Etude comparative de modélisation de langage par bigrams et par multigrams pour la reconnaissance de parole

Yassine Mami[(1)], *Frédéric Bimbot*[(2)]

(1) France Télécom - 2 Avenue Pierre Marzin 22307 - Lannion - FRANCE
yassine.mami@francetelecom.com
(2) IRISA (CNRS & INRIA) - Campus de Beaulieu - 35042 Rennes Cedex - FRANCE
bimbot@irisa.fr

ABSTRACT

The use of stochastic ngram models has a long and successful history in the research community ; nowadays ngrams are becoming quite common in oprational applications, as real-life situations demand more robust and flexible solutions. This approach is particularly interesting for its effectiveness and its robustness, but limited to modeling only local linguistic structures. To overcome this limitation, we investigate on the use of variable-length models. In this paper we consider the multigram language models and we integrate them in a speech recognition system. The experiments are carried out on a France Telecom's dialogue application for stock exchange.

1. INTRODUCTION

Le modèle de langage est parmi les plus importantes composantes dans un système de reconnaissance de parole continue. Ces modèles sont généralement des modèles de langage probabilistes et sont utilisés pour guider le décodage acoustique en apportant des contraintes linguistiques. Actuellement, les modèles de langage couramment utilisés sont les ngrams qui constituent l'état de l'art en la matière. Dans ces modèles, on estime la probabilité d'une phrase à partir des probabilités conditionnelles d'apparition d'un mot ou d'une classe de mots, étant donnés les $n-1$ mots ou classes de mots précédents. Cette approche est particulièrement intéressante pour son efficacité et sa robustesse mais limitée à la modélisation des structures linguistiques à horizon fixe. Afin de pallier les difficultés des modèles ngrams, nous proposons d'utiliser des modèles à séquences de longueurs variables [3] [5]. Les modèles multigrams font l'hypothèse qu'une phrase peut être décomposée en séquences de mots de longueur variable en faisant intervenir des probabilités sur les différentes segmentations possibles [4] [7]. Ces segmentations donnent au modèle multigram le pouvoir de capter des contraintes fortes que ne peut pas saisir un modèle ngram en raison de sa taille d'historique fixe. D'autres variantes des modèles à séquences de longueurs variables ont été proposées comme les multiclasses [10]. Cette approche est basée sur la hiérarchisation des séquences de classes syntaxiques de longueur variable. Ce type de modèle apporte aux modèles de type multigram la prise en compte de dépendances entre les séquences de classes syntaxiques de longueur variable.

Ainsi, cet article s'articule en trois parties. Dans la première, nous rappelons le principe de la modélisation par ngrams. La deuxième partie est consacrée à la modélisation par multigrams. Dans la dernière, nous présentons des évaluations comparatives de ces deux familles d'approches sur les données d'une application de dialogue dans le domaine boursier. Enfin, un ensemble de conclusions et de perspectives terminent cet article.

2. MODÉLISATION PAR NGRAMS

Le but d'un modèle de langage stochastique est d'attribuer à une séquence de mots W une probabilité $p(W)$. Soit $W = \{w_1, \cdots, w_T\}$ une séquence de mots. La probabilité de la séquence de mots s'exprime [8] :

$$p(W) = p(w_1) \prod_{i=2}^{T} p(w_i|w_1 \cdot\cdot w_{i-1}) = p(w_1) \prod_{i=2}^{T} p(w_i|h_i) \quad (1)$$

où h_i est l'historique du mot w_i.

L'approximation ngrams consiste à limiter l'historique d'un mot w_i à ses $n-1$ mots prédécesseurs, soit $h_i = w_{i-n+1} \cdots w_{i-1}$. Le modèle ngram constitue l'état de l'art actuel de la modélisation stochastique du langage. C'est un modèle stochastique pur, qui n'utilise aucune connaissance d'ordre syntaxique ou sémantique. Malgré sa simplicité de modélisation, ce type de modèle est très répandu dans les systèmes de reconnaissance vocale ; lors du décodage de Viterbi, en combinaison avec le modèle acoustique, il est assez discriminant pour favoriser certaines hypothèses de mots par rapport à d'autres et de ce fait, réduire de façon appréciable l'espace de recherche. Dans la pratique, n est rarement choisi supérieur à 3. Pour $n = 2$, la probabilité d'un mot ne dépend que du mot qui le précède ; il s'agit d'un modèle bigram :

$$P(W) \approx p(w_1) \prod_{i=2}^{T} p(w_i|w_{i-1}) \quad (2)$$

Les paramètres du modèle ngrams sont souvent estimés par maximum de vraisemblance. Pour un modèle bigram, la probabilité d'apparition du mot w_i précédé du mot w_j est :

$$p(w_i|w_j) = \frac{c(w_j w_i)}{c(w_j)} \quad (3)$$

où $c(X)$ est le nombre d'occurrences d'un événement X dans le corpus d'apprentissage.

Dans cet article, les probabilités bigrams non vues sont lissées par *discounting*. Cette technique consiste à retrancher une masse de probabilité de la masse totale accordée aux événements observés lors de l'apprentissage, masse qui se trouve ensuite redistribuée parmi les événements non observés. Plusieurs variantes de la technique de *discounting* existent : dans cet article, nous avons utilisé l'*absolute discounting* [9].

3. MODÉLISATION PAR MULTIGRAMS

Dans l'approche multigrams, une phrase est considérée comme une concaténation d'un ensemble de séquences de mots de longueurs variables [2] [5]. Soit une phrase W de T mots $W = \{w_1, \cdots, w_T\}$ et soit $S = \{s_1, \cdots, s_q\}$ une segmentation possible de la phrase W. Chaque segment est de longueur maximale m. La vraisemblance jointe de la phrase W et de la segmentation S est égale au produit des probabilités de chaque segment s_i, soit :

$$\mathcal{L}(W, S) = \prod_{i=1}^{i=q} p(s_i) \tag{4}$$

L'estimation de l'ensemble des paramètres d'un modèle multigram Θ est obtenue par maximum de vraisemblance. La vraisemblance de la phrase W est la somme des vraisemblances jointes de toutes les segmentations possibles S :

$$\mathcal{L}(W) = \sum_S \mathcal{L}(W, S) \tag{5}$$

La vraisemblance $\mathcal{L}(W)$ est une fonction non linéaire des paramètres du modèle Θ ce qui rend la maximisation directe par rapport à Θ très difficile. L'algorithme EM (*Expectation Maximisation*) permet de résoudre d'une manière élégante ce problème d'estimation relativement complexe. Au sens de cet algorithme, les données observées sont la suite des mots W et les données manquantes (ou cachées) sont les segmentations de la phrase S. L'estimation des paramètres est un processus itératif qui se déroule comme suit :

1. Initialiser les probabilités des séquences à partir des fréquences d'occurrences (cf. paragraphe 3.1).
2. Ré-estimer les probabilités des séquences multigrams (cf. paragraphe 3.2).
3. Itérer 2 jusqu'à un critère d'arrêt (la vraisemblance ne croît plus ou le nombre maximum d'itérations est atteint).

3.1. Initialisation

Soit $\mathcal{D} = \{s_1, \cdots, s_m\}$ l'ensemble des séquences multigrams formé à partir de la combinaison des mots du lexique. Les probabilités $p(s_i)$ de ces séquences peuvent être initialisées à partir de leurs fréquences d'apparition, soit :

$$p^0(s_i) = \frac{c^0(s_i)}{c^0} \tag{6}$$

où $c^0(s_i)$ est le nombre d'occurrences du segment s_i dans le corpus d'apprentissage, c^0 est le nombre total d'occurrences de touts les segments.

3.2. Estimation des probabilités des séquences multigrams

Les probabilités des séquences multigrams peuvent être apprises en utilisant l'algorithme EM ou l'algorithme Viterbi. Dans cet article, nous utilisons l'algorithme EM. La formule de ré-estimation des paramètres est alors :

$$p^{(k+1)}(s_i) = \frac{\sum_S c(s_i|S)\mathcal{L}^{(k)}(W, S)}{\sum_S c(S)\mathcal{L}^{(k)}(W, S)} \tag{7}$$

où $\mathcal{L}^{(k)}(W, S)$ est la vraisemblance de la phrase W à l'itération (k). La quantité $c(s_i|S)$ est le nombre d'occurrences de s_i dans une segmentation S, et $c(S)$ est le nombre total des séquences.

L'équation 7 est implémentée en utilisant la procédure *forward-backward*. Pour une séquence s_i de l mots, la formule de ré-estimation est :

$$p^{(k+1)}(s_i) = \frac{\sum_{t=1}^{T} \alpha_l^{(k)}(t)\beta^{(k)}(t)\delta^{s_i}_{[w(t-l+1)\cdots w(t)]}}{\beta^{(k)}(0)\gamma^{(k)}(T)} \tag{8}$$

Avec

$$\delta^{s_i}_{[w(t-l+1)\cdots w(t)]} = \begin{cases} 1 \text{ si } [w(t-l+1)\cdots w(t)] = s_i \\ 0 \text{ sinon} \end{cases}$$

La variable *forward* $\alpha(t)$ est la vraisemblance de la première partie de la phrase $W_1^t = \{w_1, \cdots, w_t\}$:

$$\alpha(t) = \sum_{l=1}^{n} \alpha(t-l)p([w(t-l+1)\cdots w(t)]) = \sum_{l=1}^{n} \alpha_l(t) \tag{9}$$

La variable *forward* $\gamma(t)$ est le nombre moyen des séquences dans une segmentation possible de $W_1^t = \{w_1, \cdots, w_t\}$:

$$\gamma(t) = 1 + \sum_{l=1}^{n} \gamma(t-l)\frac{\alpha_l(t)}{\alpha(t)} \tag{10}$$

la variable *backward* $\beta(t)$ est la vraisemblance de l'autre partie de la phrase $W_t^T = \{w_t, \cdots, w_T\}$:

$$\beta(t) = \sum_{l=1}^{n} p([w(t+1)\cdots w(t+l)])\dot{\beta}(t+l) \tag{11}$$

avec $\alpha(0) = \beta(T) = 1$, $\gamma(0) = 0$ et $1 \leq t < T$.

4. EVALUATION

4.1. Evaluation d'un modèle de langage

Si le modèle de langage est considéré comme une entité indépendante de l'application vocale, les performances sont souvent exprimées en terme de perplexité. C'est un indicateur de la capacité de prédiction du modèle de langage. La perplexité est donnée par la formule suivante :

$$PP(W) = 2^{-\frac{1}{T}\log_2 \mathcal{L}(W)} \tag{12}$$

La perplexité s'avère être un bon estimateur de la qualité d'un modèle de langage en tant qu'entité autonome. Cependant, sa valeur n'est pas forcément corrélée aux performances de ce modèle intégré dans une application de reconnaissance vocale et de son comportement en coopération avec le modèle acoustique.

4.2. Evaluation d'un système de reconnaissance de parole

Les performances d'un système de reconnaissance de parole continue sont exprimées en taux d'erreur de mots WER :

$$WER = \frac{Ins + Sub + Omi}{OK + Sub + Omi} \tag{13}$$

où Ins, Sub et Omi représentent respectivement le nombre d'insertions, le nombre de substitutions et le nombre d'omissions. OK est le nombre des mots correctement reconnus.

Nous pouvons également évaluer le système de reconnaissance en terme de précision $\mathcal{P}$ et de rappel $\mathcal{R}$ qui sont définit par [1] :

$$\mathcal{P} = \frac{OK}{OK + Sub + Ins} \text{ et } \mathcal{R} = \frac{OK}{OK + Sub + Omi}$$

Les valeurs de précision et de rappel sont combinées en une seule valeur d'évaluation en utilisant la $F\text{-}mesure$:

$$F = \frac{2 \cdot \mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}} \quad (14)$$

Le point de fonctionnement est obtenue en maximisant la $F\text{-}mesure$ ce qui permet de maximiser conjointement la précision et le rappel.

4.3. Contexte expérimental

Les expériences ont été réalisées sur une base de données interne France Télécom R&D d'une application dans le domaine boursier. Cette application de dialogue permet de gérer un portefeuille d'actions et offre les fonctionnalités suivantes :

– consultation de portefeuilles d'actions,
– consultation de carnets d'ordres,
– transactions (achats ou ventes d'actions), item et demande d'informations liées au domaine boursier.

Le domaine bancaire est un domaine très complexe, avec un vocabulaire technique spécifique et pointu. Beaucoup de recouvrements de mots existent entre les différents concepts de l'application, ainsi que des ambiguïtés pour les concepts critiques que sont les dates, les montants et les quantités. Pour cette application, le lexique est constitué de 2730 mots, dont plus de 1300 correspondent à des noms de titres d'actions sur lesquels le client peut effectuer des transactions. Le modèle de langage utilisé intègre trois classes : "ACTIONS", "MOIS" et "SOMMES". Elles représentent respectivement les noms d'actions, les mois de l'année et les sommes. Dans ce modèle, nous avons considéré que la probabilité d'appartenance d'un mois de l'année à la classe "MOIS" ou d'une somme quelconque à la classe "SOMMES" est uniforme. Par contre, trois niveaux de probabilités sont introduits au sein de la classe "ACTIONS" : le niveau de probabilité le plus fort concerne les 40 noms d'actions du CAC-40. Les deux autres permettent de classer le reste des noms d'actions.
Les données d'apprentissage se présentent sous la forme d'un corpus constitué de transcriptions de phrases prononcées par des utilisateurs de l'application de dialogue. Le corpus d'apprentissage est constitué de 24554 énoncés. Les phrases du corpus d'apprentissage sont pour la plupart des questions, des requêtes d'utilisateurs ou des commandes génériques d'annulation de requête ou de réinitialisation du dialogue, communes à toute application de dialogue. La longueur moyenne des phrases est de 6.5 mots. Le corpus de test est constitué de 1500 énoncés. La longueur moyenne des phrases est de 4.1 mots. Pour l'apprentissage des multigrams, nous avons utilisé le toolkit [6].

4.4. Evaluation de la perplexité des multigrams

Avant de tester l'intégration du modèle de langage multigram dans un système de reconnaissance de parole, nous évaluerons la perplexité du modèle multigram en fonction de m, le nombre maximal des mots dans une séquence multigram.
La figure 1 trace les variations de la perplexité pour différentes valeurs d'occurrence minimal et un nombre d'itération égal à 5. Ces deux paramètres s'avèrent peu critiques pour la perplexité. Les paramètres des multigrams ont été estimés par l'algorithme EM (équation 7). Les probabilités des séquences non vues sont lissées en apprenant ces probabilités sur le corpus constitué du corpus d'apprentissage et du lexique.

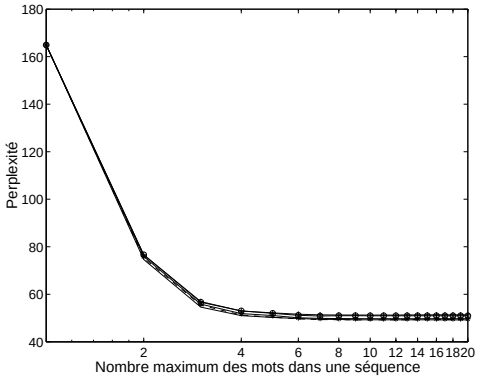

FIG. 1: Variations de la perplexité en fonction de m.

La figure 1 montre que la perplexité est maximale pour un modèle unigram (toutes les séquences ont une longueur égale à 1). La perplexité décroît très rapidement pour des valeurs du nombre maximal de mots dans une séquence inférieur à 4. Au delà de cette valeur, la perplexité varie peu et atteint une valeur minimale de 49.2 pour un nombre maximal de 5 mots dans une séquence multigrams. Au delà de 4 mots dans une séquence, le nombre d'occurrences minimal de cette séquence n'est pas un paramètre critique pour la perplexité. Par ailleurs, la perplexité du modèle bigram évaluée sur le corpus de test est égale à 32.2. Cela peut être justifié par la complexité des modèles bigrams et multigrams. En effet, le nombre de bigrams est presque trois fois supérieur au nombre d'unités du modèle multigram de taille 2. Le tableau 1 donne le nombre d'unités des autres modèles.

4.5. Performances de reconnaissance

Modèle	Nombre d'unités	WER
bigram	9719	31.0%
multigram $m = 2$	3378	37.9%
multigram $m = 3$	5280	34.8%
multigram $m = 4$	6628	34.3%
multigram $m = 5$	7523	34.0%

TAB. 1: Performances de reconnaissance ngrams vs. multigrams.

Le tableau 1 donne les taux d'erreur mots des systèmes de reconnaissance intégrant des bigrams ou des multigrams de taille des séquences $m = \{2, 3, 4, 5\}$. Le système de reconnaissance de parole intégrant les multigrams donne des meilleures performances pour des tailles de séquence plus grande, passant de $WER = 37.9\%$ pour un système multigrams de taille 2 à $WER = 34.0\%$ pour un système multigrams de taille 5. Par comparaison, le système de reconnaissance intégrant les bigrams donne les meilleures performances $WER = 31.0\%$. Cela est dû au fait que les bigrams modélisent bien la parole spontanée où les phrases ne sont pas bien structurées.

4.6. Estimation des bigrams en utilisant les séquences multigrams

Une différence substantielle entre le modèle des bigrams et le modèle des multigrams réside dans les hypothèses sous-jacentes aux dépendances existant entre les mots au sein des phrases. Le modèle bigram représente

ces dépendances sous forme d'un chaînage alors que le modèle multigram les schématise comme concaténation de blocs indépendants. Dans ce contexte, il est intéressent de considérer un modèle bigram alternatif, pour lequel l'apprentissage des probabilités est dérivé de celles des modèles multigrams. Ce modèle de bigram dérivé peut s'obtenir soit par calcul direct, soit par l'intermédiaire d'une estimation de bigrams sur des énoncés artificiels issus du modèle de multigrams (méthode de Monte-Carlo). Une première expérience dans cette direction a consisté à estimer des bigrams sur les séquences du dictionnaire de multigrams (de taille 5). Le modèle ainsi obtenu contient 5267 unités et donne un taux d'erreur de mots de 32.1%. Cette valeur, intermédiaire entre les résultats obtenus par le bigram pur et par le multigram pur, tendent à indiquer qu'une partie des erreurs causées par le multigram proviennent des contraintes structurelles du modèle. Ces résultats sont en cours de consolidation par des évaluations plus complètes et plus précises.

4.7. Diagnostic des erreurs

Pour approfondir le diagnostic entre bigrams et multigrams, nous avons comparé les types d'erreurs produits par les deux modèles sur le corpus de test. A la sortie des deux systèmes, nous obtenons 886 phrases où les sorties sont identiques et 614 phrases où les sorties sont différentes. Le taux d'erreur mots sur les 886 phrases est $WER = 16.7\%$ pour les deux systèmes. Sur les 614 autres phrases, il atteint 41.3% pour le système intégrant des bigrams et 46.3% pour celui intégrant des multigrams. Le tableau 2 présente en détail les résultats du diagnostic. Les erreurs de substitutions qui diffèrent différentes dans

bigram $\rightarrow$ multigram $\downarrow$	OK	Sub	Omi	Ins	non Ins
OK	4407	150	53	-	-
Sub	245	689	60	-	-
Omi	72	65	240	-	-
Ins	-	-	-	304	360
non Ins	-	-	-	299	-

TAB. 2: Comparaison des erreurs de reconnaissance par type d'erreur entre les systèmes à base de bigrams et de multigrams.

les deux approches correspondent souvent à des mots très différents. Quant aux erreurs d'insertions différentes sont souvent des mots courts et proches (par exemple $d' \leftrightarrow de$, $j' \leftrightarrow je$, ...). Les mots reconnus par les bigrams et omis par les multigrams sont souvent des mots courts (*de, le , la, à, pour, euh,* ...) qui servent à structurer la phrase, comme si la sortie du système multigrams était une simple suite de blocs de mots. En revanche, les mots reconnus par les multigrams et omis par les bigrams sont souvent des noms ou des verbes. La relative discordance entre les deux méthodes invite à réfléchir à l'utilisation d'une stratégie de fusion pour combiner les deux approches. En effet, si l'on calcule les performances que pourrait obtenir un système qui fusionnerait optimalement les deux approches (en ayant la connaissance du système préférable pour chaque mot décodé), on évalue (grâce à la table 2), une $F\text{-}mesure$ de 80.6%, à rapprocher des performances du bigram seul (77.4%, cf. table 3) ou du multigram seul (75.3%). Il reste donc une marge de progression de plusieurs $\%$ qui peut être en partie comblée par des méthodes hybrides ou combinées. Cela constitue un objet intéressant pour des travaux à venir.

	WER	$\mathcal{P}$	$\mathcal{R}$	$F\text{-}mesure$
bigram	31.0%	75.8%	79.0%	77.4%
multigram $m = 5$	34.0%	73.5%	77.1%	75.3%

TAB. 3: $F\text{-}mesure$ pour les bigrams vs. multigrams.

5. CONCLUSION

Dans cet article, nous avons comparé les deux approches de modélisation de langage ngrams et multigrams. Les évaluations des systèmes de reconnaissance intégrant ces deux approches montrent que l'intégration d'un modèle bigram donne des meilleures performances de reconnaissance. Cela semble être dû au fait que les bigrams modélisent bien la parole spontanée où les phrases ne sont pas bien structurées. Pour des performances similaires, les multigrams présentent l'intérêt d'avoir un nombre réduit de paramètres. L'analyse diagnostique des erreurs concordantes et discordantes pour les deux méthodes et la marge de progression qu'il met en évidence incite à explorer des approches combinées alliant la souplesse des ngrams et la capacité de structuration des multigrams.

RÉFÉRENCES

[1] F. Bimbot and G. Gravier. Evaluation des systèmes de reconnaissance de la parole. In S. Chaudiron, editor, *Traité des Sciences et Techniques de l'Information*, pages 189–213. Hermès, 2004.

[2] F. Bimbot, R. Pieraccini, and B. Atal. Variable-length sequence modeling : Multigrams. In *IEEE Signal Processing Letters*, volume 2, pages 111–113, 1995.

[3] F. Bimbot, R. Pieraccini, E. Levin, and B. Atal. Modèles de séquences à horizon variable : Multigrams. In *JEP*, 1994.

[4] S. Deligne. *Modèles de séquences de longueurs variables, application au traitement du langage naturel et de la parole*. PhD thesis, ENST, 1996.

[5] S. Deligne and F. Bimbot. Language modeling by variable length sequences : Theoretical formulation and evaluation of multigrams. In *ICASSP*, pages 169–172, 1995.

[6] S. Deligne and F. Bimbot. *Multigram Package*. ENST, 1997.

[7] S. Deligne and F. Bimbot. Learning a syntagmatic and paradigmatic structure from language data with a bi-multigram model. In *International Conference on Computational Linguistics*, pages 300–306, 1998.

[8] M. Federico and R. De Mori. Spoken dialogue with computers. *Academic Press*, pages 202–210, 1998.

[9] I. H. Witten and T. C. Bell. The zero-frequency problem : estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 1991.

[10] I. Zitouni, K. Smaili, J.-P. Haton, S. Deligne, and F. Bimbot. A comparative study between polyclass and multiclass models. In *ICSLP*, 1998.