GlossaNet 2: a linguistic search engine for RSS-based corpora

Cédrick Fairon, Kévin Macé, Hubert Naets

Centre de Traitement Automatique du Langage
Université Catholique de Louvain
Louvain-la-Neuve, Belgique
{cedrick.fairon,kevin.mace,hubert.naets}@uclouvain.be

Abstract

This paper presents GlossaNet 2, a free online concordance service that enables users to search into dynamic Web corpora. Two steps are involved in using GlossaNet. First, users define a corpus by selecting RSS feeds in a preselected pool of sources (they can also add their own RSS feeds). These sources will be visited on a regular basis by a crawler in order to generate a dynamic corpus. Second, the user can register one or more search queries on his/her dynamic corpus. Search queries will be re-applied on the corpus every time it is updated and new concordances will be recorded for the user (results can be emailed, published for the user in a privative RSS feed, or they can be viewed online). This service integrates two preexisting software: Corporator (Fairon, 2006), a program that creates corpora by downloading and filtering RSS feeds and Unitex (Paumier, 2003), an open source corpus processor that relies on linguistic resources. After a short introduction, we will briefly present the concept of "RSS corpora" and the assets of this approach to corpus development. We will then give an overview of the GlossaNet architecture and present various cases of use.

1. Introduction

Over the last 10 years, growing needs in the fields of Corpus Linguistics and NLP have led to an increasing demand for text corpora. In this context, the development of Internet was seen as a real opportunity to help meeting the demand (Kilgarriff and Grefenstette, 2003; Hundt et al., 2007). The Web is indeed immense, very diverse and easily accessible…but at the same time it is mixed up, incoherent and most of the time unforeseeable. Although it is technically easy to get access to online documents, it is still a challenge to extract from these documents clean data to compose a corpus. The Cleaneval competition[1] is a good indicator of the state of the art in this domain.

Corpus linguists and NLP specialists have been using the Web as corpus in two directions that complement each other. A first approach considers the Web itself as a very large corpus. Systems that implement this approach usually offer an interface for querying and concordancing the Web. Basically, they add a layer to traditional search engines like Google or Microsoft Live Search and offers various display options that are useful for linguistic work (concordances, extraction of collocates, stop lists, etc.). Among other systems: WebCorp[2] (Renouf, 2003), WebCorpus[3] (Fletcher, 2007), Corpeus[4] (Leturia et al., 2007).

A second approach consists in using the Web as a source for extracting texts that will be collected, filtered and recorded in a standalone corpus. Systems of this second category rely on the use of crawlers and filtering techniques. The Wacky[5] project is a typical example of this option (Baroni and Bernardini, 2006) but other example are numerous (Duclaye et al., 2003; Sekiguchi and Yamamoto, 2004; Fletcher, 2007).

Likewise the first category, the GlossaNet system we are about to present, is an online tool tailored for use by linguists. But the way it sees corpora is closer to the second category. It offers indeed tools for defining "dynamic corpora" that will be downloaded and refreshed on a regular basis by the system.

The original GlossaNet service[6] was in use since 1998 (Fairon, 1999) and limited to press corpora. At the time, newspapers Web sites were seen as a good source for generating "dynamic corpora", because they are updated on a daily basis[7] (new texts come continuously day after day).

GlossaNet users can create an account, select a series of newspapers (which define a virtual corpus) and register search queries. The system integrates the programs and linguistic resources of Unitex[8], an open source corpus processor (Paumier, 2003). Unitex is used for applying several analysis tasks on the corpus: tokenization, sentence segmentation, dictionary lookup. Once the corpus is processed, it becomes possible to search for linguistic patterns. To make it short, one can describe GlossaNet as a linguistic search engine of limited scope. Until recently, this scope was very limited as it covered only a hundred newspapers Web sites. GlossaNet 2 goes beyond this limitation as it can download corpora from any RSS/ATOM feed (see section 2.). GlossaNet integrates two preexisting software: Corporator (Fairon, 2006), a program that creates corpora by downloading and filtering RSS feeds and Unitex. Both software are available as standalone applications.

Two steps are involved in using GlossaNet. First, users define a corpus by selecting RSS feeds in a preselected pool of sources (they can also add their own RSS feeds). These sources will be visited on a regular basis by a crawler in

[1] http://cleaneval.sigwac.org.uk/
[2] http://www.webcorp.org.uk/
[3] http://webascorpus.org/searchwac.html
[4] http://www.corpeus.org/
[5] http://wacky.sslmit.unibo.it/
[6] http://glossa.fltr.ucl.ac.be/
[7] Also, newspapers are available in many different languages,they cover many different themes, they represent various styles, various types of text (argumentative, informative, technical, pedagogical, etc.) and they provide texts of a constant quality.
[8] http://www-igm.univ-mlv.fr/~unitex/

```
<rss version="2.0">

<channel>
    <title>Paris Libre</title>
    <lastBuildDate>
    Tue, 4 Mar 2008 13:12:31 +0100
    </lastBuildDate>
    <item>
        <title>Un sport</title>
        <link>
        http://parislibre.lalibreblogs.be/
        archive/2008/03/04/un-sport.html
        </link>
        <pubDate>
        Tue, 4 Mar 2008 10:45:00 +0100
        </pubDate>
        <description>
        C'est un peu comme courir le marathon.
        Quand on est journaliste de presse...
        </description>
</item>
</channel>
</rss>
```

Figure 1: Example of RSS feed

```
<feed xmlns="http://www.w3.org/2005/Atom"
xml:lang="fr">

<title>Paris Libre</title>
<updated>2008-03-04T13:08:38+01:00
</updated>
<entry>
        <title>Un sport</title>
        <link rel="alternate" type="text/html"
        href="http://parislibre.lalibreblogs.be/
        archive/2008/03/04/un-sport.html" />
        <updated>
        2008-03-04T10:57:03+01:00
        </updated>
        <published>
        2008-03-04T10:45:00+01:00
        </published>
        <summary>
        C'est un peu comme courir le marathon.
        Quand on est journaliste de presse...
        </summary>
        </entry>
</feed>
```

Figure 2: Example of Atom feed

order to generate a dynamic corpus. Second, the user can register one or more search queries on his/her dynamic corpus. Search queries will be re-applied on the corpus every time it is updated and new concordances will be recorded for the user (results can be emailed, published for the user in a private RSS feed, or they can be viewed online).

2. RSS and Atom feeds

2.1. What are RSS and Atom ?

RSS is the acronym for Really Simple Syndication. XML-based format, RSS is used to facilitate news publication on the Web and content interchange between websites. Atom is another standard built with the same objective.

Actually, most of the newspapers and blogging websites offer RSS and / or Atom-based news feeds to allow easy access to the recently published news articles. Each RSS / Atom file containts a list of articles recently published, often grouped by theme or category. Usually, these files do not containt full articles, but the title, the date of publication, a link to the full article available on the publisher website and a summary or a truncated text. On a regular basis (every day, every hour or even more frequently), the RSS / Atom feeds are updated with the new published content. The feeds are often organized by theme and / or in accordance with sections of the newspaper or the blog ("politics", "social", "nature", "editorial","regions",... for newspapers, or "my cats", "my friends" and "computational linguistics",... for blogs). Feeds can be also created for special hot topics like "Partition of Belgium" in the French-speaking press for example.

Figures 1 and 2 show respectively the basic structure of an RSS and an Atom feed. These XML-based structures are very simple. For RSS, it mainly consists in a "channel" wich contains a list of "items" such as a title, a link, a description and a publication date.

An Atom feed is composed of more or less the same information, i.e. a "feed" wich contains a list of "entries" comprising, among other things, a title, a link, a summary and information about the last update of the entry.

2.2. RSS and corpora

Most of the time, RSS and Atom feeds contain few text in the "description" or "summary" field (even though some publishers incorporate the full article in the RSS / Atom feeds), but each "item" or "entry" has a link to the full article. Therefore, the link can be used to download the corresponding Web page (see section3.5. and figure 3).

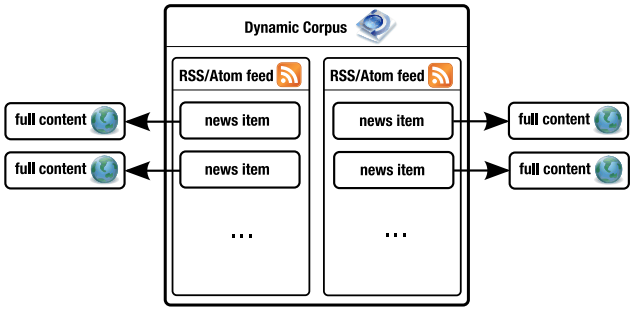


Figure 3: Dynamic Corpus using RSS / Atom feeds

As mentioned above, RSS and Atom feeds are frequently classified by genre, theme or category. Unfortunately, this classification is not standardized for newspapers, let-alone for blogs or other websites using feeds. In addition, no other indication is given about the classification criteria. However, if the classification fits the researchers needs, the RSS feeds can be used to build a specialized corpus.

A second characteristic of the RSS / Atom feeds is that they are frequently updated, which provides a continuous flow of data that can be easily collected in a corpus.

A third characteristic is that, by selecting "trusted" RSS or Atom sources (for example newspaper sources), the quality of the retrieval texts will be constant, which resolves the well known problem of Web corpus quality: when the Web is crawled for finding source, the quality between documents may differ tragically.

For all these reasons and despite the problem of classification criteria, using RSS or Atom feeds represents probably one of the most interesting way to build a corpus from the Web.

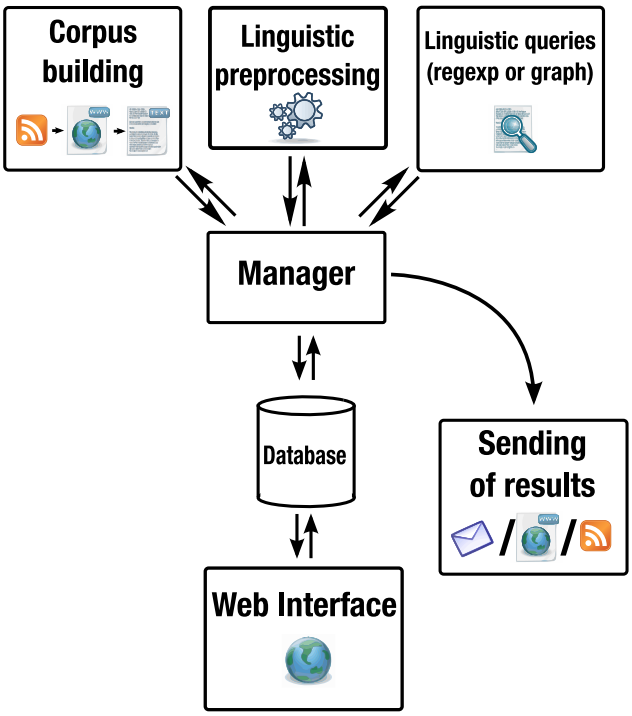


Figure 4: GlossaNet architecture

3. Architecture and process

3.1. Architecture

GlossaNet is made of three parts (figure 4) :

1. a Web interface to interact with users;
2. a database which contains data about each user, his / her corpora and his / her linguistic queries;
3. a server-side part composed of five servers in an asynchronous and distributed framework.

This architecture allows to use this five servers on various computers and to split a server in two or more instances. This characteristic was necessary to prevent server overloads due to the ability given to the users to create many corpora from virtually an infinity of RSS and Atom feeds. We use the Perl Object Environment[9] (POE) framework to provide the asynchronous and distributed layer.

3.2. Process

The basic mechanism of GlossaNet is the next one (figure 4). A Web interface (section 3.3.) allows the user to build one or more dynamic corpora from one or more RSS / Atom feeds selected from the Web by himself. In addition, the user can create or use linguistic queries (made of regular expressions or graphs) and apply these queries to one or more previously build corpora. The different types of information (about user, corpora, linguistic queries and application of queries to corpora) are recorded into a database.
This database is regularly scanned by a server, the "Manager" (section 3.4.) which detect from a new corpus is added. The "Manager" adds the RSS / Atom feeds from the corpus to the list of the feeds to download. On a regular basis, these are sent to a second server, ("corpus building", section 3.5.) which downloads each feed, extracts the new entries ("items" in the RSS jargon) and, for each entry, gets and cleans the corresponding Web page. These cleaned texts are transmitted to the "Manager" which stocks them into the database.

[9] `http://poe.perl.org/`

Figure 5: Login screen

Once a day, the "Manager" collects all the new texts of a corpus, concatenates and sends them to the "Linguistic preprocessing" server (section 3.6.). This server uses the Unitex software to tokenize, normalize and tag the text if the corresponding ressources are available for the corpus language. Then a link to the preprocessed corpus is sent back to the "Manager".
The "Manager" then checks the database to determine which linguistic queries may be applied to this preprocessed corpus. The links to the corpus and each query (in the form of a finite state transducer) are posted to the next server which role is to retrieve all the concordances between the corpus and the query (section 3.7.). The results are sent back to the "Manager".
When all the queries were applied to all the corpora for a given user, the results of the day are transmitted to him (section 3.8.). The sending frequency depends on the user's preferences.

3.3. Web interface

The Web interface is divided into various pages which offer to create and manage his / her user's profile (figure 5), his / her own copora (figure 7), his / her linguistic queries (figure 6) and to apply these queries to the corpora (figure 8).

3.4. Manager

The manager role consists in supervising all the process and sending at the suitable time the data needed by each server. It makes sure the results or useful information (from example about the unavailability of some RSS / Atom feeds) are sent to the user. These tasks are very important and complex within an asynchronous and distributed framework.

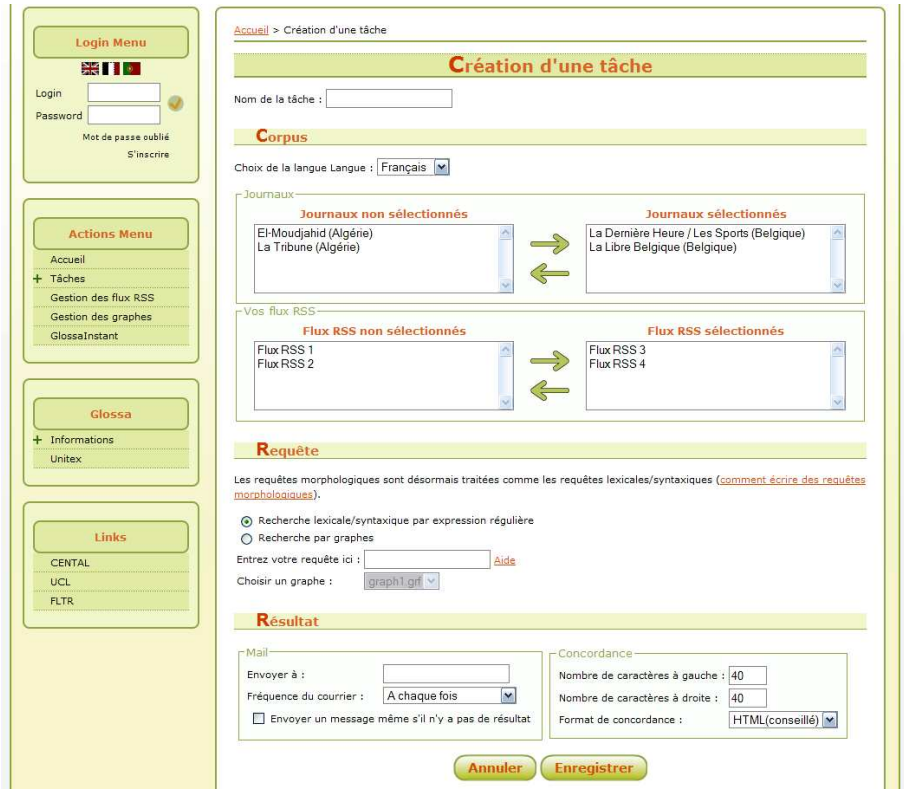


Figure 8: Task creation

Figure 6: Graphs manager

Figure 7: Feeds manager

3.5. Corpus building

The "corpus building" server frequently receives RSS or Atom feeds to check from the "Manager". Each time a RSS feed is updated, the server selects the new entries and extracts from them a title, a description and an URL. Then, it downloads each Web page corresponding to the URL, removes the boilerplate and send the page to the "Manager" in text format.

During this processing, two specific problems occur : the boilerplate suppression and the duplicated news.

3.5.1. Boilerplate removal

The main difficulty for building the corpus is to remove irrelevant text and links from the Web page. This automatic boilerplate suppression is necessary to have cleaned texts which can be used with Unitex.
There are three differences with the Cleaneval task:

- First, GlossaNet can use the title and the description of each RSS item (and the title and the summary of each Atom entry);
- Next, a lot of newspaper websites split long articles in

two or more pages (where generally only one page or an entire website are considered);

- Last, some websites using RSS / Atom have common features, like the “print” button that open a new webpage containing a formatted and more or less cleaned text.

Considering all these specificities, we are developping two kinds of filters:

- specific filters for frequently used newspapers websites or for common blogging systems like Blogger, Wordpress, etc.;
- and general filters (“by default”) using RSS titles and descriptions, or frequent features like the “print” button.

These filters allow for an important coverage of the Web and the best quality for some sources frequently used by the GlossaNet users.

3.5.2. Duplicated news

Another difficulty comes from the updating of the pages. Regularly, Web pages (and consequently RSS items) are modified owing to updates and spellchecking.
In this case, a new RSS item (or Atom entry) is added into the RSS / Atom file to inform the user about a modification. The problem is that almost all the content of the original news is similar to the updated content, which have for consequence that the more or less same content appears two or more times into the corpus, causing , among other things, important effects of the words frequency. A linguistic query will also match with more text than normal. =¿ causing the content to appear twice or more in the corpus. This has important consequences on word frequencies; linguistic queries may match more text than they should.
We are working on this problem, using similarity measures to identify and eliminate the duplicated news.

3.5.3. Corporator

The “corpus buiding” server corresponds to the new core of Corporator. This new version of the software will be a standalone version with a GUI and will include the improvements made for GlossaNet about better boilerplate filters and similarities detection. Corporator will be difused under open source licence.

3.6. Linguistic preprocessing

The preprocessing step is mainly dependent on the Unitex software. If Unitex have linguistic resources for the language of the corpus (witch is the case for English, Finnish, French, German, Ancient Greek, Modern Greek, Italian, Korean, Norwegian, Polish, Portuguese from Brazil, Portuguese from Portugal, Russian, Serbian (Cyrillic and Latin), Spanish and Thai), the text will be tokenized, normalized, lemmatized and tagged with morphosyntactic (and semantic) tags. Otherwise, the text is simply preprocessed to be used with the linguistic queries.

3.7. Linguistic queries

If the corpus language is supported by Unitex, i.e. if the corpus is tokenized, normalized, lemmatized and tagged, very complex queries may be created, using form, lemma, parts of speech, morphological and semantic information. Figure 9 shows a complex query using

- simple form (“to”);
- lemma (“be” between chevrons, i.e. all the inflectional forms of “be”);
- part of speech and morphological information (“<V:G>”, i.e. any verb with an -ing form, and “<V:W>, i.e. any verb with an infinitive form);
- part of speech and semantic information (“<N+conc>”, i.e. any concrete noun).

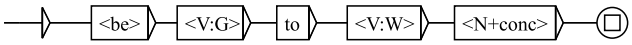


Figure 9: Example of a complex linguistic query

This query matches for example with the sentence “I am going to eat bread” but also with “He is looking to get books”. Otherwise, if the language is not supported, more simple queries using linguistic forms can be created (figure 10).

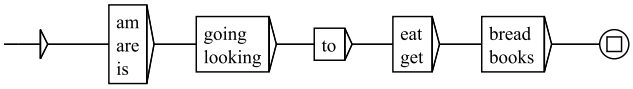


Figure 10: Example of a simple linguistic query

The “linguistic queries” server sends finaly a concordances set to the “Manager”. The characteristics of the concordances can be determined by each user in the Web interface.

3.8. Sending of results

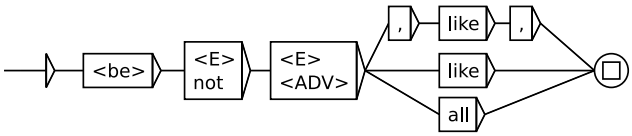
The user can also define the sending frequency and method of the results : when a result is available, or via a daily or a weekly digest; by e-mail, by RSS feed, or via the GlossaNet website interface.

4. Use cases

The use cases presented here are just a few examples of enquiries that can be achieved using GlossaNet.

4.1. Linguistic

Fairon and Singler (2006) used GlossaNet in their study of a particular type of quotative that occurs frequently in American Vernacular English and might be becoming part of the Standard English: (be) like. To evaluate how this quotative is spreading in written English, GlossaNet was used to monitor newspapers from various parts of the World. The Finite State Graphs search option facilitated the extraction of variants of this quotative.



4.2. Web sites monitoring

If GlossaNet was firstly conceived for linguists, it also proved to be useful for another category of users. We have indeed noticed that many users are more interested by "information" than "linguistic patterns". Their queries contain keywords intended to collect news clippings. Once registered into the system, queries are reapplied every time corpora are updated. It is therefore possible to monitor Web sites or information sources. Finite state graphs offer a convenient way for representing many variants of keywords.

5. Conclusion

In this paper, we have presented the second main version of GlossaNet. This online application allows now to build dynamic corpora from the Web using RSS and Atom feeds with the goal to extract data from these corpora with help of linguistic patterns.
Four points should be highlighted:

1. Completely rebuilt, with a much better design, the new Web interface is the essential entry point of GlossaNet. As such, it had to be completely redesigned, especially as the features provided for the users have been expanded to manage the Atom and RSS feeds.

2. If the corpus dynamic proposed in the previous version of GlossaNet were a hundred, their number is now virtually infinite. With the use of an increasing number of RSS and Atom feeds, an important part of the Web has become the corpus of GlossaNet.

3. The number of accessible corpora has changed; their building criteria have also changed. The new version of GlossaNet allows the creation of corpora as well as from websites and blogs talking about desserts, than from the pages "political" of such newspaper, that from newsgroups about Asian elephants.

4. The server-side architecture has been completely redesigned. Very flexible, it can easily integrate new features and interact with other software. We are already working on an application connecting GlossaNet to other software.

6. References

Mario Baroni and Silvia Bernardini, editors. 2006. *Wacky! Working papers on the Web as Corpus*. GEDIT, Bologna.

F. Duclaye, F. Yvon, and O. Collin. 2003. Unsupervised incremental acquisition of a thematic corpus from the web. In *Proceedings of Natural Language Processing and Knowledge Engineering*. IEEE.

Cédrick Fairon and John V. Singler. 2006. I'am like, 'Hey, it works': Using GlossaNet to find attestations of the quotative (be) like in English-language newspapers. In A. Renouf and A. Kehoe, editors, *The Changing Face of Corpus Linguistics*, number 55, pages 325–337. Language and computers: Studies in Practical Linguistics, Amsterdam - New York.

Cédrick Fairon. 1999. Parsing a web site as a corpus. In Cédrick Fairon, editor, *Analyse lexicale et syntaxique: Le système INTEX, Lingvisticae Investigationes*, volume XXII, pages 327–340. John Benjamins Publishing, Amsterdam/Philadelphia.

Cédrick Fairon. 2006. Corporator: A tool for creating rss-based specialized corpora. In *Proceedings of the Workshop Web as corpus*, Trento. EACL.

William H. Fletcher. 2007. Implementing a BNC-Compare-able Web Corpus. In C. Fairon, H. Naets, A. Kilgariff, and G.-M. de Schryver, editors, *Building and Exploring Web Corpora*, volume 4, Louvain-la-Neuve. Cahiers du Cental.

Marianne Hundt, Nadja Nesselhauf, and Carolin Biewer, editors. 2007. *Corpus Linguistics and the Web, Language and computers studies in practical linguistics*, volume 59. Rodopi, Amsterdam - New York.

Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3):333–348.

Igor Leturia, Antton Gurrutxaga, Iñaki Alegria, and Aitzol Ezeiza. 2007. CorpEus, a 'web as corpus' tool designed for the agglutinative nature of basque. In C. Fairon, A. Kilgariff, H. Naets, and G.-M. de Schryver, editors, *Building and Exploring Web Corpora*, number 4, Louvain-la-Neuve. Cahiers du Cental.

Sébastien Paumier. 2003. *De la reconnaissance de formes linguistiques à l'analyse syntaxique*. Ph.D. thesis, Université de Marne-la-Vallée.

Antoinette Renouf. 1993. A word in time: first findings from the investigation of dynamic text. In J. Aarts, P. de Haan, and N. Oostdijk, editors, *English Language Corpora: Design, Analysis and Exploitation*, pages 279–288, Amsterdam. Rodopi.

Antoinette Renouf. 2003. Webcorp: providing a renewable energy source for corpus linguistics. In S. Granger and S. Petch-Tyson, editors, *Extending the scope of corpus-based research: new applications, new challenges*, pages 39–58, Amsterdam. Rodopi.

Youichi Sekiguchi and Kazuhide Yamamoto. 2004. Improving quality of the web corpus. In *Proceedings of The First International Joint Conference on Natural Language Processing (IJCNLP-04)*, pages 201–206.