Réécriture statistique de phrases basée sur des modèles de langage

Eric Charton[1], Juan-Manuel Torres-Moreno[1,2] et Eric SanJuan[1]

[1]Laboratoire Informatique d'Avignon, BP1228 – 84 911 Avignon cedex 9 – France

{eric.charton, juan-manuel.torres, eric.san-juan}@univ-avignon.fr

[2]Ecole Polytechnique de Montréal, Département de génie informatique
H3C3P8 Montréal (Québec) Canada

Abstract

In this article, we present our preliminary research in the field of the automatic sentence generation. With a corpus used as a basis to generate new sentences, we have explored various techniques of word and n grams combinations in conjunction with original approaches of filtering generated sentences to reduce the impact of combinatory explosion. Finally, we have adapted the statistical approach of machine translation to rewrite a sentence. We have obtained an innovative system for text and automatic rewriting in the same language. This introduces a new research perspective in the field of automatic document rewriting.

Résumé

Dans cet article, nous présentons nos travaux préliminaires de recherche dans le domaine de la génération automatique de phrases. Nous avons exploré des méthodes de construction de phrases syntaxiquement et sémantiquement correctes, en utilisant plusieurs approches. Nous avons transformé le système de décodage et le modèle de traduction par corpus bilingues alignés, en modèle de réécriture de phrase. Nous avons obtenu des résultats encourageants qui nous poussent à introduire un thème de recherche original : la réécriture automatique de textes. Dans une perspective plus générale, ces travaux s'inscrivent dans le cadre de la génération automatique de textes en langue naturelle.

Mots-clés : génération automatique de textes, apprentissage assisté par ordinateur, réécriture automatique de textes, réécriture statistique de textes, modèles de langage, génération de paraphrases.

1. Introduction

La génération automatique de phrases est l'une des pierres angulaires de l'activité de génération automatique de textes. Quel que soit le texte à produire (mise en forme de données lisibles en langage naturel, réponse automatique à des courriels, production de documentation technique) (Reiter E. 1997), le processus de génération de texte intègre toujours des suites consécutives de phrases au sein d'un document architecturé par un plan.

Mais il existe aussi un ensemble de domaines de recherches connexes, sans rapport direct avec la génération automatique de textes, qui déploient des modèles mathématiques et statistiques susceptibles de produire des phrases entièrement nouvelles, ou le cas échéant de compléter des phrases existantes, en partant le plus souvent d'un modèle de langage. C'est le cas par exemple des systèmes de traduction automatiques, ou de décodage pour la reconnaissance vocale.

Nous avons dans le cadre de notre recherche sur la génération automatique de textes, cherché à identifier dans quelle mesure ces modèles existants pouvaient être appliqués à la génération automatique de phrases syntaxiquement et sémantiquement correctes, en vue de leur réutilisation dans un système de génération de textes. Nous présentons ici ces travaux préliminaires de recherche.

1.1. Les modèles utilisés

Nous avons exploré des méthodes de construction de phrases syntaxiquement et sémantiquement correctes, en utilisant plusieurs approches.

- Nous avons exploré sur un plan purement expérimental les possibilités combinatoires de *n_grammes* issus d'un modèle de langage généré à partir d'un corpus (Deft07, 2007). Nous avons élaboré plusieurs techniques de filtrage et de guidage pour limiter l'explosion combinatoire et obtenir des phrases syntaxiquement correctes, en partant de simples combinaisons de *n_grammes*.
- Nous avons utilisé ensuite un modèle de langage associé à un décodeur, tel que ceux mis en oeuvre dans les systèmes de reconnaissance de la parole (Aubert L., 2002 ; Nocera P., Linares G., 2004). Ce système utilise l'algorithme de recherche A* en tant que sonde pour compléter des phrases dont tout ou partie est non ou mal reconnu.
- Pour finir, nous avons, cherché à transformer le modèle de traduction par corpus bilingues alignés (Brown F., Cocke J., et al, 1990), en modèle de réécriture dans une même langue.

Nous avons obtenu des résultats préliminaires encourageants qui nous poussent à introduire un thème de recherche original : la réécriture automatique de phrases. Dans une perspective plus générale, ces travaux s'inscrivent dans le cadre de la génération automatique de textes en langue naturelle.

Cet article est organisé comme suit : dans la *section 2* nous dressons un état de l'art des systèmes de génération automatique de phrases. Dans la *section 3*, nous détaillons les trois modèles statistiques et probabilistes (combinatoire, décodage de la parole, traduction automatique) que nous avons étudiés dans une optique de génération automatique de phrases.

En *section 4*, nous présentons nos résultats préliminaires obtenus avec le corpus « débat » de *Deft'07* (Deft07, 2007). Pour finir, dans la *section 5*, nous présentons les perspectives de recherches nouvelles que nous envisageons avec ces systèmes, notamment dans le domaine de la réécriture automatique.

2. État de l'art de la génération automatique de phrases

La construction des anagrammes est un divertissement dont la première source connue remonterait à l'antiquité. Une anagramme est le résultat de la permutation des lettres d'un ou plusieurs mots de manière à produire d'autres mots et donc de nouvelles phrases, qui ont un sens[1]. On évoquera aussi le principe de la paraphrase, théorisé par (Mel'CuK, 1988) qui consiste à remplacer les mots d'une phrase par des synonymes, et à modifier sa structure. Ce principe à fait l'objet de quelques recherches connexes (Duclaye F., Collin O., Yvon F., 2003)

[1] L'art de l'anagramme aurait été inventé par le poète grec Lycophron[wikipedia].

appliquées à la recherche automatique de paraphrases en utilisant des ressources externes, telles que des sites internet.

2.1. Génération de phrases par modèle combinatoire

Le premier à explorer les formes possibles de génération automatique de textes en utilisant une méthode probabiliste est Shannon dans (Shannon C.E., 1948). L'auteur explore en préambule de sa *Théorie mathématique de la communication* les diverses formes de combinaisons alphabétiques, puis lexicales qui permettent de produire des phrases de plus en plus proches d'un anglais « intelligible »[2], en partant de procédés aléatoires. Il utilise pour cela des modèles markoviens de second ordre, élaborés d'après les fréquences des bigrammes des mots anglais calculés dans (Dewey, 1923). L'une des phrases qu'il fournit à titre d'illustration est la suivante :

THE HEAD AND IN FRONTAL ATACK ON AN ENGLISH WRITER TAHTE THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

Faute de puissance machine suffisante en 1948 pour modéliser un véritable langage, Shannon ne livre ses démonstrations qu'à titre d'illustration. Ses phrases n'ont que la « sonorité » de l'anglais. Il faudra attendre les années 80 et divers travaux dont ceux de (Katz S. M. 1987 ; Bahl L.R. et. al. 1983) pour imaginer des systèmes de génération ou reconstruction de phrases en partant de modèles de langage.

2.2. Traduction automatique de phrases

L'une des premières applications concrètes des travaux exploratoires de Shannon fût l'approche statistique proposée pour la traduction assistée par ordinateur de (Brown F., Cocke J., et al, 1990). Dans le modèle proposé, on utilise le théorème de Bayes pour minimiser le risque d'erreur d'une traduction de phrases, en exploitant deux corpus alignés. L'idée est que pour une phrase T dans un langage cible, il soit possible de choisir une phrase S la plus probable, en postulant qu'elle maximise $p(S|T)$. Cette approche a depuis donné naissance à de multiples systèmes de traduction assistés par ordinateurs.

On notera que si l'on fait abstraction du fait que T et S sont des phrases écrites dans deux langues différentes, transformer une phrase en une autre, en conservant son sens, est une activité de réécriture.

2.3. Décodage et réécriture de phrases

Une autre application qui peut être considérée comme une forme de réécriture de phrase est le décodage de parole. Son modèle applicatif cherche à transcrire un signal audio numérisé en un texte écrit. Les difficultés particulières que pose cette transcription – bruitage, difficulté de segmentation, contextualisation[3] – ont conduit à mettre au point un système de reconstruction du signal, inspiré des méthodes de décodage proposées par Shannon. Ici, on considère que le signal bruité transcrit sous sa forme écrite est incomplet, et qu'il est possible par un décodage reposant sur l'exploitation probabiliste d'un modèle de retranscription, de le reconstituer.

[2] Shannon parle de « resemblance to ordinary english text ».

[3] Par exemple pour identifier des mots phonétiquement identiques mais dont l'écriture diffère.

Un modèle de langage *n_grammes* est conçu d'après un corpus et permet d'élaborer un graphe de probabilité de transitions entre groupes de termes. Ce modèle tel qu'il est présenté par (Delaney B., Anderson T. 2006) pour perfectionner la proposition de (Brown F., Cocke J., et al, 1990), connaît de nombreuses variantes.

Des systèmes tels que ceux préconisés par (Nocera P., Linares G., 2004) utilisent des algorithmes d'exploration A* améliorés pour la transcription de signal de parole en temps réel.

Ils nous intéressent en tant que dispositifs capables de reconstruction de phrases incomplètes permettant d'identifier des mots manquants dans une transcription. Nous postulerons qu'ils sont capables de ré-écrire une phrase artificiellement bruitée (c'est à dire dont on aurait supprimé une partie des mots en vue de la soumettre à un processus de reconstruction).

3. Propositions de modèles de génération et de réécriture automatique de phrases

La principale difficulté rencontrée lors de la génération statistique de phrases, est proche de celle rencontrée dans le problème du décodage de parole ou de la traduction automatique. Cette difficulté réside dans les explosions combinatoires qui découlent des méthodes de ré-assemblage des phrases.

En tant que tel, assembler des combinaisons de mots, des phrases ou de *n_grammes* pour former des phrases intelligibles est relativement aisé (cf. Shannon plus haut). Identifier ou élaborer les algorithmes et les heuristiques qui vont permettre de sélectionner dans les « réservoirs de phrases » produits par assemblage, celles qui répondent précisément à un besoin, tout en respectant les contraintes sémantiques et syntaxiques d'une langue, est beaucoup plus difficile.

3.1. Formulation générale du problème de décodage

On utilise pour résoudre cette difficulté des heuristiques de décodage dont le fondement est le plus souvent l'équation générique du cadre probabiliste Bayesien, appliqué à la composition ou à la reconstruction de phrases d'après un modèle de langage :

$$W' = Argmax_W \, P(O|W)P(W)$$

Ou $O=o_1, ..., o_T$ est la séquence d'observation représentant le signal d'entrée plus ou moins bruité (le signal audio à contenu vocal, les phrases à traduire, des phrases incomplètes, par exemple), vu durant un temps T, et $W=w_1, ..., w_n$ une séquence de mots W pris dans un vocabulaire de taille N, extrapolé d'après un corpus d'apprentissage.

En pratique on gagne à remplacer les mots W par des séquences d'états HMM indépendants du contexte, constitués de probabilité d'apparitions de *n-grammes* (voir (Slava et Katz 1987) et (Aubert 2002) pour un tutoriel plus détaillé)).

Dans ce cas, la séquence de mots reconnus W pour un signal d'entrée O est déterminée par la séquence dont les états sont les plus probables, après exploration des réseaux bayésiens.

3.2. Algorithmes et méthodes proposées

Dans les paragraphes qui suivent, nous allons décrire les trois algorithmes que nous avons expérimentés pour générer ou réécrire des phrases en appliquant ces méthodes[4].

3.2.1. Génération de phrases par combinaisons de n_grammes guidée

Notre première expérience a consisté en partant d'un modèle de langage *n_grammes* construit d'après le corpus des « Débats parlementaires » de la campagne DEFT'07, à générer aléatoirement un « réservoir de phrases » syntaxiquement et sémantiquement correct. Nous expliquerons dans les conclusions en quoi un tel dispositif peut représenter un attrait dans le cadre d'une réécriture statistique. Pour illustrer notre propos, partons de la phrase suivante issue de ce corpus :

j'ai connu cela en tant que maire

Un professionnel de la réécriture (secrétaire de rédaction, résumeur, relecteur d'une société d'édition, etc), en recourant à des permutations, pourra recomposer cette phrases sous l'unique forme suivante :

en tant que maire j'ai connu cela

Si nous cherchons maintenant à recomposer cette phrase avec une méthode statistique, il nous faut dans un premier temps explorer toutes les permutations (soit pour une phrase de 8 mots, *8 !*) et dans un second temps sélectionner dans le sac de phrases produit par ce procédé, un ensemble de « phrases candidates ». Dans les 40 320 phrases candidates de notre exemple précédent, nous trouverons des phrases légèrement défectueuses, mais utilisables dans une application dégradée :

cela en tant que maire j'ai connu

D'autres seront totalement inutilisables, telle que ci-dessous :

j'ai en cela que maire tant connu

Ces exemples triviaux nous permettent de souligner la double difficulté que pose la réécriture ou la génération statistique de texte :

- L'explosion combinatoire mobilise très rapidement de manière inacceptable les capacités de traitement des ordinateurs moderne (gérer les permutations *20 !* d'une phrase de 20 mots pourtant courante dans une multitude de document est techniquement impossible si l'on adopte la force brute).
- Toute recombinaison de phrases d'un corpus en vue de générer de nouvelles phrases produit immanquablement des déchets inutilement coûteux.

Il faut donc pour générer ou ré-écrire des phrases, utiliser des modèles « guidés ». On peut réduire l'explosion combinatoire en déterminant la probabilité qu'un segment débute ou se termine par un terme ou un groupe de terme. Nous intitulons ces informations les « *terminaisons* » et les *« amorces »,* et les évaluons en générant d'après le corpus un sous ensemble du modèle de langage décrivant les probabilités :

[4] Les programmes en perl ayant servi à réaliser ces expériences préliminaires sont disponibles sur le site d'un des auteurs http://www.echarton.com/logiciels.html.

$$P(Sc) = P(a_n).P(t_n)$$

Ou *P(Sc)* est la probabilité que la phrase *Sc* issue d'une combinaison des mots de la phrase d'origine *S* soit syntaxiquement correcte, sachant qu'elle débute par l'amorce *a* de *n* mots, et qu'elle s'achève par la terminaison *t* de *n* mots.

En utilisant ces deux informations - « amorces » A et « terminaisons » T – nous réduisons considérablement la complexité du processus de génération de phrases. Les phrases obtenant une probabilité nulle sont écartées (c'est le cas si, par exemple, leur amorce ne correspond à aucun des *n_grammes* a_n de *A*). Celles restantes sont triées par leurs scores de probabilité. On considère que les *s* combinaisons obtenant le meilleur score sont des phrases potentiellement correctement ré-écrites.

3.2.2. Adaptation du modèle de décodage de parole à la réécriture de phrases

Une autre approche peut consister à reconstruire une phrase d'après les « mots » intermédiaires qu'elle contient, en utilisant un modèle de langage *n_grammes*. Cette technique exploite l'algorithme *A** pour trouver le meilleur chemin dans un graphe issu du modèle de langage.

L'algorithme *A** utilise une fonction d'estimation *F(x,y)*, qui correspond à la somme des coûts :

- des transitions du chemin optimal entre le noeud de départ et le noeud courant (g(x))
- de la transition entre le noeud courant et le noeud suivant (c(x,y)
- de la fonction h(y) qui estime le coût du chemin restant entre le noeud suivant et le dernier noeud (h(y))

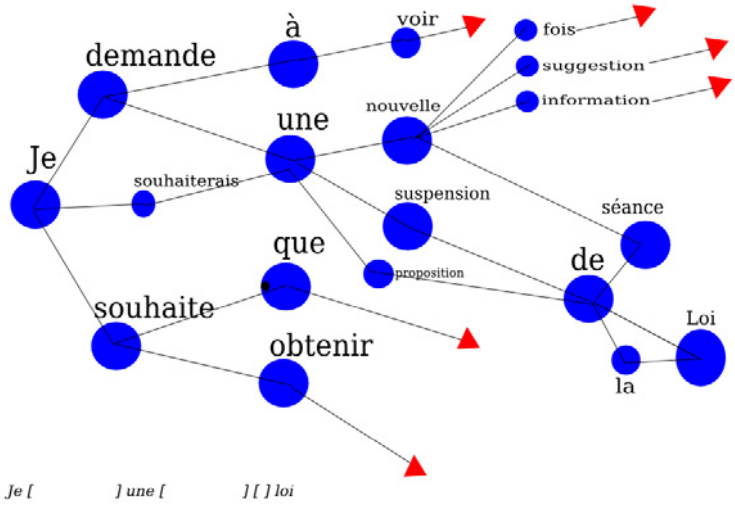

Figure 3.1 : exemple de graphe exploré pour reconstituer une phrase bruitée

L'algorithme *A** utilise une liste ordonnée intitulée *Open*, qui contient tous les noeuds à explorer dans l'ordre décroissant de leur valeur *F*. A chaque itération de l'algorithme, le premier noeud *x* de *Open* est retiré de la liste, et pour chaque noeud *y* successeur du noeud *x* dans le graphe, la fonction d'estimation *F(x,y)=g(x)+c(x,y)+h(y)* est calculée, et la nouvelle hypothèse *y* ajoutée dans la liste *Open.*

Pour faire fonctionner ce modèle, nous générons un modèle de langage *3-grammes*[5] d'après le corpus d'apprentissage. Dans un second temps, nous soumettons les phrases du corpus d'origine au modèle de langage en vue de leur réécriture.

Ces phrases sont filtrées en fonction du poids des mots qu'elles contiennent. En pratique, nous utilisons les logarithmes des probabilités d'apparition d'un mot calculés dans le modèle de langage en conjonction avec un seuil, pour décider si le mot original est conservé, ou considéré comme un signal bruité. Dans l'exemple de la figure 3.1, la phrase originale avec ses *LOG* de probabilités d'apparition des mots est :

je(-2.34) souhaiterais(-4.28) une(-2.11) proposition(-3.43) de(-1.23) loi (-2.57)

Si nous fixons le seuil de log-probabilité auquel nous définissons un mot comme bruité à -3, nous obtenons :

je(-2.34) ***[bruit]*** *une(-2.11)* ***[bruit]*** *de (-1.23) loi (-2.57)*

Nous sommes alors en mesure d'appliquer l'algorithme d'exploration de graphe pour compléter les mots manquants par des propositions issues du modèle de langage. Ces propositions seront classées par ordre de probabilité que la phrase d'origine *O*, corresponde à la phrase reconstituée *R* sachant *O*, soit *P(R=O|R)*. Ce qui nous donne :

- Je [souhaiterais] une [proposition] de loi (1)
- Je [voudrais] une [proposition] de loi (0.71)
- je [demande] une [suspension] de loi (0.51)

3.2.3. Adaptation du modèle de traduction à la réécriture

Une autre approche sera d'adapter le *modèle de traduction par approche statistique* proposé par (Brown et Al, 1990). L'avantage de ce modèle est qu'il est très éprouvé et sa fiabilité reconnue. Son inconvénient est qu'il fonctionne comme un système de décodage et impose l'usage de corpus alignés pour fonctionner.

Dans le *modèle de traduction*, on met en correspondance pour chaque phrase *S*, du corpus d'une langue à traduire, une phrase *T* correspondant à sa traduction comme ceci :

S=Nick does beat the dog

T=*Le chien est battu par Nicolas*

Pour calculer les probabilités d'alignement, dans ce modèle, les système de traduction « état de l'art »[6] mettent en oeuvre plusieurs techniques (Och et Ney, 2000) qui tiennent compte de la *fertilité (*le nombre de mots qu'une source « produit ») et des correspondances contenues dans les corpus alignées. Les meilleures techniques reposent sur l'usage d'apprentissage avec Viterbi ou HMM. De manière simplifiée, on peut schématiser ce processus par la correspondance suivante :

(Le chien est batu par Nicolas | Nick (6) does beat (3,4) the(1) dog(2))

Ou on multipliera les probabilités que « Nick » ait une fertilité *f* de 1 par *P(Nicolas | Nick)* et ainsi de suite, ce qui nous donne :

[5] Spécifications ARPA (http://www.speech.sri.com/projects/srilm/manpages/ngram-format.html).

[6] Voir notamment l'outil Giza++ : www.fjoch.com/GIZA++.html.

P(f=1| Nick).P(Nicolas| Nick) . P(f=0|does) . P(f=2|beat).P(est|beat).P(battu|beat). P(f=1| the).P(le | the) . P(f=1| dog).P(chien| dog)

Notre application ne concerne qu'une transcription dans le même langage, et est donc moins sensible aux effets induits par les réorganisations de phrases lors du passage d'une langue à une autre, pris en charge par les ré-alignements. Nous pouvons donc utiliser pour notre application un algorithme simplifié, tel que celui utilisé dans les premiers modèles de traduction. Notre démarche pour adapter ce modèle à un processus de réécriture pourraît être ici d'aligner deux segments de sens identique comme ceci :

Le chien est battu par Nicolas

Nicolas a molesté l'animal

Et d'introduire les informations de correspondance sous cette forme :

(Le chien est battu par Nicolas | Nicolas (6) a molesté (3,4) l'animal (1,2))

Dans ce modèle, l'alignement décrit non plus la traduction, mais la réorganisation de la phrase cible, ce qui revient d'ailleurs parfois à l'introduction de synonymes, de meronymes ou d'hyperonymes[7]. Pour généraliser le modèle de traduction, il est possible d'introduire des « *jokers* » qui décrivent des séquences inconnues dans l'alignement, comme suit :

*(Le chien est battu par * | * (6) a molesté (3,4) l'animal (1,2))*

Par suite, on cherchera avec le modèle ainsi conçu, à trouver la phrase S qui maximise *P(S)P(T|S)*. Le principe retenu est celui de la recherche par pile (Bahl et al. 1983). La recherche est réalisée par itération et sélectionne une proposition de la liste obtenant le meilleur score de probabilité. La recherche est interrompue quand le meilleur alignement est obtenu.

4. Résultats préliminaires

Nous avons réalisé un ensemble d'expériences préliminaires avec ces trois modèles. Nous avons consulté les résultats produits visuellement et vérifié la qualité syntaxique et sémantique d'une centaine de phrases proposées par chaque modèle, pour une dizaine de phrases soumises à la réécriture.

Le modèle combinatoire associé à un guidage propose des résultats difficiles à exploiter : les meilleures propositions d'assemblage, bien que syntaxiquement et sémantiquement acceptables dans 30% des cas, ne reproduisent que rarement le sens original de la phrase. Nous en concluons que seule l'idée d'un modèle statistique des amorces et des terminaisons de phrases construit d'après un corpus est exploitable et utilisable dans un système hybride utilisant des modèles de langage et un système de décodage.

Le modèle de décodage offre 30 à 70% de réécritures acceptables. Les variations de résultats sont largement dues au mode d'introduction de bruit dans les phrases à réécrire. Les meilleurs résultats ont été obtenus en ne conservant que les 40% de mots de plus fort poids TF.IDF dans les phrases. Le modèle par traduction a été brièvement exploré à ce stade de nos recherches.

[7] On notera ici que pour ce qui concerne le cas particulier de l'hyperonyme, le modèle mis en oeuvre ne peut fonctionner que si l'hyperonyme est utilisé dans la phrase cible du corpus T dit de « traduction ». L'inverse, c'est à dire l'introduction d'un hyponyme dans la phrase source S utilisée dans le corpus de référence conduirait à créer un fort risque de réécritures de contre sens (i.e. : *chien* peut être ré-écrit par *animal*, mais *animal* – pris en tant que classe généralisatrice - ne peut pas être ré-écrit par *chien)*.

5. Conclusions et perspectives

Dans cet article, nous avons décrit des modèles de décodage issus des technologies de traitement et de l'analyse de la langue naturelle que nous avons adaptés à la réécriture automatique de phrases par voie statistique. La combinaison des trois modèles proposés permet d'envisager la construction d'un système hybride, capable de récrire des phrases syntaxiquement correctes par une méthode statistique.

C'est une proposition innovante qui ouvre de nouvelles perspectives de recherches. De très nombreuses applications peuvent tirer partie d'un système de réécriture automatique de textes. Production de contenus, écriture de documents, rédaction assistée, correction automatique en sont quelques exemples.

Nos expériences préliminaires nous ont convaincus de la validité de nos modèles théoriques. Nous nous attachons maintenant à concevoir un système prototype permettant d'explorer les différentes adaptations possibles de ces modèles, à la réécriture de textes et aux paraphrasages.

A terme, les phrases produites par ce système pourraient être les éléments de base d'un système de génération automatique de texte performant.

Références

Aubert L. (2002). An overview of decoding techniques for large vocabulary continuous speech recognition. C*omputer Speech and Language,* Vol(16): 89-114.

Bahl L. R. and Jelinek F. and Mercer R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE transactions on pattern analysis and Machine Intelligence PAMI-*5(2:179-190).

Brown F. and Cocke J. et al (1990). A Statistical approach to machine translation. *Computational Linguistics,* Vol(16: 2).

Deft'07 (2007). 3e DÉfi Fouille de Textes, Grenoble (38).

Delaney D. and Anderson T. (2006). *An efficient Graph Search Decoder for Phrase-Based Statistical Machine Translation*.

Dewey G. (1923). *Relative Frequency of English Speech Sounds.* Harvard University Press.

Duclaye F., Collin O., Yvon F. (2003). Apprentissage automatique de paraphrase pour l'amélioration d'un système de Questions-Réponses. *TALN'2003,* Batz/mer.

Katz M. S. (1987). *Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer*.

Mel'Cuk I. (1988). Paraphrase et lexique dans la théorie linguistique Sens-Texte. *Lexique et paraphrase,* Lexique (Lexique), ISSN 0756-7138.

Nocera P. and Linares G. and Massonié D. (2004). Phoneme Lattice Based A* Search Algorithm for Speech Recognition. *Lecture Notes in Computer Science,* Vol.2448/2002.

Och F. J. and Herman N. (2000). Improved Statistical Alignement Model. In *Proceedings of ACL'00,* Hong Kong.

Reiter E. (1997). Building Applied Natural Language Generation Systems. *Natural Language Engineering.*

Shannon C. E. (1948). A Mathematical Theory Of Communication. *Reprint from The Bell System Technical Journal,* Vol.(27): 379-423.