21ème Traitement Automatique des Langues Naturelles, Marseille, 2014

Comparaison de deux outils d'analyse de corpus japonais pour l'aide au linguiste, Sagace et Mecab

Résumé. L'objectif est de comparer deux outils d'analyse de corpus de textes bruts pour l'aide à la recherche en linguistique japonaise. Les deux outils représentent chacun une approche spécifique. Le premier, Sagace, recherche un patron sans prise en compte de son environnement. Le second, un dispositif à base de Mecab, recherche les patrons après analyse morphologique complète des phrases. Nous comparons les performances en temps et en précision. Il ressort de cette analyse que les performances de Sagace sont globalement un peu inférieures à celles des dispositifs à base de Mecab, mais qu'elles restent tout à fait honorables voire meilleures pour certaines tâches.

Abstract. The purpose is to compare two tools used for helping linguist to analyze large corpora of raw japanese text. Each tool is representative of a specific approach. The first one, Sagace, search a pattern without taking into account its distribution. The second one is based on the morphological analyzer Mecab. It first analyzes the whole sentence before counting the searched pattern. We compare the processing time, needed ressources, and the quality of the results. It appears that performances of Sagace are globaly slightly lower than the Mecab system, but it doesn't defer so much. It may even be punctually better.

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

Le linguiste travaillant sur le japonais écrit contemporain dispose de nombreux outils d'analyse de corpus, que ce soit pour dénombrer des collocations ou constituer des concordanciers. On peut séparer ces outils en deux groupes. Le premier rassemble les outils qui s'en tiennent à chercher un patron dans du texte brut, sans contrôle sur l'environnement de ce patron. Les outils du second groupe procèdent en deux temps, avec en général une combinaison de logiciels : un premier lemmatise le texte complet, le second procède aux analyses statistiques.

On peut s'attendre à ce que la première approche, plus légère, soit plus performante en termes de rapidité, de ressources requises, et de mise en œuvre. Mais avec une langue comme le japonais qui ne dispose que de très peu de séparateurs de mots non ambigus, et en l'absence d'une analyse complète de phrases, les erreurs d'analyse sont sensiblement plus nombreuses qu'avec un outil qui effectue préalablement une analyse complète des phrases. Du coup, le gain en temps de traitement pourrait être perdu en temps de correction. Pour choisir l'approche et l'outil le plus efficace, il est nécessaire de comparer les performances qualitatives et quantitatives des deux approches dans les différentes tâches usuelles en linguistique. C'est l'objectif du présent travail. Nous ne tenons pas compte ici de l'ergonomie (facilité de mise en œuvre, interfaces etc.).

Dans la première partie, nous présentons les outils retenus pour le comparatif, Sagace et un dispositif à basé sur l'analyseur morphologique Mecab. Nous présentons aussi les lexiques utilisés dans les comparatifs. Dans la deuxième partie, nous présenterons plusieurs tests. Un premier ensemble de tests permet de comparer la vitesse d'exécution et la quantité de ressources mobilisées. Dans un second ensemble de tests, nous étudierons le taux d'accord entre les résultats obtenus par analyse automatique de corpus à l'aide des deux dispositifs, et les résultats obtenus à partir de la segmentation manuelle de ces corpus.

1 Présentation des dispositifs d'analyse utilisés pour la comparaison

Pour comparer les deux approches, avec ou sans analyse complète de la phrase, nous avons choisi Sagace et un dispositif basé sur l'analyseur morphologique Mecab. Les deux ont en commun de prendre du texte brut en entrée. Ils sont en ligne de commande (pour être facilement intégrables à d'autres dispositifs), et sont open source. Dans cette section, nous présentons leurs principales caractéristiques. Comme les performances des dispositifs sont très dépendantes du contenu des lexiques associés, nous consacrerons une sous-section aux lexiques utilisés pour les tests.

1.1 Sagace

Parmi les dispositifs reposant sur le principe de la recherche de patron dans du texte brut sans contrôle sur l'environnement, ceux qui utilisent le caractère comme unité sont d'un intérêt très limité pour le linguiste. Il existe d'autres outils, qui prennent comme unité de description des « mots ». Nous avons choisi comme représentant de cette approche l'outil Sagace.

Sagace (Blin, 2012) est conçu dès l'origine comme un outil d'aide à la recherche en linguistique. Il est utilisé pour le japonais depuis plus d'une décennie mais c'est un outil générique exploitable pour n'importe quelle langue, à cette réserve près qu'il est certainement plus adapté aux langues faiblement flexionnelles. Il peut produire un concordancier ou lister et compter des collocations.

Il se veut plus adapté au linguiste en permettant une manipulation très rapide et souple des lexiques. En effet, le linguiste a souvent besoin de modifier ses systèmes de parties du discours pour les tester, et de remodeler le lexique en conséquence. Mais modifier le lexique, qui peut contenir des centaines de milliers de mots, est un travail lourd. Pour contourner cette difficulté, le logiciel dispose d'un langage de description des catégories (de type langage propositionnel) qui permet de décrire les patrons à chercher en créant « à la volée » des nouvelles catégories, sur la base des catégories existantes dans le lexique, et sans intervenir dans celui-ci.

Une autre caractéristique pensée pour le linguiste est qu'il se veut facile à mettre en œuvre. Pour cela, il est tout en un : il contient une interface d'interrogation, un moteur de recherches et un compilateur de dictionnaire. Il ne nécessite pas de logiciels externes (comme c'est le cas pour Mecab qui ne fait pas les comptages), il ne nécessite pas de segmentation préalable (comme c'est le cas de Himawari (Yamaguchi, 2011)), il ne nécessite pas non plus d'entraînement comme les logiciels statistiques.

1.2 Dispositif basé sur Mecab

Les dispositifs qui reposent sur une lemmatisation préalable sont des assemblages basés sur un analyseur morphologique et un outil de comptage. A notre connaissance, il n'existe pas de dispositif « tout en un ». Pour des questions pratiques, dans le présent article, nous ne pouvions pas évaluer les performances de toutes les combinaisons possibles de tous les analyseurs morphologiques et outils de comptage. Nous avons opté pour un seul lemmatiseur-analyseur morphologique, associé à des outils de comptage *ad hoc*.

Pour des raisons de disponibilités et de réputation, nous avons choisi l'analyseur morphologique statistique Mecab[1], parmi les plus utilisés du moment. Il est disponible pour plusieurs distributions Linux, ainsi que l'OS de Apple. Le logiciel est en plus distribué avec plusieurs lexiques libres eux aussi.

Les outils de traitement des sorties de lemmatisation se répartissent en deux groupes : des outils dédiés (ex. Himawari) et des outils génériques (ex. : sed, grep). Ce sont ces derniers qui ont été retenus pour le comparatif car ils sont libres et faciles d'accès (installés en standard dans tous les systèmes) et qu'ils sont réputés performants. Ils ont en plus l'intérêt d'être en ligne de commande et d'être donc combinables avec d'autres outils.

Pour simplifier la présentation, désormais, nous désignerons le dispositif à base de Mecab « dispositif Mecab ».

1.3 Le lexique de travail

Les performances des dispositifs d'analyse varient en fonction des lexiques. C'est la raison pour laquelle nous avons mené les tests avec plusieurs lexiques qui diffèrent par le nombre d'entrées et la stratégie de lemmatisation. Ils sont au nombre de trois : Ipadic, Jumandic et Unidic. Nous avons en plus eu recours à un lexique limité aux noms communs qui appartiennent à ces trois lexiques.

UniDic [2] contient 756 463 entrées lexicales, noms propres compris, ce qui en fait le plus volumineux lexiques parmi ceux utilisés pour le présent travail. C'est une des versions de ce lexique qui a servi de référence pour la segmentation manuelle d'un corpus que nous utiliserons pour l'analyse qualitative (voir sections 2.2 et 2.3).

A coté de cela, les concepteurs et mainteneurs de l'analyseur morphologique Mecab recommandent un second lexique, Ipadic[3] . La définition des lemmes y est beaucoup plus libre que dans Unidic. Il contient 392 126 entrées

[1] http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

[2] http://sourceforge.jp/projects/unidic/

[3] Pour le présent travail, nous avons utilisé Mecab-Ipadic.x86_64 ; 2.7.0.20070801-6.fc18.1 dans la distribution Fedora 18 de Linux

Enfin, il faut compter avec un troisième lexique, Jumandic[4]. Comme son nom l'indique, il est à la base conçu pour un autre analyseur, JUMAN (Kurohashi et al. 1994). Il comprend 515 996 entrées.

Dans ces trois lexiques figurent diverses informations morpholexicales et syntaxiques, ainsi que les « poids », utilisés pour la lemmatisation statistique. Certaines entrées apparaissent plusieurs fois, avec des graphies différentes.

Avec Sagace, nous avons utilisé pour le présent travail une version modifiée de mecab-ipadic[5]. Elle compte plus de 90 000 noms communs. Nous décrirons ce lexique dans la section 2.1.2.

2 Expériences et résultats

Nous avons comparé Sagace et un dispositif Mecab pour l'analyse de corpus de textes bruts sur les points suivants : la vitesse d'exécution, quantité de ressources utilisées et taux d'accord avec une analyse manuelle de ce texte.

2.1 Temps d'exécution et ressources requises

Nous comparons les temps d'exécution de Sagace et du dispositif Mecab, ainsi que les ressources requises, pour trois tâches : recherche d'une chaîne spécifique, recherche de n'importe quel mot appartenant à une catégorie spécifique, recherche d'une chaîne composée.

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 donc utilisé un corpus de taille raisonnable de 2 Go. Il est constitué de déclarations de brevets[6].

Les temps d'exécution et les quantités de ressources requises sont mesurés à l'aide de la commande time du shell.

2.1.1 Recherche d'un morphème spécifique

Nous avons comparé les temps d'exécution et la mémoire requise avec Sagace et un dispositif Mecab pour chercher la particule casuelle *wo* dans le corpus. La particularité de ce morphème est d'être le seul dans la langue écrite contemporaine à contenir ce caractère. Autrement dit, aucune erreur d'analyse n'est possible, quel que soit le dispositif. Cela signifie que si les mesures diffèrent, ce n'est pas parce que les sorties sont différentes.

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` et une sauvegarde sur le disque a été imposée pour mettre le dispositif à égalité avec Sagace, qui fait nécessairement une sauvegarde disque. Le nombre d'accès au disque est donc identique pour tous les dispositifs. cependant, la taille du fichier sauvegardé étant très réduite, cette contrainte est certainement de peu d'effet sur le temps d'exécution. La commande pour Mecab est :

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

Le temps d'exécution du dispositif Mecab a été mesuré pour les trois dictionnaires Ipadic, Jumandic et Unidic.

A titre de comparaison, nous avons aussi fait la recherche avec une combinaison de sed et grep

```
sed s/を/を\\n/g ../corpus/brut/* | grep -c "を" > resuAnalyseGrep
```

Comme on peut s'y attendre, pour ce type de tâche, les temps d'exécutions et les ressources requises sont défavorables pour le dispositif Mecab (voire même très défavorables avec Unidic), que ce soit en temps d'exécution ou en ressource. Il est évident que le dispositif Mecab n'a guère d'intérêt pour la recherche d'une chaîne spécifique, non discontinue, et a fortiori non susceptible de provoquer des erreurs d'analyse.

---

[4] Pour le présent travail, nous avons utilisé Mecab-Jumandic.x86_64, ref ; 5.1.20070304-7.fc18 dans la distribution fedora 18)

[5] mecab-naist-jdic-0.4.3-20080812

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

	Temps d'exécution (en mn:s)	Taille max. du processus en mémoire physique (en Ko)
grep+sed	1:54.47	1 296
Sagace	4:16.79	1 292
Mecab+Ipadic	7:01.42	44 224
Mecab+Jumandic	6:48.28	43 416
Mecab+Unidic	17:53.59	104 468

TABLE 1 : Temps (secondes) de comptage des occurrences de la particule *wo* et occupation mémoire maximale (en Ko) Mesure à l'aide de la commande « time ».

2.1.2 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 variera sensiblement en fonction de la taille de la catégorie. Plus la catégorie est petite, plus la charge de travail est faible et proche de celle d'une recherche de chaine 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. Certaines entrées lexicales peuvent légèrement différer d'un lexique à l'autre, mais nous estimons que les résultats sont très peu affectés.

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[7]. Cela regroupe en gros tous les noms communs usuels avec des redondances comme pour les trois autres lexiques, et quelques sigles comme (株, *kabu*) (« SA »). Cela correspond à peu près aux dictionnaires de noms communs de Ipadic, Jumandic et Unidic. Nous avons exclu les dictionnaires spécialisés (mots de la CIM par exemple) , utilisés dans la version publique en ligne de Sagace par exemple et qui ne figurent pas dans les trois dictionnaires associés à Mecab. 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 le lexique (qui se limite aux 90 000 noms communs cherchés) avant de procéder à l'analyse. Cela le ralentit.

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. Pour mettre Sagace et Mecab sur un pied d'égalité en terme d'accès au disque, on a imposé au dispositif Mecab une sauvegarde sur disque. La commande pour Mecab prend la forme :

```
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. Avec cette sélection, seuls les mots comptés sont enregistrés, ce qui réduit le nombre d'accès aux disques.

	Simple parsage		90 000 noms communs		40 connecteurs	
	temps	ressource	temps	ressource	temps	ressources

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

Sagace	-	-	**16:51**	65 924	**5:27**	1 300
Mecab Ipadic	6:58.65	43 944	**9:45**	132 352	**7:13**	494 584
Mecab Jumandic	6:33.58	43420	**9:20**	131324	**7:03**	282 076
Mecab Unidic	17:12.04	104 468	**22:04**	149 216	**17:52**	494 588

TABLE 2 : Temps de comptage des occurrences des mots d'une catégorie spécifique (les noms communs et les connecteurs) . Temps en minutes, ressources en octets.

En terme de ressources, Sagace sort en tête du comparatif.

Pour les temps d'exécution, 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 lemmatisation peuvent être importantes selon les mots. Les temps de corrections 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.

2.1.3 Comptage de chaînes à 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 l'opération, nous avons conçu un programme simple avec *pipe*. La sortie de Mecab est bufferisée et traitée au fur et à mesure :

```
mecab -d $dico -b 20000 $fichier |  ./denombreOccPatron
```

Les résultats se présentent comme suit :

	temps	ressource	Nb d'occurrences
Sagace	**6:46.71**	51444	10 068 630
Mecab Ipadic	**6:26.60**	44224	17 591 975
Mecab Jumandic	**6:16.63**	43416	13 937 724
Mecab Unidic	**19:13.31**	104468	16 299 330

TABLE 3 : Temps (secondes) de comptage des occurrences des motifs contigus de particule+nom commun+particule ; environ 90 000 noms communs et entre 10 et 20 particules . Temps en minutes, ressources en octets.

Le temps de traitement est légèrement inférieur pour Mecab avec Ipadic et Jumandic. Cela peut s'expliquer entre autres par la nécessité avec Sagace de compiler le dictionnaire de noms. On observe une différence notoire dans les résultats. Cela est facilement justifié pour Sagace par rapport à Mecab (entre autres les chaînes dépassant une certaine longueur sont rejetées par sagace), elle s'explique moins pour Mecab et les différents dictionnaires. Une étude plus poussée serait nécessaire pour expliquer ce dernier point.

2.2 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 complète 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 Balance 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 lemmes) 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	(inapproprié)
Mecab + Ipadic	825 073
Mecab + Jumandic	765 056
Mecab + Unidic	833 038
Analyse manuelle	934 654

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

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.

2.3 Taux d'accord d'extraction

Dans cette section nous procédons aux analyses qualitatives. Nous travaillons avec le corpus segmenté manuellement (section 2.2) et une version non segmentée de ce même corpus. D'un coté, avec les 2 dispositifs, Sagace et dispositif Mecab, nous comptons les occurrences de chaînes dans la version non segmentée du corpus. De l'autre, nous comptons les occurrences de chaînes dans la version segmentée manuellement. Nous comparons ensuite les résultats obtenus.

Pour que les différences entre les lexiques ne biaisent pas les comparaisons, nous ne comparons que les nombres d'occurrences des mots qui sont enregistrés à la fois dans les trois lexiques Ipadic, Jumandic et Unidic et qui en plus apparaissent dans le corpus lemmatisé manuellement. Dans ce sous-lexique commun, nous travaillons sur les noms communs. La comparaison porte au final sur environ 4000 noms communs.

Nous effectuerons deux types de recherches. La première recherche porte sur une chaîne constituée d'un seul nom commun. La seconde recherche est identique, mais nous posons des contraintes sur l'environnement du lemme.

2.3.1 Taux d'accord sans prise en compte de l'environnement

Nous comptons les noms communs, sans poser de contraintes sur l'environnement immédiat droite et gauche.

Sagace	Moyenne des écarts : 16.84 Ecart type : 95.39 Nombre d'entrées avec nombre d'occurrences identique : 3 305 (66.54 %)
Mecab+ Ipadic	Moyenne des écarts : 0.94 Ecart type : 9.71 Nombre d'entrées avec nombre d'occurrences identique : 3 681 (74.11 %)
Mecab + Jumandic	Moyenne des écarts : 0.22 Ecart type : 25.67 Nombre d'entrées avec nombre d'occurrences identique : 3 710 (74.69 %)
Mecab + Unidic	Moyenne des écarts : 0.06 Ecart type:1.21 Nombre d'entrées avec nombre d'occurrences identique : 4 406 (88.71 %)

TABLE 5 : Résultats de Sagace et du dispositif Mecab par rapport au comptage manuel basée sur le lexique Unidic ; comptage des noms communs, sans contraintes sur l'environnement

Comme attendu, les écarts par rapports à l'analyse manuelle sont les plus élevés avec Sagace. En particulier, l'écart type est considérable. Ceci peut s'expliquer par le fait que certaines sous-chaînes du corpus sont interprétées à tord par

Sagace comme des mots du lexique et comptabilisés. Les nombres d'occurrences de ces mots sont alors surestimés. Plus le mot est court et moins il contient de sinogrammes, plus il est détecté à tord.

Cette faiblesse de Sagace quand l'environnement n'est pas pris en compte est connue et inévitable (Blin, 2013). Il est évident que l'outil n'est pas adapté pour une telle recherche.

Pour ce qui concerne le dispositif Mecab, les résultats sont meilleurs avec le dictionnaire Unidic, ce qui s'explique par le fait que c'est ce même dictionnaire Unidic qui a servi de référence à la lemmatisation manuelle.

Les résultats de ce premier test sont globalement conformes aux prévisions et très défavorable à Sagace.

2.3.2 Taux d'accord avec prise en compte de l'environnement

On cherche à nouveau les noms communs (qui appartiennent en même temps aux trois lexiques et au vocabulaire du corpus lemmatisé manuellement), mais cette fois-ci dans un environnement particulier, constitué par des morphèmes et symboles peu susceptibles de provoquer des ambiguïtés :

marque de début de phrase ponctuation particule	nom commun	ponctuation particule copule

Sagace	Moyenne des écarts : 3.12 Ecart type : 21.02 Nombre d'entrées avec nombre d'occurrences identique : 2 007 (40.41 %)
Mecab+ Ipadic	Moyenne des écarts : 2.24 Ecart type :10.13 Nombre d'entrées avec nombre d'occurrences identique : 1 513 (30.46 %)
Mecab + Jumandic	Moyenne des écarts : 0.08 Ecart type : 3.36 Nombre d'entrées avec nombre d'occurrences identique : 1 932 (38.90 %)
Mecab + Unidic	Moyenne des écarts : 1.94 Ecart type : 7.31 Nombre d'entrées avec nombre d'occurrences identique : 1 662 (33.46 %)

TABLE 6 : Résultats de Sagace et du dispositif Mecab par rapport à l'analyse manuelle ; comptage des mots d'une catégorie, avec contraintes sur l'environnement

On constate une très nette amélioration des résultats de Sagace, qui devient tout à fait « compétitif » en termes de qualité des résultats. L'écart type reste élevé mais cela peut s'expliquer par la présence de lemmes ambigus qui peuvent ponctuellement provoquer des écarts sensibles.

Du coté du dispositif Mecab, la surprise vient du fait que l'analyse avec Jumandic est plus proche de l'analyse manuelle que ne l'est celle avec Unidic. Nous n'avons pas d'explication. Une analyse manuelle des résultats serait nécessaire.

3 Conclusion

Dans cette étude, nous avons comparé les performances de deux dispositifs d'analyse de corpus à destination des linguistes du japonais. Ces dispositifs ont été choisis comme représentatifs de deux types de traitement des corpus : une approche « légère » qui s'en tient à la recherche de patrons sans analyse générale des phrases, et une approche complète, plus lourde mais dont on s'attendait à ce qu'elle soit plus fiable, basée sur un traitement préalable complet des énoncés.

Les résultats montrent que, à condition d'éviter certains types de recherches, les performances du premier dispositif sont raisonnables. Si l'on tient en plus compte de l'ergonomie, on peut estimer qu'il s'agit d'un outil tout à fait adapté et suffisant pour le débroussaillage de corpus et l'aide à l'analyse linguistique.

Références

BLIN R. (2012). SAGACE v4.2.0. http://crlao.ehess.fr/japonais-coreen/corpus/sagace/manuel/Manuel.pdf.

BLIN R. (2013). DIctionnaire de Fréquence Du Japonais Contemporain - 16.000 Noms-. Librairie You Feng. Paris.
Kurohashi S., Toshihisa N., Yuji M., Makoto N. (1994) Improvements of Japanese Morphological Analyzer JUMAN. *In* Proceedings of The International Workshop on Sharable Natural Language Resources Pp. 22–28.

Maekawa K. (2009). Daiyousei Wo Yû Suru Daikibo Nihongo Kakikotoba Kôpasu (<tokushyû> Nihongo Kôpasu) [Corpus de japonais écrit, de grande taille et représentatif] En Japonais. Journal of Japanese Society for Artificial Intelligence 24(5): 612–622.

Yamaguchi M. (2011) Zenbun Kensaku Sisutemu “Himawari” Riyousha Manyuaru vers.1.3 [Système D’extraction “Himawari”; Manuel de L’utilisateur vers.1.3]. http://www2.ninjal.ac.jp/lrc/index.php?%C1%B4%CA%B8%B8%A1%BA%F7%A5%B7%A5%B9%A5%C6%A5%E0%A1%D8%A4%D2%A4%DE%A4%EF%A4%EA%A1%D9%2F%CD%F8%CD%D1%BC%D4%A5%DE%A5%CB%A5%E5%A5%A2%A5%EB%2F1_3.