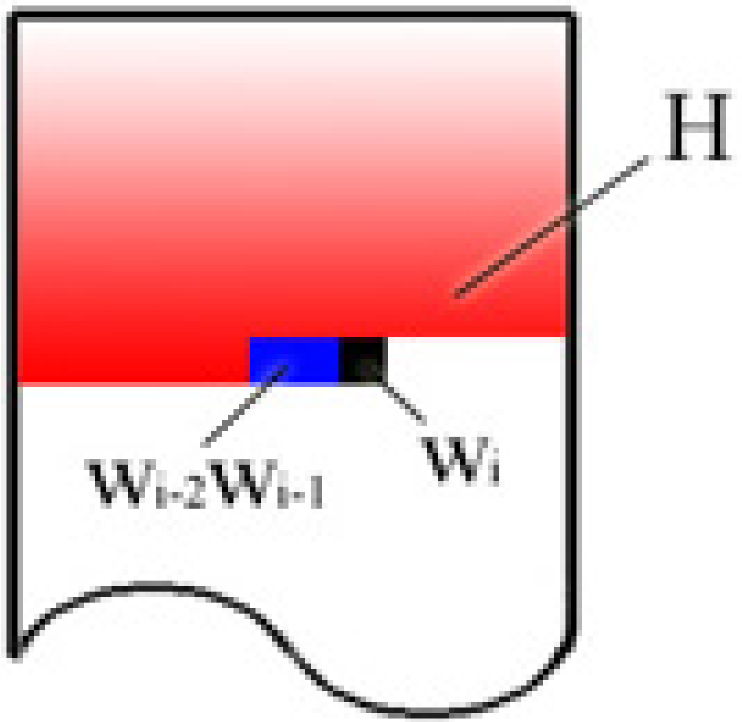
Modèles de langue adaptatifs

Laurent Nepveu

7 avril 2003

Modèles de langues adaptatifs



- On voudrait utiliser la totalité de H pour améliorer la prédiction de w.

Les modèles adaptatifs

- Le modèle cache
- Le topic-specific clustering
- (Le modèle avec triggers)

Modèle de langue cache

[Kuhn et de Mori, 1990]

LM = 3g-gram + composante cache

- La composante cache suit le même principe que les mémoires cache des ordinateurs.
- Le principe de localité : plusieurs accès mémoire consécutifs ont tendance a toujours accéder aux même blocs.
- De la même façon, lorsque l'on parle ou écrit, on a tendance à utiliser les mêmes mots d'une phrase à l'autre.

Motivation pour la composante cache

- Dans un système de reconnaissance de la parole, on analyse un son qui suit la séquence de mot *"the old"*.
- La composante acoustique retourne comme probabilité maximale :

$$p(man) \simeq p(band)$$

- Admettons que suite à l'entraînement d'un modèle trigram :

$$p(man|the\,old) = 0.1$$
$$p(band|the\,old) = 0.01$$

- Dans cette circonstance, un modèle de reconnaissance de la parole statique, optera pour *man*.

- Mais si, dans les quelques phrases qui précédaient cette analyse :

$$f(\textit{the old band}) \gg f(\textit{the old man})$$

- La prédiction devrait être *band*.
- On veut donc que le modèle avec cache utilise l'information contenue dans l'historique du texte analysé pour améliorer la prédiction.

Le modèle 3g-gram

$$p(w_i|g_{i-2}, g_{i-1}) = \sum_{g_j \in G} p(w_i|g_j) \times p(g_i = g_j|g_{i-2}, g_{i-1}).$$

$p(w_i\|g_i)$	unigram entraîné sur chaque POS
$p(g_i\|g_{i-2}, g_{i-1})$	trigram où $V = \{POS\}$

Le modèle combiné

- Le modèle cache n'affecte que l'unigram pour chaque POS. Le trigram sur les POS reste inchangé.

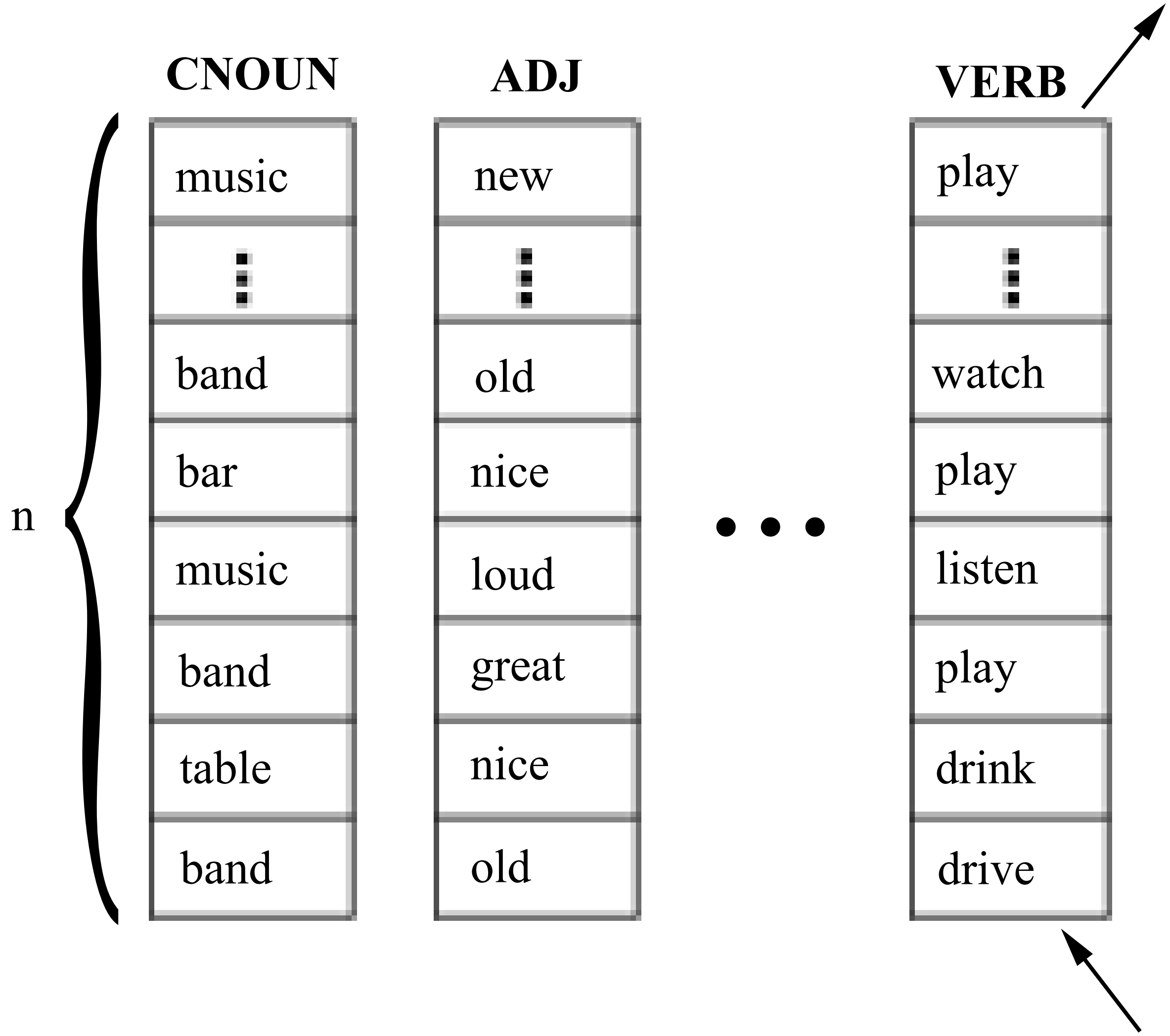
$$p(w_i|g_i = g_j) = k_{M,j} \times f(w_i|g_i = g_j) + k_{C,j} \times C_j(W, i)$$

où

$$k_{M,j} + k_{C,j} = 1$$

- On espère donc que la prédiction d'un mot étant donné le *part of speech* soit modifié par l'historique du texte analysé et favorise les mots qui reviennent dans cet historique.

La composante cache



Résultats de Kuhn et de Mori

- Kuhn et de Mori rapportent que la perplexité du modèle 3g-gram passe de 332 à 107 en lui ajoutant une composante cache.
- C'est bon, mais, à la base, le 3g-gram est plutôt mauvais.

Clarkson et Robinson

- Admettons un historique de longueur 200. Avec les caches conventionnelles, le mot à la position w_{i-200} a le même poids dans la cache que le mot en position w_{i-10}. Clarkson et Robinson pallie à ce problème en introduisant les caches à affaiblissement exponentiel.

$$p_{cache}(w_i|w_1 w_2 ... w_{i-1}) = \beta \sum_{j=1}^{i-1} I_{\{w_i = w_j\}} e^{-\alpha \times (i-j)}$$

- α est le taux d'affaiblissement, β est un facteur de normalisation et I une fonction indicatrice qui vaut 1 si $w_i = w_j$, 0 sinon.

Une cache bigram

- Martin, Liermann et Ney utilise un trigram de mots interpolé avec une cache bigram.

$$p(w_i|h) = \lambda\, p_{tri}(w_i|w_{i-2}w_{i-1}) + (1-\lambda)\, p_{cache}(w_i|w_{i-1}, h)$$

- λ est spécifié par validation.

Topic-specific clustering

[Martin, Liermann et Ney] et [Clarkson et Robinson]

- Plutôt que d'utiliser directement l'historique h pour modifier les probabilités, on l'utilise pour spéficier quel partie du corpus d'entraînement sert à créer le modèle utilisé.
- Étant donné le langage utilisé dans H, on tente de mieux spécifier le language utilisé dans le document.
- Fonctionne bien sous l'hypothèse où le corpus est divisé en sujets (topics).
- Peut entraîner des problèmes de sous-représentation de données étant qu'on utilise qu'une fraction du corpus à la fois.

Clustering du corpus d'entraînement

- Créer une partition du corpus d'entraînement où chaque ensemble représente de façon théorique un sujet précis.
- On entraîne un trigram sur chaque cluster.
- Le critère utilisé pour faire le clustering est log-vraisemblance d'un l'unigram sur chaque cluster (calculs).
- On maximise ce critère sur toutes les partitions.

$$F = \sum_{a=1}^{A} \sum_{n=1}^{N_a} log\ q(w_{a,n}|c(a))$$

- On utilise l'*exchange algorithm* pour réaliser le clustering.

- On utilise une mixture de modèles pour représenter le modèle résultant.

$$p(w_n|h_n) = \sum_{i=1}^{I} \lambda_n(i) \times p_i(w_n|h_n)$$

- $\lambda_n(i)$ est le paramètre d'interpolation du i^e modèle à la position n dans le texte.
- On recherche les meilleurs paramètres $\lambda_n(i)$ étant donné l'historique H par un algorithme EM.
- Théoriquement, les $\lambda_n(i)$ seraient calculés à chaque mot du document. Par contre, pour des raisons de capacité de calcul, on calcul les $\lambda_n(i)$ à tous les dixièmes du document.

- Une autre approche, plus simple, consiste à ne choisir le meilleur model.
- On calcul la log-vraisemblance de H étant donné chacun des modèles.

$$p(w_n|h_n) = p_i(w_n|h_n)$$

$$i = \underset{i}{argmax} \sum_{m=0}^{M-1} log\, p_i(w_{n-m}|h_{n-m})$$

Résultats

[Clarkson & Robinson]

nb. clusters	with full LM	Change	w/out full	Change
0 (baseline)	165.52	-	-	-
5	141.17	-14.7%	143.23	-13.5%
10	134.40	-18.8%	138.24	-16.5%
10(BNC)	139.24	-15.8%	142.87	-13.7%
20	129.09	-22.0%	134.42	-18.8%
30	126.77	-23.4%	134.14	-19.0%
40	127.52	-23.0%	135.48	-18.1%
50	125.78	-24.0%	134.91	-18.5%

- Il est difficile d'envisager plus de 50 pour des raisons de calculs.

Size	Perp.	Change
50	152.59	-7.8%
100	148.33	-10.4%
200	145.83	-11.9%
500	144.73	-12.6%
1000	144.98	-12.4%
2000	145.56	-12.1%

Decay Rate	Perp.	Change
0.000	144.98	-
0 001	143.7	-0.88
0.002	142.75	-1.53%
0.003	142.2	-1.92%
0.004	141.9	-2.12%
0.005	141.85	-2.16%
0.006	141.75	-2.23%
0.007	141.8	-2.19%
0.008	141.85	-2.16%
0.009	142.2	-1.92%
0.01	142.4	-1.78%

- Ils rapportent une perplexité de 115.86 (-30.0%) pour le modèle combiné.

Résultats

[Martin, Liermann et Ney]

	C	4M	Change	39M	Change
Word Trigram		152.1	-	96.8	-
+ Cache		120.5	-20.8%	82.8	-14.5%
+ Adaptation	10	130.9	-13.9%	81.2	-16.1%
	20	127.8	-16.0%	78.4	-19.0%
	30	126.7	-16.7%	76.7	-20.8%
+ Adaptation	10	111.5	-26.7%	73.7	-23.9%
+ Cache	20	110.4	-27.4%	72.0	-25.6%
	30	110.3	-27.5%	71.0	-26.6%
+ Adaptation	10	112.3	-26.2%	75.5	-22.0%
+ Cache	20	111.2	-26.9%	74.0	-23.6%
(max)	30	111.2	-26.9%	72.9	-24.7%

Conclusion

- L'addition d'une cache au modèle de langue permet de capturer des fluctuations à court terme dans les probabilités d'un langage.
- Le topic-specific clustering permet d'adapter et de préciser les probabilités d'un modèle étant donné le langage qu'on utilise présentement.
- Les deux approches sont non seulement viables, mais améliore grandement les résultats d'un modèle de langue.
- Dans le cas des caches, quelque soit le cas on peut considérer que c'est un amélioration.
- Dans le cas du clustering, il faut que le corpus soit "clusterable" et que les documents analysés soient de la même distribution de clusters.

Références

- Kuhn, R., De Mori, R., A Cache-Based Natural Language Model for Speech Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.
- Clarkson, P.R., Robinson, A.J., Language Model Adaptation using Mixtures and an Exponentially Decaying Cache, *Proceedings IEEE Int. Conference on Acoustics*, Speech and Signal Processing, Munich, 1997.
- Martin, S.C., Liermann, J., Ney, H., Adaptative Topic-Dependent Language Modelling Using Word-Based Varigrams, *in Proceedings of Eurospeech*, 1997.