée par les sociétés Sémantia[2] pour la partie question-réponse et Cantoche[3] pour l'animation du personnage. Sa base de connaissances métier est réalisée par EDF.

En plus des rapports mensuels fournis par Sémantia, nous avons voulu analyser finement les questions posées à Laura dans un but d'amélioration : savoir ce que disent les clients, comment ils le disent et comment ces demandes évoluent avec le temps. Toutes ces questions constituent une source d'information importante pour connaître les préoccupations des clients, démarche qui peut se rapprocher du « crowdsourcing[4] ».

Pour cela, nous avons voulu mixer des techniques de visualisation et d'analyse textuelle, ce qui constitue une préoccupation actuelle comme en témoigne la tenue du premier workshop[5] sur le thème des « interfaces visuelles intelligentes pour le texte » ou les sessions « Text Visualization » des conférences InfoVis[6] et « Text Analytics » des conférences Vast[7]. Tous ces efforts correspondent au besoin croissant d'analyser des données textuelles issues de mails, blogs, forum, twitter et autres.

En effet, plusieurs approches existent pour analyser ou fouiller ce type de données. D'un côté, il existe des logiciels d'analyse lexicométrique comme Lexico [8], initialement fait pour analyser des œuvres littéraires. De l'autre côté, se trouvent des logiciels fortement interactifs (pour naviguer dans les données afin de mieux les comprendre) comme Jigsaw [9], PosVis [11], ou encore Harvest [6].

[1] - http://bleuciel.edf.com

[2] - http://www.semantia.com/

[3] - http://www.cantoche.com/

[4] - Le crowdsourcing est un néologisme créé en 2006, pouvant être traduit par « approvisionnement par la foule » http://fr.wikipedia.org/wiki/Crowdsourcing

[5] - IVITA, First International Workshop on Intelligent Visual Interfaces for Text Analysis, Hong-Kong, Chine, Février 2010.

[6] - IEEE Information Visualization Conference (IEEE InfoVis)

[7] - IEEE Symposium on Visual Analytics Science and Technology (IEEE VAST)

Et enfin, il existe des bibliothèques graphiques ou des techniques de visualisation, plus ou moins adaptées à chaque domaine, plus ou moins généralistes comme ManyEyes [10] ou Prefuse [7].

La suite du document présente le logiciel NaviQuest, inspiré par les logiciels ou techniques précédentes et adapté à nos besoins ainsi que ses principales fonctionnalités accompagnées de quelques résultats.

2 Architecture du logiciel

Pour développer NaviQuest, nous sommes repartis d'un logiciel développé dans le cadre du projet Callsurf [5] [2] qui permet d'enregistrer des conversations téléphoniques entre des clients et des conseillers en ligne, de les transcrire automatiquement, et d'offrir une interface de navigation dans ces conversations.

Bien évidemment, un dialogue entre un client et un conseiller est très différent d'un dialogue entre un internaute et Laura, mais ils ont pour dénominateur commun d'être constitués d'une suite de tours de parole. Donc ce qui avait été développé dans le cadre de Callsurf a pu être adapté, pour les dialogues avec Laura.

Le logiciel est développé en Java. Il s'appuie sur plusieurs bibliothèques Open Source :

- Lucene[8] pour la partie moteur de recherche ;

- JFreeChart[9] pour la représentation sous forme de courbe ou histogramme ;

- JTreeMap[10] pour la représentation sous la forme de TreeMap.

Le logiciel se présente sous la forme de deux modules : le premier pour indexer les données et le second pour naviguer dans la base des questions-réponses.

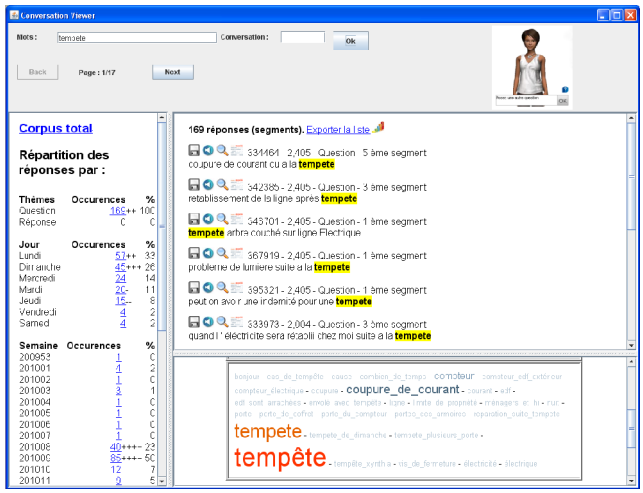

Fig. 1 : Copie d'écran de NaviQuest

Sur la fenêtre principale de l'application, se trouvent :

- En haut de l'écran, une barre de saisie pour rechercher un ou plusieurs mots ;

- A droite, la liste des questions posées à Laura contenant le ou les mots cherchés ;

- A gauche, la ventilation de ces questions selon les méta-données

- En bas à droite, un nuage de mots calculé à partir des questions posées à Laura et contenant le ou les mots cherchés.

3 Fonctionnalités du logiciel

L'indexation

Cette partie consiste à relire les fichiers de « log » de Laura et à les convertir en documents au sens de Lucene. Les questions sont indexées, les réponses ne le sont pas, elles sont juste sauvegardées dans la base Lucene. Cela signifie qu'on peut chercher un ou plusieurs mots dans les questions posées à Laura et qu'on peut voir la réponse fournie, mais qu'on ne peut pas chercher dans les réponses fournies. Un document sera constitué des éléments suivants :

- Le texte de la question posée par le client ou de la réponse fournie par Laura

- Nom du fichier (anecdotique)

- Type du tour de parole : question ou réponse

- Rang de la question/réponse : 1 pour la 1ère

[8] - http://lucene.apache.org

[9] - http://www.jfree.org/jfreechart/

[10] - http://jtreemap.sourceforge.net/

question, puis 2, etc.

- Jour de la semaine : de lundi à dimanche

- Nombre de mots de la question

- Numéro de la semaine

- Date de la question/réponse

Recherche « full text »

L'application comprend un moteur de recherche « full text » permettant de trouver toutes les questions contenant un mot donné, comme « tempête », par exemple. Grâce aux requêtes floues autorisées par Lucene (`FuzzyQuery`) on s'affranchit, dans une certaine mesure, des nombreuses erreurs de saisies[11] ou faute d'orthographe. L'application autorise de saisir des suites de mots comme « service relevé client » grâce aux `MultiTermQuery` ou `QueryPhrase`. Il est également possible, avec des requêtes booléennes `BooleanQuery`, de faire des requêtes plus sophistiquées[12].

Après la recherche dans la base de données, le logiciel affiche la liste des questions correspondantes avec mise en évidence du ou des termes recherchés :

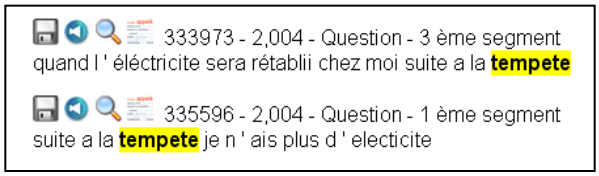

Fig. 2 : Surlignage de mot

En cliquant sur une des questions, on accède à l'ensemble du dialogue entre le client et Laura.

Ventilation des résultats de la requête

Au démarrage, le logiciel calcule la ventilation des questions selon toutes les méta-données disponibles (jour de la semaine, numéro de la semaine, etc.). Ensuite, pour chaque requête, il calcule la ventilation des réponses trouvées selon ces mêmes méta-données, ce qui lui permet de comparer les deux ventilations, et d'indiquer grâce à des signes + et - une sur ou sous représentation par rapport à l'ensemble du corpus.

Semaine	Occurences	%
200953	1	0
201001	4	2
201002	1	0
201003	3	1
201004	1	0
201005	1	0
201006	1	0
201007	1	0
201008	40++++	23
201009	85++++	50
201010	12	7
201011	9	5
201012	5	2
201013	5	2

Fig. 3 : Ventilation des requêtes par semaines

Pour une recherche avec le mot « tempête », on constate une très forte sur représentation en semaine 8 et 9, d'où la présence de signes ++++.

Raffinement des recherches

L'application permet de raffiner une recherche en spécifiant une ou plusieurs méta-données. Sur l'exemple précédent, il est possible de rechercher le mot « tempête » en semaine 1 pour s'apercevoir qu'il s'agit de problèmes relatifs à une tempête habituelle, alors qu'en semaine 8 et 9, il s'agit de questions relatives à la tempête Xinthia qui a frappé le pays entre le 26 février et le 1er mars, à cheval sur les semaines 8 et 9.

Analyses temporelles

Les mêmes meta-données vont permettre de mesurer la dynamique des questions et d'afficher leur occurrence selon les jours ou semaines, soit de manière absolue, soit relative au nombre de questions posées pendant une période donnée.

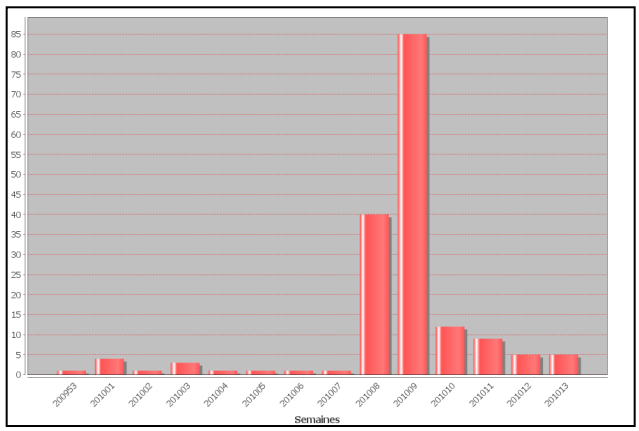

Fig. 4 : Distribution temporelle des questions

[11] - Pour le mot « tempête » par exemple, on trouve : « tempoet », « tepete », « tempëte », « tempêtes », « tempête », « tempéte », « tempete ».

[12] - Mots obligatoires (`BooleanQuery.MUST`), mots à éliminer (`BooleanQuery.MUST_NOT`) ou mots requis (`BooleanQuery.SHOULD`).

L'histogramme correspondant au mot « tempête » montre bien l'apparition soudaine de ce mot en semaine 8 et 9 puis sa lente décroissance. Cette analyse permet de voir qu'un mot a tendance à être cité de manière plus importante à certaines périodes.

Nuage de mot ou « Tag Cloud »

L'invention des « Tag Cloud » ou nuage de mots est attribuée, selon Wikipedia[13], à D. Coupland [3], mais leur utilisation a été popularisée par le site Web de photos Flickr[14]. Pour chaque résultat de la requête, NaviQuest calcule un « nuage de mots » à partir de la liste des mots encadrant le ou les mots recherchés. La liste des mots trouvés (unigramme, bigramme ou trigramme) est visualisée sous la forme d'un nuage avec un code couleur et une police de caractère dépendant de la fréquence des mots : plus le mot ou la séquence de mot est fréquent[15] plus sa police est grande et sa couleur est rouge.

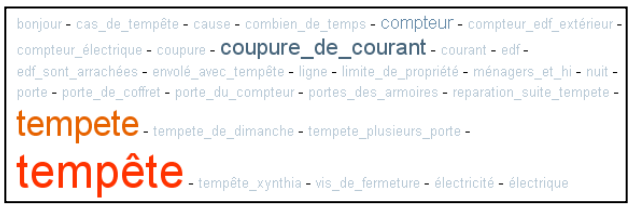

Fig. 5 : Nuage de mots associés à la requête « tempête »

Cette représentation a l'avantage de faire apparaître des associations de mots ou concepts, plus ou moins évidentes, comme ici : « coupure de courant », « porte » (de compteur, d'armoire, de coffret) arrachées ou envolées, « tempête de dimanche », « tempête Xinthia »…

Détection de nouveauté et TreeMap

Une manière d'explorer le corpus consiste à détecter les mots nouveaux ou qui évoluent fortement (à la hausse ou à la baisse) d'une semaine sur l'autre. L'algorithme utilisé est le suivant : il calcule la liste de tous les mots (unigramme, bigramme ou trigramme) utilisés pour une semaine calendaire donnée. Ensuite, il calcule un score pour chacun d'eux : simple calcul de fréquence ou bien, pour les bigrammes, le rapport de vraisemblance RV [4] calculé de la manière suivante :

Pour tout couple (x_i,x_j) de mots qui se suivent, on calcule la table de contingence suivante :

- a_s : occurrences de (x_i,x_j) en semaine s
- b_s : occurrences de (x_i,x_l) en semaine s avec $l \neq j$
- c_s : occurrences de (x_k,x_j) en semaine s avec $k \neq i$
- d_s : occurrences de (x_k,x_l) en semaine s avec $k \neq i$ et $l \neq j$

$$RV_s(x_i,x_j) = a_s*\log(a_s) + b_s*\log(b_s) + c_s*\log(c_s) + d_s*\log(d_s) - (a_s+b_s)*\log(a_s+b_s) - (a_s+c_s)*\log(a_s+c_s) - (b_s+d_s)*\log(b_s+d_s) - (c_s+d_s)*\log(c_s+d_s) + (a_s+b_s+c_s+d_s)*\log(a_s+b_s+c_s+d_s)$$

Enfin, l'algorithme compare le score de chaque mot ou suite de mots, d'une semaine à l'autre pour faire ressortir les mots qui apparaissent, ceux qui disparaissent, et ceux en forte variation (à la hausse ou à la baisse).

Les résultats sont affichés sous la forme de TreeMap [1], une technique qui consiste à représenter en 2D des informations de dimension supérieure.

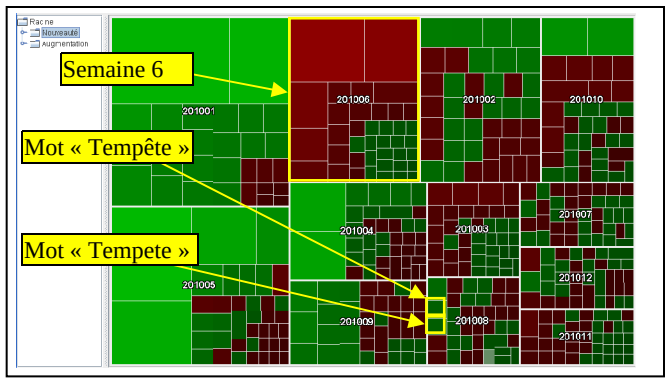

Fig. 6 : TreeMap des nouveaux mots

La figure montre les semaines et les mots qui présentent les variations les plus importantes (taille des rectangles) et leur sens de variation (verte pour une baisse et rouge pour une hausse).

---

[13] - http://en.wikipedia.org/wiki/Tag_cloud

[14] - http://www.flickr.com/

[15] - Afin de faire « ressortir » les bigramme et trigramme, un coefficient multiplicateur est attribué à leurs fréquences.

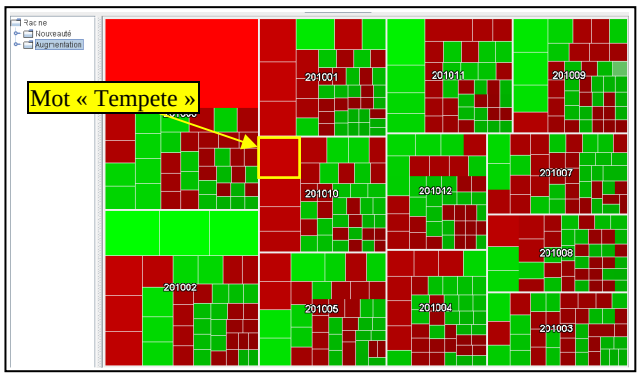

Fig. 7 : TreeMap des mots en augmentation

Les figures 6 et 7 font ressortir les fortes variations du mot « tempête », avec une brusque augmentation en semaine 8 et une forte chute en semaine 10.

La représentation sous forme de TreeMap présente l'avantage de mettre visuellement en évidence certains phénomènes, comme sur la figure 6, où l'on voit clairement qu'il y a eu beaucoup de disparitions de mots, car l'aire correspondant à cette semaine est globalement rouge. Cela constitue un indicateur pour aller ensuite regarder plus finement ce qui s'est passé cette semaine là.

4 Quelques résultats

Pour illustrer les possibilités du logiciel, voici quelques résultats obtenus :

- le lundi est le jour de la semaine où le nombre de questions est le plus élevé ;

- certaines questions très longues sont, en fait des mails, dans lesquels le client explique de manière détaillée sa demande ou sa problématique. On pourrait, dans ce cas, rediriger la question vers le traitement des mails ou bien lui faire une réponse adaptée en lui expliquant qu'il faut des questions courtes et que si il souhaite nous adresser un mail qu'il le fasse par le « bon » canal ;

- l'utilisation des évolutions de vocabulaire permet de détecter des questions qui apparaissent ponctuellement comme celles relatives à la « tempête ».

5 Conclusion

Afin de mieux comprendre les demandes qui lui sont faites et dans un but d'amélioration de Laura, l'agent conversationnel du site Bleu Ciel d'EDF, le logiciel NaviQuest a été développé pour naviguer dans la base des questions qui lui ont été posées. Ce logiciel mixe des techniques de visualisation et d'analyse textuelle Il s'agit d'une première version du logiciel. Plusieurs évolutions sont envisageables :

- intégrer d'autres techniques de représentation ou visualisation pour fouiller le corpus de questions (évolution des réponses ou thèmes détectés)

- intégrer des informations de type analyse de sentiments ou d'opinions, qui constitueraient des méta-données supplémentaires pour fouiller le corpus.

Une étape importante consistera à prendre en compte les retours de la Direction Marketing d'EDF pour la définition précise et les évolutions d'un tel logiciel.

Références

[1] B.B. Bederson, B. Shneiderman, M. Wattenberg. Ordered and Quantum Treemaps : Making Effective Use of 2D Space to Display Hierarchies. ACM Transactions on Graphics (TOG), 21, (4), Octobre 2002

[2] L. Bozzi, P. Suignard, C. Waast-Richard. Segmentation et classification non supervisée de conversations téléphoniques automatiquement retranscrites, Actes de la conférence TALN'09 2009.

[3] D. Coupland. Microserfs. Flamingo, 1996.

[4] B. Daille, 1994, Approche Mixte pour l'Extraction Automatique de Terminologie, Thèse de Doctorat.

[5] M. Garnier-Rizet, S. Guillermin-Lanne, F. Cailliau. CallSurf, Search by content, navigation and knowledge extraction on Call Center Conversational Speech, for marketing and strategic intelligence,

RIAO 2010.

[6] D. Gotz, Z. When, J. Lu, P. Kissa, N. Cao, W.H. Qian, S.X. Liu. HARVEST : An Intelligent Visual Analytic Tool for the Masses, IVITA, First International Workshop on Intelligent Visual Interfaces for Text Analysis, Hong-Kong, Chine, Février 2010.

[7] J. Heer. Prefuse : a software framework for interactive information visualization in Masters of Science, Computer Science Division, University of California, Berkeley (2004)

[8] A. Salem *et al.* (2002). Manuel d'utilisation de Lexico. Université Paris 3. http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/lexicoWWW/

[9] J. Stasko, C. Görg, Z. Liu, and K. Singhal. Jigsaw: Supporting investigative analysis through interactive visualization. In Proceedings of IEEE, VAST '07, Sacramento, CA, Octobre 2007

[10] F. B. Viégas, M. Wattenberg, F. Van Ham, J. Kriss, M. McKeon Many Eyes: A Site for Visualization at Internet Scale,. Infovis, 2007.

[11] Vuillemot, R., Clement, T., Plaisant, C., Kumar, A. (April 2009) What's Being Said Near "Martha" ? Exploring Name Entities in Literary Text Collections In Proceedings of IEEE VAST 2009.