



Master 2 ATAL

Applications Multilingues

---

# Segmentation en mots du Japonais

---

*Étudiante :*  
Coraline MARIE

*Encadrant :*  
Florian BOUDIN

3 novembre 2014



UNIVERSITÉ DE NANTES

## Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Première approche</b>	<b>2</b>
1.1 La segmentation par les Modèles de Markov cachés . . . . .	2
1.2 Gestion des probabilités . . . . .	2
<b>2 Implémentation des ngrammes</b>	<b>3</b>
2.1 Implémentation des trigrammes . . . . .	3
2.2 Trigrammes avec backoff . . . . .	3
2.3 N-grammes . . . . .	3
<b>3 Reconnaissance des alphabets</b>	<b>4</b>
3.1 Analyse des alphabets par bigrammes . . . . .	4
<b>Conclusion</b>	<b>4</b>
<b>Annexes</b>	<b>5</b>

## Introduction

Le Japonais est une langue très particulière, dont l'écriture diffère totalement des langues latines. L'utilisation d'alphabets différents et l'absence de marqueurs explicites, tels que les espaces, la rend plus délicate à traiter, notamment dans le cadre de la Recherche d'Informations.

Afin de faciliter le traitement de textes en langue japonaise, il faut au préalable appliquer un processus de segmentation en mots, appelée *tokenisation*. Ce rapport présente donc différentes méthodes combinables permettant de segmenter en mots, des textes en japonais. Il y aura d'abord une description de l'implémentation de la méthode des Modèles de Markov cachés de C. P. Papageorgiou[1]. Il y aura également une présentation de l'amélioration de cette méthode par l'implémentation de ngrammes, avant de finir par l'explication d'une méthode reposant sur l'analyse des différents alphabets utilisés par la langue japonaise.

## 1 Première approche

### 1.1 La segmentation par les Modèles de Markov cachés

L'article *Japanese word segmentation by hidden markov model* de Constantine P. Papageorgiou[1] présente une méthode de segmentation en mots du Japonais très efficace. Cette méthode plutôt simple, utilise à la fois un corpus d'entraînement déjà segmenté et un second corpus qu'il faudra segmenter.

La première étape de cette méthode consiste à analyser bigramme par bigramme tout le corpus d'entraînement, afin de mémoriser le comportement de chaque duo de caractères : *coupure* ou *non-coupure*. Cette analyse permet ensuite d'obtenir des probabilités de comportement sur l'ensemble des bigrammes rencontrés.

La seconde étape consiste à analyser bigramme par bigramme tout le corpus à segmenter, puis par l'intermédiaire d'un *HMM* (Hidden Markov Model : Modèle de Markov caché), elle définit s'il faut couper ou non le bigramme.

Les résultats de l'implémentation de cette méthode sur le corpus de test donnent un peu plus de 89,2% de f-mesure :

Avg Precision	0.904005681695
Avg Recall	0.881382517888
Avg f-measure	0.892550767507

### 1.2 Gestion des probabilités

La première limite qui est observable sur cette méthode est la gestion pauvre des probabilités non observées. En effet, dans le cas où un bigramme du corpus à segmenter n'a pas été observé dans le corpus d'entraînement, la méthode attribue la même probabilité à la coupure et à la non coupure du bigramme.

Or, si on regarde plus attentivement les probabilités de coupure des bigrammes dans le corpus d'entraînement, on remarque qu'il est plus probable de couper que de ne pas couper un bigramme. De plus, si le bigramme n'a pas été observé, c'est également possible que ce soit car il n'a aucun sens dans la langue japonaise.

Ainsi, la première amélioration faite à cette méthode est l'attribution de deux probabilités différentes à la coupure et à la non-coupure pour les bigrammes non rencontrés. Ces probabilités sont 0.02 pour la coupure et 0.01 pour la non-coupure. Elles ont été définies pour permettre d'avantager la coupure par rapport à la non-coupure pour les raisons décrites précédemment.

Les résultats de cette première amélioration donnent environ 92% de f-mesure :

Avg Precision	0.87553154073
Avg Recall	0.965715790723
Avg f-measure	0.918415054529

## 2 Implémentation des ngrammes

### 2.1 Implémentation des trigrammes

La méthode de C. P. Papageorgiou s'appuie sur une analyse bigramme par bigramme. Une autre amélioration possible serait donc de voir, si la méthode ne donnerait pas de meilleurs résultats en remplaçant les bigrammes par des trigrammes. Malheureusement en remplaçant le traitement des bigrammes par des trigrammes, les scores de segmentation sont beaucoup moins bons, avec seulement 85.3% de f-mesure :

Avg Precision	0.749078005914
Avg Recall	0.991451976152
Avg f-measure	0.85338934337

### 2.2 Trigrammes avec backoff

L'implémentation des trigrammes donne de moins bons résultats que ceux des bigrammes seuls, car il y a plus de trigrammes non reconnus. Pour résoudre ce problème, il suffit de fusionner la méthode des bigrammes avec celle des trigrammes. Ainsi, lorsqu'un trigramme n'est pas reconnu, la méthode teste un caractère de moins, pour voir si un bigramme est reconnu. Cette nouvelle méthode de trigrammes avec backoff donne de bien meilleurs résultats sur le corpus de test avec 92.5% de f-mesure :

Avg Precision	0.878153290557
Avg Recall	0.977583661324
Avg f-measure	0.925204736713

### 2.3 N-grammes

Après avoir obtenus de très bon résultats avec la méthode précédente, il paraît logique d'essayer avec 4-grammes et plus. Cependant cette nouvelle mé-

thode est très décevante, car elle donne les pires résultats rencontrés jusqu'à présent : environ 50% de f-mesure pour 4-grammes avec backoff. Cette méthode a donc été abandonnée car le taux de f-mesure est trop bas.

### 3 Reconnaissance des alphabets

#### 3.1 Analyse des alphabets par bigrammes

Comme il l'a été dit précédemment, le Japonais est composé de plusieurs alphabets : les Hiraganas, les Katakanas, les Kanjis, et les Romajis. De plus, il est possible de rencontrer divers éléments de ponctuations ainsi que des caractères spéciaux. Aussi, il est logique de penser que lorsqu'il y a plusieurs alphabets dans une même phrase, ces derniers peuvent donner des indices sur la segmentation.

Dans cette nouvelle méthode, la première étape analyse d'abord les bigrammes et les trigrammes, puis elle calcule des probabilités sur les alphabets. Ainsi, lors de la seconde étape, si un bigramme n'a pas été rencontré dans le corpus d'entraînement, elle applique les probabilités calculées avec les alphabets. Cette nouvelle fonctionnalité donne de très bon résultats, puisqu'il y a un taux d'environ 94.6% de f-mesure :

Avg Precision	0.957560380247
Avg Recall	0.934352408883
Avg f-mesure	0.945814049226

### Conclusion

Ce projet a permis de délimiter les performances et les limites de la méthode de C. P. Papageorgiou, et de concevoir quelques améliorations :

Méthode	Précision	Rappel	F-mesure
Baseline	90.0%	88.0%	89.2%
Probabilités unseen	87.5%	96.5%	91.8%
Trigrammes + backoff	87.8%	97.7%	92.5%
Gestion des alphabets	95.7%	93.4%	94.6%

Bien que les scores obtenus grâce à ces diverses méthodes soient satisfaisants (94.6% de f-mesure sur le corpus de test), il reste encore des erreurs. Ceci peut être reproché au corpus d'entraînement dont la taille reste limitée, mais pas seulement.

Il y a également eu pendant ce projet, d'autres idées qui n'ont pas eu le temps d'être implémentées, comme par exemple :

- le traitement des alphabets par n-grammes ;
- l'ajout d'un dictionnaire : ce qui devrait éliminer les groupes de caractères inexistants ;
- l'entraînement du modèle de Markov caché sur d'autres types de corpus tokenisés (Wikipédia, revues de presse, ...).

## Annexes

Le projet est actuellement disponible sur un dépôt GitHub à l'adresse suivante : [https://github.com/Slayerxoxo/AM\\_segmentation\\_du\\_japonais](https://github.com/Slayerxoxo/AM_segmentation_du_japonais)

### Compilation :

Commande pour construire les ensembles de train/test à partir du KNBC :

```
python knbc_to_xml.py knbc-train.xml knbc-test.xml knbc-reference.xml
```

Commande pour segmenter le fichier de test avec mon implémentation de hmm :

```
python hmm_segmenter.py knbc-train.xml knbc-test.xml knbc-hmm.xml
```

Commande pour évaluer la performance d'un système :

```
python evaluation.py knbc-hmm.xml knbc-reference.xml
```

## Références

- [1] Constantine P. Papageorgiou. Japanese word segmentation by hidden markov model. *Proceedings of the workshop on Human Language Technology, HLT '94*, 1994.