Comparaison des temps de traitement de corpus en japonais par Sagace et un système basé sur MeCab

Raoul Blin

Comparaison des temps de traitement de corpus en japonais par Sagace et un système basé sur MeCab

Raoul Blin
CRLAO, CNRS, 131 Bd St.Michel, 75006 Paris
blin@ehess.fr

Résumé. Ce texte figurait dans la première version de l'article court (Blin 2014). Il a été retiré de la version finale par manque de place. L'objectif est de comparer les temps de traitement de corpus japonais et par Sagace et un système basé sur Mecab.

Abstract. This text was originaly part of the short paper (Blin 2014) . It has been removed from the final version because of lack of space. The purpose is to compare the processing time of a japanese corpus by Sagace and a system based on MeCab.

Mots-clés : Japonais, Corpus, Analyseurs, MeCab, Sagace
Keywords: Japanese, Corpus, Comparison, MeCab, Sagace

1 Expériences et résultats

(Les outils et les ressources utilisés pour les expériences, les spécificités du japonais et les conséquences sur les résultats sont présentés en détails dans (Blin 2014)).

Nous mesurons les vitesses de traitement et les ressources mémoire utilisées. Nous prenons en compte le temps d'analyse par MeCab alors que dans l'usage, Mecab, qui est essentiellement utilisé en TAL, est appliqué une fois pour toute à un texte. Les analyses ont lieu alors sur le texte segmenté et éventuellement balisé. Mais la pratique en linguistique est différente. Le linguiste travaille plus en amont sur la défintiion même des catégories. Dès lors, il est amené à vouloir modifier les catégories ou ces morphèmes. Cela implique une modification du dictionnaire et une réanalyse du texte. Pour être complet, il faudrait même réentrainer un analyseur statistique de type MeCab sur un corpus retravaillé. Nous n'allons pas aussi loin et nous en tenons à prendre en compte ici seulement le temps d'analyse, en ne mesurant pas le temps de modification du corpus d'entraînement.

1.1 Temps d'exécution

Nous comparons les temps d'exécution de Sagace et du dispositif MeCab pour trois tâches, parmi les plus courantes en linguistique : recherche d'un mot donné, recherche de n'importe quel mot appartenant à une catégorie spécifique, recherche d'une chaîne composée de plusieurs mots. Une caractéristique des recherches contemporaines en linguistique est de s'appuyer sur l'analyse de corpus de grande taille. Pour tester les systèmes dans des conditions réalistes, nous avons utilisé un corpus de taille raisonnable de 2 Go. Il est constitué de déclarations de brevets[1]. Le temps d'exécution est mesuré à l'aide de la commande `time`. Pour toutes les opérations, on s'est assuré que les dispositifs accédaient au disque un nombre de fois égal.

[1] Extraits du site : http://www.patentjp.com

1.1.1 Recherche d'un morphe spécifique

A titre de référence, nous avons comparé les temps d'exécution avec Sagace, avec un dispositif MeCab et avec une combinaison de commandes shell. La recherchepour porte sur la particule casuelle *wo* , dont la particularité est d'être le seul mot dans la langue écrite contemporaine à contenir le caractère を. Autrement dit, aucune erreur d'analyse n'est possible, quel que soit le dispositif.

Sagace étant conçu pour effectuer ce type de tâche, il n'y a pas de manipulation particulière à effectuer. MeCab a été combiné avec `grep`. La commande pour MeCab est[2] :

```
mecab -d $DICO -b 20000 $CORPUS | grep -c "を"> fichier_de_resultats
```

La combinaison de commande du shell est :

```
sed s/を/を\\n/g $CORPUS | grep -c "を" > fichier_de_resultats
```

Comme on peut s'y attendre, pour ce type de tâche, les temps d'exécution (Table 1) sont défavorables pour le dispositif MeCab qui fait inutilement l'analyse de la phrase complète. Sagace exécute aussi des contrôles inutiles, ce qui le ralenti aussi.

	Temps d'exécution (en mn:s)	Taille max. du processus en mémoire physique (en Ko)
`grep+sed`	1:54	1 296
Sagace	4:16	1 292
MeCab+Ipadic	7:01	44 224
MeCab+Jumandic	6:48	43 416
MeCab+Unidic	17:53	104 468

TABLE 1 : Temps de comptage des occurrences de la particule *wo*.

1.1.2 Comptage des occurrences d'une catégorie spécifique

Avec le même corpus, nous comparons cette fois-ci les temps d'exécution pour compter le nombre d'occurrences des mots d'une catégorie spécifique. La charge de travail pour Sagace varie sensiblement en fonction de la taille de la catégorie. Plus la catégorie est petite, plus la charge est faible et proche de celle d'une recherche de chaîne spécifique (voir section précédente). Pour comparer Sagace à un dispositif MeCab, il est préférable d'effectuer la recherche sur deux catégories de tailles sensiblement différentes. Nous avons choisi la volumineuse catégorie des noms communs, et la petite catégorie des connecteurs de phrases (*setuzokusi*).

Pour le dispositif MeCab, nous utilisons les catégories prédéfinies de noms communs présentes dans les trois dictionnaires. Pour Sagace, la catégorie des noms communs regroupe les 90 000 entrées lexicales marquées « nom commun » (*meisi, ippan*) du dictionnaire Mecab-naist-jdic-0.4.3-20080812 [3]. Il existe des différences dans la définition des entrées lexicales. Nous estimons que cela n'affecte pas plus de 10 % des entrées. La catégorie des *setuzokusi* comprend 40 entrées, toutes présentes dans les trois lexiques ainsi que dans le vocabulaire du corpus étudié. Contrairement à la tâche de recherche de chaîne spécifique (section précédente), Sagace doit cette fois-ci compiler les noms communs sous forme d'arbre lexical avant de procéder à l'analyse. Cela le ralentit.

---

[2] L'option -d spécifie le dictionnaire à utiliser ; l'option `-b` règle la taille du tampon mémoire, dont la valeur par défaut est très insuffisante. En sortie, MeCab affiche par défaut chaque lemme sur une ligne. Le nombre d'occurrences du lemme cherché est donc égal au nombre de lignes contenant ce lemme. C'est ce qui justifie que l'on utilise la commande `-c` de `grep`.

[3] Dans les dictionnaire Ipadic, Jumandic et Unidic, les morphes de cette catégorie peuvent être classés pareillement (*meisi-ippan*) ou comme « noms communs » (*hutuumeisi*).

Sagace étant conçu pour ce type de requête, il suffit de configurer le fichier de requête pour la circonstance. Avec MeCab, le comptage s'est fait à l'aide des commandes `grep`, `sort` et `uniq`. La commande pour MeCab prend la forme[4] :

```
mecab -d $DICO -b 20000 $CORPUS | grep $CRITERE  >> fSauvegardeForcee
LC_ALL=C sort fSauvegardeForcee | uniq -c | LC_ALL=C sort -nr >
resuAnalyseMecab1
```

La variable $CRITERE permet de sélectionner exclusivement les noms communs et les connecteurs. Elle diffère d'un lexique à l'autre.

	Simple analyse morphologique		90 000 noms communs		40 connecteurs	
	temps (mn:s)	mem.max (Ko)	temps	mem.max	temps	mem.max
Sagace	-	-	16:51	**65 924**	**5:27**	**1 300**
MeCab Ipadic	6:58	43 944	9:45	132 352	7:13	494 584
MeCab Jumandic	**6:33**	**43420**	**9:20**	131324	7:03	282 076
MeCab Unidic	17:12	104 468	22:04	149 216	17:52	494 588

TABLE 2 : Temps de comptage des occurrences des mots de deux catégories spécifiques.

Pour les temps d'exécution (Table 2), la combinaison MeCab et Ipadic est globalement de loin la meilleure. Certes, Sagace reprend l'avantage sur la recherche d'une petite catégorie mais dans l'ensemble, on sait qu'avec Sagace les erreurs de segmentation peuvent être importantes selon les mots. Les temps de correction pourraient être tels que le faible avantage de Sagace pour les petites catégories ne présenterait pas grand intérêt. Sagace est donc intéressant pour les petites catégories sous réserve de manipuler des mots peu sensibles aux erreurs d'analyse, c'est à dire à dire ceux écrits en sinogrammes et de plus d'un caractère.

1.1.3 Comptage de chaînes de trois éléments contigus

Dans ce test, nous évaluons la vitesse d'exécution du comptage d'un motif de mots contigus. Sagace permet aussi d'extraire des motifs discontinus mais nous n'aborderons pas cette fonction ici. L'opération ne nécessite pas d'instructions particulières pour Sagace, si ce n'est de décrire le patron dans le fichier de requête. Pour optimiser l'opération avec Mecab, il a fallu créer un programme ad-hoc (en C). La sortie de MeCab est bufferisée et traitée au fur et à mesure :

```
mecab -d $DICO -b 20000 $FICHIER | ./denombreOccPatron
```

Le temps de traitement (voir Table 3) est légèrement inférieur pour MeCab avec Ipadic et Jumandic. Cela peut s'expliquer entre autres par la nécessité avec Sagace de compiler le lexique. On observe une différence notoire dans les résultats. Cela est facilement justifié pour Sagace par rapport à MeCab, entre autres parce que Sagace rejette les phrases dépassant une longueur donnée, alors que celles-ci sont prises en compte par MeCab. Elle s'explique moins pour MeCab et les différents dictionnaires. Une étude plus détaillée serait nécessaire pour expliquer ce dernier point.

	temps (mn:s)	mem.max (ko)	Nb d'occurrences
Sagace	6:46	51 444	10 068 630
MeCab Ipadic	6:26	44 224	17 591 975
MeCab Jumandic	**6:16**	**43 416**	**13 937 724**
MeCab Unidic	19:13	104 468	16 299 330

TABLE 3 : Temps de comptage des occurrences des patrons <particule+nom commun+particule> ; environ 90 000 noms communs et entre 10 et 20 particules .

[4] L'assignation LC_ALL=C est nécessaire pour que sort gère correctement les caractères japonais.

1.2 Mesure des tailles de corpus

La taille des corpus est en générale mesurée en nombre de mots. Dans de nombreuses langues, il s'agit du mot « graphique », délimité par des séparateurs graphiques aisément reconnaissables : espaces, signes de ponctuation. Dans une langue écrite où les mots ne sont pas séparés par des espaces, l'analyse est plus difficile. En japonais, où c'est le cas, l'unité de mesure est en général le lemme. Mais cela suppose une analyse morphologique des phrases. Le résultat sera aussi très dépendant de la définition des lemmes, qui peut différer d'un lexique à l'autre.

D'autres unités sont possibles mais toutes ont des avantages et inconvénients et c'est certainement la finalité des comptages qui permettra de choisir. Les caractères sont une première alternative. Leur intérêt est d'être graphique et facile à repérer. Cependant, en japonais, il existe pour la plupart des mots plusieurs graphies possibles. Les différentes graphies peuvent avoir un nombre de caractères, et donc une longueur, différente. Le caractère n'est donc pas un critère fiable pour comparer des longueurs de textes en japonais. Une autre alternative est la phrase, dont l'intérêt (en japonais contemporain) est d'être marquée par un délimiteur graphique aisément repérable. L'inconvénient est que la longueur des phrases (... en nombre de lemmes) peut sensiblement varier d'un texte à l'autre.

Pour cette expérience, nous travaillons sur un corpus dont il existe une version segmentée manuellement. Il s'agit d'un sous-corpus du Balanced Corpus of Written Japanese, version 2009 (Maekawa, 2009). Sa segmentation a été faite en s'appuyant sur le lexique UniDic 1.3.12 . Nous comparons la taille (en nombre de morphes) calculée à partir de la version lemmatisée manuellement, et la longueur calculée à partir de la version lemmatisée automatiquement par MeCab. Pour ce test, Sagace est inadapté et n'est pas utilisé.

Sagace	MeCab + Ipadic	MeCab + Jumandic	MeCab + Unidic	Analyse manuelle
(inapproprié)	825 073	765 056	833 038	934 654

TABLE 4 : Mesure de la taille du corpus de référence en nombre de morphes.

L'écart maximal entre la mesure manuelle et la mesure automatique vaut 137 307, soit 14,69% de la taille de référence (obtenue manuellement), ce qui est loin d'être négligeable. On voit que même la lemmatisation automatique à l'aide du lexique UniDic n'améliore pas sensiblement les résultats. La différence s'explique par la défintiion des morphes qui varie pour les morphes fonctionnels (conjugaison etc). Il serait possible d'unifier les analyses mais cela demanderait un outil supplémentaire, basé sur un analyseur conçu par un humain et à base de règles. Dans ce cas, le recours à des analyseurs statistique perdrait de son intérêt.

2 Conclusion

Si l'on ne tient pas compte du temps et de l'investissement pour entraîner les analyseurs statistiques, la comparaison de l'ensemble des temps de traitement de corpus de japonais par l'analyse Sagace et un système basé sur l'analyseur morphologique statistique MeCab ne fait pas apparaître de différences nette entre les deux dispositifs. L'avantage de l'un ou l'autre dépend du type de chaîne à chercher.

Remerciements

Je remercie Pierre Marchal (ERTIM - INALCO) pour les nombreux détails techniques sur les commandes shell.

Références

Blin, Raoul
2014 Comparaison de deux outils d'analyse de corpus japonais pour l'aide au linguiste, Sagace et MeCab. *In* Actes TALN-RECITAL 2014 P. 497. Marseilles, France. http://hal.archives-ouvertes.fr/hal-01054370.

Maekawa, Kikuo

2009 Daiyousei Wo Yû Suru Daikibo Nihongo Kakikotoba Kôpasu (<tokushyû> Nihongo Kôpasu) [Compilation D'un Corpus Équilibré de Textes Contemporains En Japonais. Journal of Japanese Society for Artificial Intelligence 24(5): 612–622.