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

Cédrick Fairon, Kévin Macé, Hubert Naets

Centre de Traitement Automatique du Langage — Cental
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. At 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. Secondly, 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 the 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 clean data from these documents 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 seeking concordances within the Web. Basically, they add a layer to traditional search engines like Google or Microsoft Live Search and offer 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 examples are numerous (Duclaye et al., 2003; Fletcher, 2007).

Similarly to 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. Indeed, it offers tools for defining "dynamic corpora" that will be downloaded and refreshed on a regular basis by the system.

The original GlossaNet service[6] has been in use since 1998 (Fairon, 1999) and was limited to press corpora. At the time, newspapers'websites 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 websites. GlossaNet 2 goes beyond this limitation as it can download corpora from any RSS / Atom feed (see section 2.). GlossaNet integrates two preexisting pieces of 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. At 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] In addition, 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. Secondly, 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. As 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 contains a list of articles recently published, often grouped by theme or category. Usually, these files do not contain 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", etc. for newspapers, or "my cats", "my friends" and "computational linguistics", etc. 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 respectively show the basic structure of an RSS and an Atom feed. These XML-based structures are very simple. For RSS, it mainly consists of a "channel" which 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" which contains a list of "entries" comprising, among other things, a title, a link, a summary and information about the time of the last update.

2.2. RSS and corpora

Most of the time, RSS and Atom feeds contain little 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 webpage (see section 3.5. and figure 3).

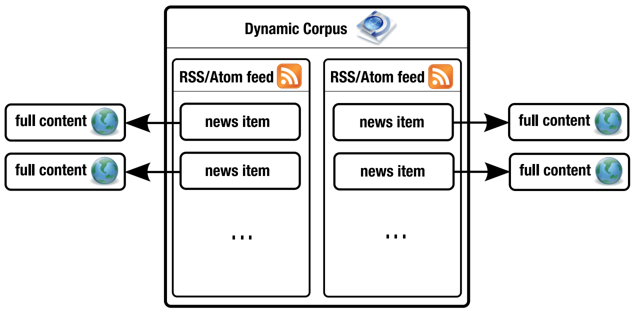


Figure 3: Dynamic Corpus using RSS / Atom feeds

As mentioned above, RSS and Atom feeds are frequently classified by genre, theme or category. Unfortunately, this classification is not standardized among newspapers, let alone 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 sources, the quality between documents may differ tragically.

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

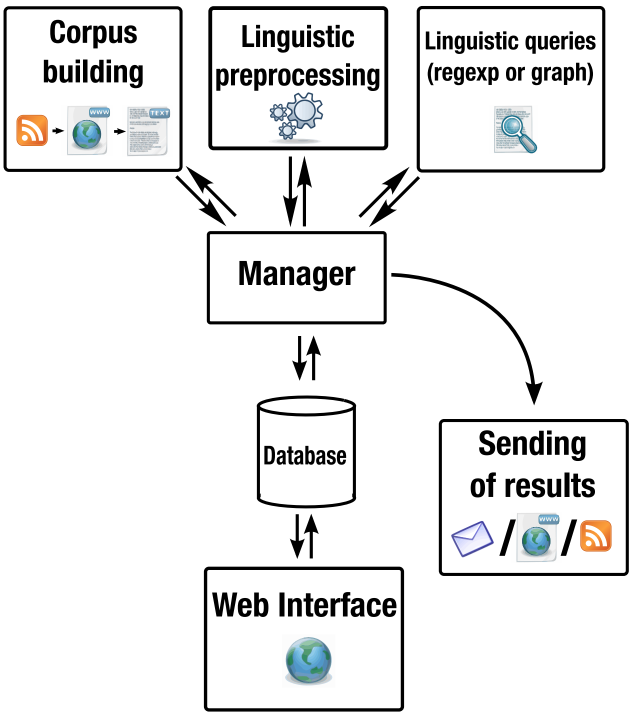


Figure 4: GlossaNet architecture

Web. But it also comes with some limitations since all of the Web content is not available through these feeds.

3. Architecture and process

3.1. Architecture

GlossaNet is made of three parts (figure 4):

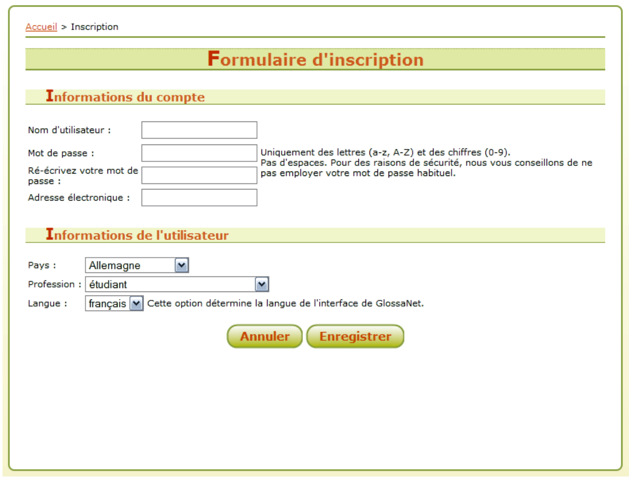
1. a Web interface to interact with users;
2. a database which contains data about each user, his / her corpora and his / her linguistic queries;
3. a server-side part composed of five servers organized in an asynchronous and distributed framework.

This architecture enables the use of these 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 a virtually infinite number 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 following (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 (any feed available online can be selected). 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 built corpora. Information about user, corpora and linguistic queries are recorded into a database.

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


Accueil > Inscription

Formulaire d'inscription

Informations du compte

Nom d'utilisateur :		
Mot de passe :		Uniquement des lettres (a-z, A-Z) et des chiffres (0-9). Pas d'espaces. Pour des raisons de sécurité, nous vous conseillons de ne pas employer votre mot de passe habituel.
Ré-écrivez votre mot de passe :		
Adresse électronique :		

Informations de l'utilisateur

Pays :	Allemagne	
Profession :	étudiant	
Langue :	français	Cette option détermine la langue de l'interface de GlossaNet.

Figure 5: Login screen

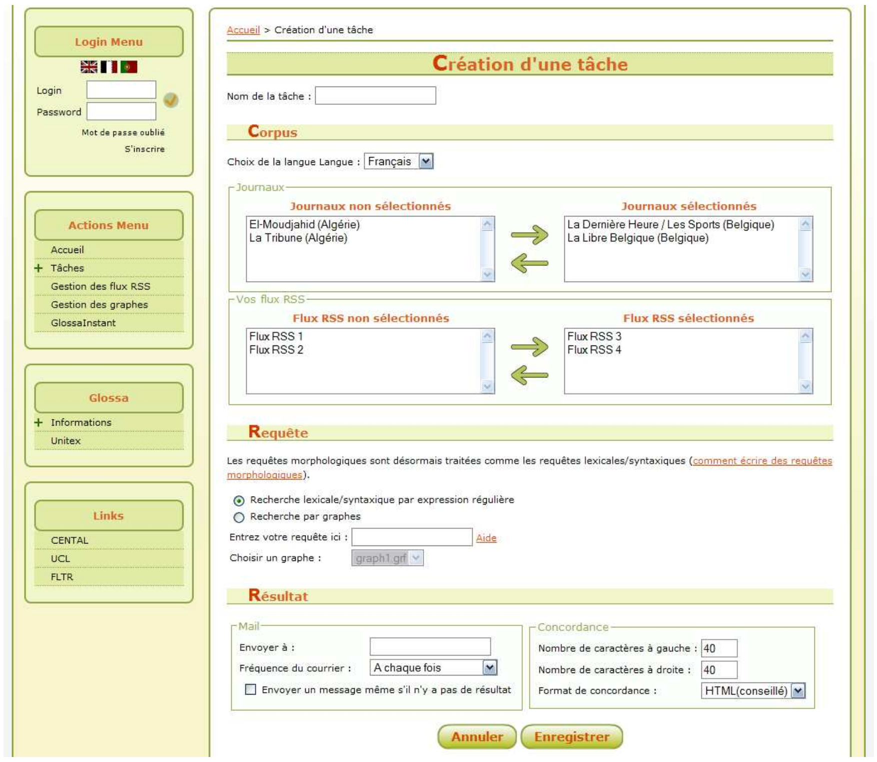
A central server named "Manager" (section 3.4.) detects when a new corpus is added into the database. When it happens, the "Manager" appends the RSS / Atom feeds of this corpus to the list of the feeds to download. On a regular basis, the list is sent to the "corpus building" server (section 3.5.) which downloads each feed, extracts the new entries ("items" in the RSS jargon) and, for each entry, gets and cleans the corresponding webpage. Clean texts are transmitted back to the "Manager" which stores them into the database.
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 and normalize the text. If linguistic resources are available for the corpus language, Unitex is also used to apply dictionaries and tag the corpus. Then a path to the preprocessed corpus is sent back to the "Manager".
When corpora are ready, the "Manager" checks the database to determine which linguistic queries have to be applied. The path to the corpus and each query (in the form of a finite state transducer) are posted to the next server which will retrieve all the concordances between the corpus and the query (section 3.7.). Results are sent back to the "Manager".
Once all the queries have been applied for a given user, results are transmitted to him (section 3.8.). Depending on the users preferences, results are sent on a daily or weekly basis.

3.3. Web interface

The Web interface provides several control pannels for creating and managing user's profile (figure 5), copora (figure 8) and linguistic queries (figure 7 and 6).

3.4. Manager

The manager role consists in supervising all the processes and sending the data needed by each server at a suitable time. It makes sure the results or useful information (for example about the unavailability of some RSS / Atom feeds) is sent to the user.



Accueil > Création d'une tâche

Création d'une tâche

Nom de la tâche :

Corpus

Choix de la langue Langue : Français

Journaux

Journaux non sélectionnés	Journaux sélectionnés
El-Moudjahid (Algérie)	La Dernière Heure / Les Sports (Belgique)
La Tribune (Algérie)	La Libre Belgique (Belgique)

Vos flux RSS

Flux RSS non sélectionnés	Flux RSS sélectionnés
Flux RSS 1	Flux RSS 3
Flux RSS 2	Flux RSS 4

Requête

Les requêtes morphologiques sont désormais traitées comme les requêtes lexicales/syntaxiques (comment écrire des requêtes morphologiques).

[✓ Recherche lexicale/syntaxique par expression régulière]
Recherche par graphes

Entrez votre requête ici : Aide

Choisir un graphe : graph1.grf

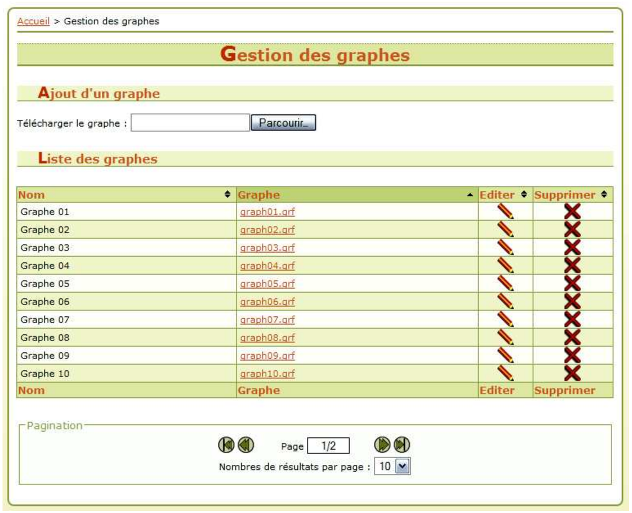
Résultat

Mail
Envoyer à :
Fréquence du courrier : A chaque fois
Envoyer un message même s'il n'y a pas de résultat

Concordance
Nombre de caractères à gauche : 40
Nombre de caractères à droite : 40
Format de concordance : HTML(conseillé)

Annuler Enregistrer

Figure 6: Task creation



Accueil > Gestion des graphes

Gestion des graphes

Ajout d'un graphe

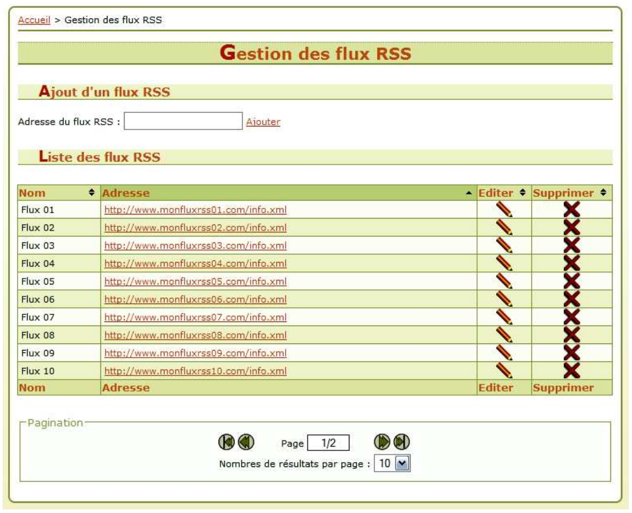
Télécharger le graphe : Parcourir...

Liste des graphes

Nom	Graphe	Editer	Supprimer
Graphe 01	graph01.grf	✎	✗
Graphe 02	graph02.grf	✎	✗
Graphe 03	graph03.grf	✎	✗
Graphe 04	graph04.grf	✎	✗
Graphe 05	graph05.grf	✎	✗
Graphe 06	graph06.grf	✎	✗
Graphe 07	graph07.grf	✎	✗
Graphe 08	graph08.grf	✎	✗
Graphe 09	graph09.grf	✎	✗
Graphe 10	graph10.grf	✎	✗
Nom	Graphe	Editer	Supprimer

Pagination
Page 1/2
Nombres de résultats par page : 10

Figure 7: Graphs manager



Accueil > Gestion des flux RSS

Gestion des flux RSS

Ajout d'un flux RSS

Adresse du flux RSS : Ajouter

Liste des flux RSS

Nom	Adresse	Editer	Supprimer
Flux 01	http://www.monfluxrss01.com/info.xml	✎	✗
Flux 02	http://www.monfluxrss02.com/info.xml	✎	✗
Flux 03	http://www.monfluxrss03.com/info.xml	✎	✗
Flux 04	http://www.monfluxrss04.com/info.xml	✎	✗
Flux 05	http://www.monfluxrss05.com/info.xml	✎	✗
Flux 06	http://www.monfluxrss06.com/info.xml	✎	✗
Flux 07	http://www.monfluxrss07.com/info.xml	✎	✗
Flux 08	http://www.monfluxrss08.com/info.xml	✎	✗
Flux 09	http://www.monfluxrss09.com/info.xml	✎	✗
Flux 10	http://www.monfluxrss10.com/info.xml	✎	✗
Nom	Adresse	Editer	Supprimer

Pagination
Page 1/2
Nombres de résultats par page : 10

Figure 8: Feeds manager

3.5. Corpus building

The "corpus building" server frequently receives new RSS / Atom feeds from the "Manager" to check. When the program detects that a RSS feed was updated, it retrieves the new items from the RSS file and, for each item, stores its title, description and URL. Then, it downloads each webpage corresponding to the URL, removes the boilerplate and sends the page to the "Manager" in text format.

During this process, two tasks must be considered: the boilerplate filtering and the identification of duplicated news.

3.5.1. Boilerplate removal

The main difficulty for building the corpus is to remove irrelevant text and links from webpages. This automatic boilerplate suppression is necessary to obtain clean texts which can be used with Unitex.

There are three differences with the Cleaneval task:

- At first, GlossaNet filters can rely on the metadata provided by the RSS feed (title and description). It is for instance helpful to locate the title in the HTML document because it indicates where the (relevant) text begins;

- Next, a lot of newspaper websites split long articles over two or more pages, but the RSS item refers only to the first page;
- Finally, some websites using RSS / Atom have common features, like the “print” button that opens a new webpage displaying the entire text in a simplified layout.

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

- specific filters for frequently used newspaper websites or for common blogging systems like Blogger, Wordpress, etc.;
- generic filters using RSS titles and descriptions, or frequent features like the “print” button.

As it is the case in most web corpus projects, filtering remains a challenge.

- Although it is possible to create specific filters for the most popular sources (like the main national newspapers or blog publishing systems), generic filters remain necessary for all other type of webpages.
- Since website layouts change over the time, specific filters must be continuously adapted.

3.5.2. Duplicated news

The system also needs a filter that prevents a single text from appearing more than once in the corpus. Several situations may lead to text duplicates:

- a text that was published earlier on a RSS feed can be broadcasted again for various reasons: that can happen simply by mistake or on purpose, for instance after small corrections (typos, etc.) or after the addition of a new paragraph to the text;
- on newspaper websites, it is common to find, next to a news article, links to other related articles. These links can consist in short titles or long paragraphs borrowed from the referred article;
- on blogs, several posts can be displayed on the same webpage. Each time a new post is added, the RSS feed is updated and when the Corporator crawler follows the link to get the full article, it retrieves not only the last article but also older articles presented on the same webpage.

As we can see, these difficulties go beyond the usual ‘boilerplate removal’. Of course, if we do not address this problem, it could have important consequences on word frequencies in the corpus. Moreover, concordances built from this corpus will contain duplicates.

3.5.3. Corporator

The “corpus building” server corresponds to the new core of Corporator. This new version of the software will be released as a standalone version with a GUI and will include the improvements made for GlossaNet (boilerplate filters and similarities detection). Corporator will be distributed under an open source licence.

3.6. Linguistic preprocessing

The preprocessing step is mainly dependent on the Unitex software. The corpus is first normalized and tokenized and then, if linguistic resources are available for the corpus language, a dictionary lookup program is used to tag the text (Unitex provides linguistic resources for English, Finnish, French, German, Ancient Greek, Modern Greek, Italian, Korean, Norwegian, Polish, Portuguese from Brazil, Portuguese from Portugal, Russian, Serbian, Spanish and Thai).

3.7. Linguistic queries

If the corpus language is supported by Unitex, complex queries may be created, using word form, lemma, part of speech, morphological and semantic information.
Figure 9 shows a complex query using:

- a simple form (“to”);
- a lemma (“be” between chevrons, i.e. all the inflectional forms of “be”);
- parts 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);
- parts of speech and semantic information (“<N+conc>”, i.e. any concrete noun).

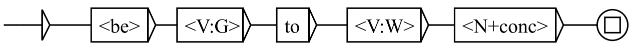


Figure 9: Example of a complex linguistic query

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

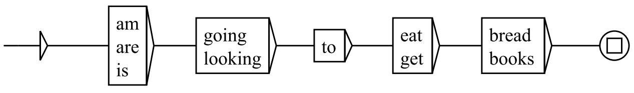


Figure 10: Example of a simple linguistic query

The “linguistic queries” server build a concordance and sends it to the “Manager”. The characteristics of the concordance (sort order, length, number of occurrences, etc.) can be determined by the user in the Web interface.

3.8. Sending of results

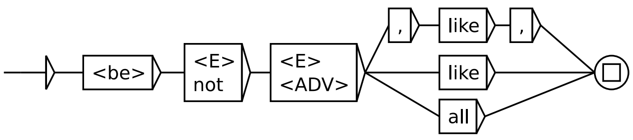
The user can also define how and how often results will be provided to him: on a daily or weekly basis; by e-mail, by RSS feed, or via the GlossaNet website interface.

4. Use cases

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

4.1. Linguistics

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. In order 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. Websites monitoring

If GlossaNet was initially 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 websites or information sources. Finite state graphs offer a convenient way for representing many variants of keywords.

5. Conclusion

In this paper, we have described a new version of GlossaNet which represents a major update of the system. Now, this online application allows the user to build dynamic corpora from the Web using RSS and Atom feeds. Then, these dynamic corpora can be searched for linguistic patterns and results are presented under the form of a concordance that can be transmitted to the user by email, on a privative RSS feed or that can be read online in the user account. Four points should be highlighted:

1. The Web interface, which is the main entry point of GlossaNet, was entirely redesigned. This was necessary to improve ergonomics but also for integrating the new features related to the management of the RSS / Atom feeds.

2. The previous version of GlossaNet was limited to a series of approximately one hundred dynamic corpora (which were generated by a crawler bound to a preselected pool of newspaper websites). In this new version, the number of possible corpora is virtually infinite. Each user can design his / her own corpus by combining RSS / Atom feeds that are every day more numerous on the Web.

3. If GlossaNet corpora were formerly limited to newspapers, it is no longer the case: any site that offers content through RSS / Atom feeds can be used. It is easy to build thematic corpora (finance, recipes, sport, health, etc.) or corpora representing certain genres (blog, forum, etc.)

4. The system was completely rebuilt on improved software architecture. Thanks to the modularity of the new architecture, it is now easy to integrate new features or to bridge the system with other pieces of software. For instance, we use the system for generating flows of textual data that feed other applications (for corpus lexicography, information extraction, etc.)

6. References

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. 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.

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

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.

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.