Looking for French deverbal nouns in an evolving Web (a short history of WAC)

Nabil Hathout, Franck Sajous, Ludovic Tanguy

Looking for French deverbal nouns in an evolving Web (a short history of WAC)

Nabil Hathout **Franck Sajous** **Ludovic Tanguy**
CLLE / CNRS and University of Toulouse, France
{hathout,sajous,tanguy}@univ-tlse2.fr

Abstract

This papers describes an 8-year-long research effort for automatically collecting new French deverbal nouns on the Web. The goal has remained the same: building an extensive and cumulative list of noun-verb pairs where the noun denotes the action expressed by the verb (e.g. *production - produce*). This list is used for both linguistic research and for NLP applications. The initial method consisted in taking advantage of the former Altavista search engine, allowing for a direct access to unknown word forms. The second technique led us to develop a specific crawler, which raised a number of technical difficulties. In the third experiment, we use a collection of web pages made available to us by a commercial search engine. Through all these stages, the general method has remained the same, and the results are similar and cumulative, although the technical environment has greatly evolved.

1 Introduction

The Web has been successfully used as a corpus for more than 10 years now, and as everything web-related, things have been evolving at tremendous speed. From the pioneer hackings of early search-engines in the late 20th century to the current development of linguistically-aware web corpus builders, many different efforts have been made to tap into this bottomless pit of linguistic data. What we present here is the technical evolutions of a narrow-focused research effort we have been working on for about 8 years. Our goal is the automatic acquisition of new French words, to be used as descriptive materials for morphology, and to a certain extent as a resource for natural language processing. More precisely, we search for new suffixed word forms, based on a set of productive French suffixes: mainly *-age*, *-ion* and *-ment*, which are used to coin nouns from verbs. Section 2 describes more precisely our objectives.

Although this task is quite simple with regards to current techniques in traditional corpus linguistics, complications arise when it is applied to the Web, as noted by Lüdeling et al. (2007). The main problem is that we are looking for word forms we know to be quite rare, and for which we only know the ending substring. If the Web is a very good answer to the former characteristic (because of its size and constant evolution), it is not adapted to the latter. This led us to use three different techniques for getting to our end. Each change from one technique to the other can be explained by the evolution of Web access. Section 3 describes the main method we used. In section 4, we try to draw a short history of the main evolutionary steps in using the Web as a corpus. Finally, section 5 describes more technically the three different solutions we applied along the last 8 years and the corresponding results.

2 The quest for French derived words

2.1 Data for NLP and extensive morphology

There is a large number of inflexional lexica available for many languages but very few derivational ones. For instance, we only know of two morphological databases for English: CELEX (Baayen et al., 1995) and Catvar (Habash and Dorr, 2003). CELEX also includes databases for German and Dutch. For French, hardly any such database exists. One exception is Verbaction[1] which describes the deverbal nouns of a large set of French verbs.

Derivational databases have initially been set up and used by psycho-linguists working on the mental lexicon and on the processing of derived words. They have also been used in NLP applications and Information Retrieval experiments. For instance, the French parser Syntex (Bourigault and Fabre, 2000) uses Verbaction for the disambiguation of PP attachments and Jing and Tzoukerman (1999) propose a method of query expansion with morphologically related words from CELEX. Derivational resources are also used in linguistics as corpora for the description of morphological pro-

[1] w3.erss.univ-tlse2.fr/verbaction/

cesses. These resources must be very large in order to allow for the observation and study of rare phenomena. This approach is known as "extensive morphology." Morpho-phonological studies such as (Plénat, 2000) or morpho-semantic ones such as (Hathout et al., 2003) have shown the fruitfulness of this approach and how the use of great quantity of data leads to new insights on the morphological phenomena (see (Hathout et al., 2008) for a detailed presentation of extensive morphology).

In order to study a given morphological phenomenon, say the effect of the length of a stem on the truncation of its final rhyme (for instance, why is the stem truncated in *inoxydation* 'process that makes steel become stainless' which should be *inoxydabilisation* and not in *dénationalisation*), one needs lots of examples for a large number of configurations. The existing databases are rather small and do not contain enough examples to carry out these studies. The only place where the needed amounts of examples could be found and collected from is the Web.

Once the data has been gathered, the linguist is faced with an even harder problem: manually checking all of them in order to remove the erroneous ones such as words in foreign languages, spelling errors, tokenization errors, etc. (see §3.2). Note that this philological verification has to be done even when the examples are collected from a standard corpus such as news archives or text databases like Frantext or the BNC. But when the examples are collected from the Web, the problem is their number. There are usually thousands of candidates which occur in millions of contexts. For some examples, one may have to go through hundreds of pages. Checking all the candidates by hand is, therefore, not practicable. Some of the collected examples have to be filtered out automatically. However, the filtering must not be too harsh because speakers are often unsure about how to spell neologisms. For example, *débogage*[2] 'debugging' is also often written *déboggage*, *débugage*, *débuggage*, etc. and the same fluctuation is observed for the corresponding verbs: *débogguer*, *débuguer*, *débugger*, etc.

2.2 Morphological aspects

In all the experiments presented here, we only look for new words that do not belong to the word lists of the common dictionaries, such as the TLFi.[3] We are also concerned only with deverbal nouns, *i.e.* derived nouns that denote the action expressed by the verb such as *production*, deverbal noun of *produce*. We are interested in this class of nouns because (*i*) they have been widely studied, (*ii*) the deverbal nouns and their verb bases share semantic features and distributional properties, (*iii*) they are coined by very productive morphological processes such as the *-age*, *-ion* and *-ment* suffixations, (*iv*) they are easy to identify and therefore easy to check, (*v*) the existing Verbaction database can be completed with our experiments, and we can use its current content for boostrapping.

French deverbal nouns can be coined by suffixation or conversion (*i.e.* non affixal derivation) such as *marcher* 'to walk' > *marche* 'a walk.' A wide range of suffixes can be used: *-age* (*nettoyer* 'clean up' > *nettoyage* 'cleaning up'); *-ion* (*organiser* 'organize' > *organisation* 'organisation'); *-ment* (*payer* 'pay' > *paiement* 'payment'); *-ade* (*ruer* 'to buck' > *ruade* 'a buck'), *-ance* (*venger* 'retaliate' > *vengeance* 'retaliation'); *-ence* (*affluer* 'flock' > *affluence* 'crowds'); *-ure* (*couper* 'to cut' > *coupure* 'a cut'), etc. Even evaluative suffixes can be used as *-ette* in *bronzer* 'suntan' > *bronzette* 'sunbath'.

The high productivity of nominalization shows up in the diversity of the registers the deverbal nouns belong to. Some of them are quite common and are just missing in the main dictionaries such as *labellisation* 'labelization'; other belong to special purpose languages as *débasage* 'debasing' (chemistry); *étrangéisation* 'make something become foreign' (philosophy); *ballonisation* 'floppy syndrome' (medicine), etc. Slang words have been also collected such as *gamellage* 'fall'.

In the following, we focus only on the nouns coined by *-age*, *-ion* and *-ment* suffixations. These nouns can therefore be searched and found on the basis of their endings: *-age*, *-ion* and *-ment* in the singular and *-ages*, *-ions* and *-ments* in the plural. However, this criterion is insufficient because of all the error sources discussed in §3.2, one of them being that many non-French nouns have these endings such as English *carriage*, *colonization* or *commitment*. One technique that can be used to find out if a word is a French deverbal noun or not is to look for contexts where it cooccurs with its possible base verb. This method

[2] *Débogage* is the term recommended by French authorities.

[3] www.atilf.fr/tlfi.htm

has been used by Xu and Croft (1998) in order to select morphologically related words that co-occur in a 100-words window. This kind of co-occurrence has also been studied by Baayen and Neijt (1997) who showed that the contexts where derived words occur often contain anchors used as clues for the interpretation of these words.

In the experiments we have run, the co-occurrence is looked for in the entire web page. For instance for a candidate as *débasage*, we will search for pages where it occurs with one of the following verb forms:

débasa débasai débasaient débasais débasait ... débases débasés débasez débasiez débasions débasons.

This technique is effective for two reasons: (*i*) it rejects many errors because the chances for a erroneous candidate to co-occur with a word similar but having a verb inflexional ending are quite low; (*ii*) if we suppose that documents have a good thematic and referential continuity, then the deverbal noun candidate and its base verb candidate have good chances to be semantically close.

3 Overview of the method

The experiments presented in this paper use the same method. The acquisition of the deverbal nouns and their base verbs is performed in three steps. In the first one, we look for words that are likely to be deverbal nouns. In the second one, we determine the inflected forms of their possible verb bases. In the third, we look for contexts where the deverbal noun candidates co-occur with one of these hypothetical verb forms.

3.1 A 3 steps approach

The first step of the general method is to look for words that are likely to be deverbal nouns. There are several ways to find them. When one has access to an entire index or to an entire corpus, these candidates can be identified by their endings. But when we do not have access to the index of the engine or the corpus, other techniques must be used in order to predict word forms that are likely to be deverbal nouns. The first one is to generate word forms by suffixing verb stems (*miroiter* 'shimmer' > *miroitage* 'process of making a surface become sparkling') and also stems that belong to other categories such as adjectives (*machinal* 'mindless' > *machinalisation* 'act of making something become mindless') or nouns (*mercenaire* 'mercenary' > *mercenairisation* 'mercenarization'). The generation of the word forms can be done as presented in (Hathout et al., 2002) or by means of the method described in the next paragraph.

In the second step, we assume that the candidates collected in the first step are deverbal nouns and we predict the inflected forms of their verb bases. For instance, for a candidate such as *débasage*, we generate the forms listed in §2.2 by using the morphological knowledge available in Verbaction. Our method is word-based (Bybee, 1985): we have associated with every noun of Verbaction all the inflected forms of its base verb. For instance, the noun *rasage* 'shaving' is associated with all the forms of the verb *raser* 'shave'. We then abstracted suffixation schemas from these couples. For instance, the couple (*rasage*, *rasons*) induces the following schemas:

```
rasage/rasons
asage/asons
sage/sons
age/ons
```

where the left-hand side represents a noun ending and the right hand side the verbal ending that has to be substituted for the former in order to get an inflected verb form. The schemas are then projected on the deverbal candidates. The inflected forms are therefore generated in one step. Because we want the prediction of the verb inflected forms to be as precise as possible, we select as model the Verbaction nouns that share the longest ending with the candidate. For instance, the model used for a candidate such as *débasage* is *rasage* and the inflected forms of its base verb (*débaser*) are generated following the example of *raser*.

In the third step, we look for attestations of the predicted inflected forms in pages which also contain the deverbal noun. A single case of such cooccurrence is enough for the noun-verb pair to be considered as valuable and submitted to manual checking: no frequency threshold is used.

3.2 Common problems and solutions

Whatever the method by which they have been harvested, candidate words come along with a lot of noise.

There is a wide litterature on error detection and correction in texts (see for example (Kukich, 1992)). However, distinguishing neologisms from errors is a specific task and processing web pages encounter specific difficulties. We identified the following noise sources and proposed some ways of dealing with them.

- *Spelling errors* are searched for with simple

Errors (%)	*-age*	*-ages*	*-ion*	*-ions*	*-ment*	*-ments*	All
Wrong part-of-speech	2.88	4.27	2.63	8.70	19.82	1.55	7.27
Tokenization error	0.82	1.71	3.95	13.83	12.78	8.53	7.35
Wrong language	3.29	6.84	5.70	5.53	24.67	31.78	11.70
Morphological error	7.00	11.11	6.14	3.95	1.32	2.33	5.01
Misc. spelling error	17.28	16.24	12.28	16.21	25.11	27.91	18.63
Correct	**68.72**	**59.83**	**69.30**	**51.78**	**16.30**	**27.91**	**50.04**

Table 1: Remaining error types for 6 deverbal noun endings

methods, for most of the genuine new words can be false positives if the correction is too greedy. Therefore we limited our algorithm (brute-force approach with a standard French dictionary) to simple editions, *i.e.* mostly to accents and repeated letters.

• *Tokenization errors* are of different types, such as extra spaces inserted in a word, or missing spaces (collided words). Both can come from the original web page, from an encoding error, or from the text conversion (especially from PDF files). We developed specific programs to detect these different situations, using both a brute-force approach and a web-based checker. More specifically, when searching for collided words, we check if an inserted space would lead to two existing words. In this case, we automatically query an online search engine to get the number of documents of the compound and split version. For example, when investigating *applaudissage* 'applauding', we examine the possibility of a missing space leading to *applaudis+sage* 'applause+wise'. The former gives 20 hits, the latter none: our conclusion is that *applaudissage* is a genuine word. On the contrary, *bulletinpage*, suspected to be a collision between *bulletin* and *page* is discarded because *bulletin+page* has 585 hits, compared to the 24 for *bulletinpage*. The same process is applied to search for extra spaces.

• *Proper names* are of no interest to us: they are discarded along with any word written in capitals.

• *Foreign language* contexts are dealt with by configuring the search engine (if any) accordingly, and by applying a stopwords-based language detection routine on the immediate context of a candidate word. However, both these methods are unsuccessful when applied to closely related languages such as Latin, Old French, Occitan, Catalan, etc. Ranaivo-Malançon (2006) studied the case of Malay and Indonesian by adding rules (based on number formats and exclusive words) to classic ngrams methods (Cavnar and Trenkle, 1994). Unfortunately, this attempt is language-specific and seems to be unfit for short contexts.

• *Computer code* is a common situation where the candidate word is in fact a variable or function name. We filter them out with the same language detection routine, as we added to our list of foreign stopwords such code-related strings as *function*, *var*, *begin*, etc. E-mail addresses and URLs are detected with simple regular expressions.

• A number of web pages are *spam documents* which can contain randomly generated strings. Although the detection of such pages is difficult, they have been more and more effectively taken into account by search engines. We nevertheless implemented a few tests, such as the detection of simple word lists (based on the fact that all words appear in the lexicographical order).

• Some candidate words belong to a *wrong part-of-speech*, such as words in *-ment* that are adverbs and not nouns (although they could be of interest in another study). Their detection would need at least some kind of automated linguistic annotation, such as part-of-speech tagging, which would be extremely ineffective in these precise situations. Dealing with unknown words when processing corpora relies on quite crude techniques, such as word-guessing, which itself relies on suffixes. POS tagging these contexts would simply lead us to consider all new *-ment* words as adverbs. Thus, this kind of error can only be solved by manually checking the contexts.

• In some cases, the base verb detection can lead to *morphological errors*. These appear when the morphological process coins the noun from something other than a verb, but which the base prediction algorithm falsely detects as such. For example, *blagounettage* 'the making of small jokes' is coined from the noun *blagounette* 'small joke', but the predicted verb *blagounetter* does not exist. Unfortunately, one of the inflected forms of this hypothetical verb is *blagounette*, thus giving a false positive because of this homography.

Overall, the filtering methods are not sufficient, and the results need to be checked manually. The breakdown of the different *remaining* error sources can be seen in table 1, for 6 different word endings. This is the result of a manual validation of 1,197 couples extracted with the third method

described below (§ 5.3). As can be seen, there are important variations between suffixes. The most difficult to process is *-ment*, with only 17% precision, mostly due to the fact that this suffix is used to coin adverbs (hence the 20% POS-related error rate) and is very common in closely related languages. On the other end of the scale, *-age* and *-ion* both lead to nearly 70% precision.

It is also known that these automatic filters are overzealous, and that some correct words are discarded, but our main objective in this process is to achieve a reasonably high precision, in order to minimize manual validation.

Before presenting the actual experiments and contexts in which we used these methods, we will now take a look at the recent evolutions that led us to adapt our approach to a changing world.

4 Evolutions in using the Web as a Corpus

Corpus linguistics researchers, used to struggle to build large corpora, facing money-, time- and copyrights-related questions, realized in the early 2000s what huge, freely and easily available source of language data the web is. From that time, both technical ways to access the web and the researcher's outlooks on its use has evolved simultaneously. We briefly recall hereafter the different steps of the WAC background.

4.1 Finding a way to the wild web

Search engines (SE) came after web directories and more features have been developed while the scope of the indexed pages underwent a tremendous increase. Some engines such as Altavista, born in 1995, enabled the user to build sophisticated queries (see §5.1). Initially, the way to automate the querying of a SE was to simulate a browser's behaviour: by submitting a query with suitable parameters and parsing the results page. Year 1998 has seen the birth of Google and 5 years later, Altavista was bought twice, causing the loss of its advanced features. The SE companies started to control automated querying by developing search APIs, providing a handy way to a massive use of SE from programming languages. Nevertheless, this solution came up with some important constraints such as a maximum number of queries per day per IP.[4]

Today, whereas the search APIs are still working with previously delivered keys, no more new licenses are delivered (Google) and finding the old API is not immediate (Yahoo). The services have been replaced by products[5] intended to develop integrated web services embedded in web pages, not suitable for our task. Only Microsoft Live Search's latest API is still supported.[6] Fletcher (2007) has shown how he used it as a starting point to build a BNC-comparable corpus.

To cope with APIs restrictions and sudden changes in SE's policies, designing non-retail crawlers seems to be the ideal solution. Castillo (2004) studied how to make crawling *effective*. Among several available spiders, Heritrix is an opensource and free software, and is probably the most complete one. We will see in §5.2 that succeeding in such a scheme is a thorny issue.

4.2 The WAC initiative: from distinct goals to common challenging issues

As the practical details of the access to the web changed, the WAC problematics evolved too. Nobody wonders *"is the web a (good) corpus?"* any longer. Kilgarriff and Grefenstette (2003) already answered in the early stages and the question switched to *"is the web a corpus suitable for my task?"* The whole community usually agrees on the legitimacy of using the web. It is sometimes the only reasonable-sized source of linguistics material at disposal. The Crúbadán project (Scannell, 2007), for example, resulted in the automatic development of large text corpora for minority languages, and may not have been possible without recourse to the web.

The researchers' individual aims vary widely, from extracting large amounts of named entities to building classical general-purpose corpora. There is a also a wide range in the way they take advantage of the web. For example, Keller and Lapata (2002) use Google's result counts to retrieve frequencies of part-of-speech bigrams while Sharoff (2006) generates queries made of selected words and fetches the result pages to build large corpora. A common shared issue, apart from the way the corpus is collected and used, is the process of cleaning a messy set of pages. It has been pre-

[4] 1000 queries for Google SOAP Search API and 5000 queries for Yahoo Search API, never going beyond 1000 pages for a given query. The *per IP* restriction really matters when all workstations located behind a firewall are seen as having the same IP by the SE's server.

[5] Yahoo BOSS API and Google Ajax API.

[6] With 25000 queries per day per *application*, it is the most permissive.

sented as a tedious and unglamorous engineering task, but is a crucial bottleneck one has to deal with before using web data. The Cleaneval competition (Baroni et al., 2008) arose in year 2007 and could result in a joint effort to provide methods and tools. Unsurprisingly, even this low-level task raised non-trivial questions. Just to mention one, the task of boilerplate removal pointed out a divergence on defining what *"textual data of no linguistic interest"* means. The portion of quoted text after '>' in a forum post may skew statistical results of a lexicometry study whereas it may be relevant to keep it in a discourse-oriented analyse.

Our approach, confronted to these questions, is more straightforward as we do not try to build a balanced corpus, nor do we use frequency counts in any way.

5 Three different approaches

We will now present how we technically adapted our search for derived words along these years and evolutions. We will focus on our most accomplished objective, extending the Verbaction database (§2.1).

5.1 Webaffix: using AltaVista's wildcards

The first large-scale campaign we launched (in 2001) was based on a program named Webaffix (now unfortunately obsolete), as described in (Hathout and Tanguy, 2002).

This program took advantage of the wildcard querying capability provided at this time by the Altavista search engine, which allowed for example to query for *bra*age* to get documents containing words beginning with *bra* and ending with *age*. The only restriction was that the wildcard meta-character needed to be preceded by at least 3 letters. We bypassed this constraint by building the 3000 plausible trigrams found at the beginning of French words. Another advantage of this regretted search engine was the almost unlimited query length, which allowed us to add a negative clause to the query, excluding known words from the query. A typical query would then be:

```
aqu*age -aquaplanage -aquarellage
```

(*aquaplanage* and *aquarellage* being the only two French words in our dictionary beginning with *aqu* and ending with *age*.

At this time, Altavista could be automatically queried with no restriction or quota (except for a self-imposed curtesy policy of waiting 2 seconds between queries). Each resulting web page then had to be downloaded and analysed: first to actually identify the new word candidate (no snippets were provided by Altavista), and to check for errors, as described in §3.2. This lead to the analysis of about 120,000 web pages, a process taking around 150 hours. This stage provided a list of 13,500 new nouns candidates.

Each of these words were analysed to predict their matching base verb, and thus produced 13,500 new queries, where both the candidate noun and one of its inflected base verb forms were searched for in the same document. Each resulting document was analysed to once again filter out a number of errors. As a final result, this campaign provided 1,821 new noun-verb pairs, which were finally submitted to a manual validation process, which left 926 correct ones (51%).

5.2 Trifouillette: a home-made dedicated crawler

However, these first experiments could not be continued, as Altavista stopped allowing wildcards in 2003. We then simply -and naively- decided to design our own crawler: *Trifouillette*. The principle seems pretty simple: from a given seed of URLs, fetch the pages, parse them, extract the interesting words if any, extract the links and start again.

We studied the existing crawlers but even Heritrix did not meet our needs. First, at this time, nothing was done to detect and handle spider traps.[7] Moreover, we wanted a light architecture dedicated to our task, namely not building a huge corpus, but rather gathering a collection of "interesting" pages (containing lexical creations) and storing the occurrences in a database, thus getting to the heart of the matter. This architecture enabled us to crawl and process up to 600,000 pages a day on a single machine. The NLP part of the work, though not straightforward, was usual. The pages analyser implemented the filtering heuristics described in §3.2. Conversely, the management of the crawler required unexpected daily maintenance to a discouraging extent. To spend time dealing with non-compliance with standards (servers, pages) is fair game. Cleverly handling spider traps is crafty. But using the HTTP response header to speed up the process of discarding non-French pages and discovering that all personal pages from the `free.fr` domain are assumed to be in Polish because of a misconfigura-

[7] Still today, the user manual only mentions the detection of URLs with repeated patterns or too many path segments.

	-age	*-ages*	*-ion*	*-ions*	*-ment*	*-ments*	All
Unfiltered new word forms							
Forms	48,217	12,263	158,181	38,358	71,795	11,399	340,213
Web pages	543,060	112,869	1,270,059	377,085	902,426	372,705	1,801,445
Automatic filtering							
N-V pairs	750	117	1,678	272	1,170	129	4,116
Web pages	6,862	609	17,499	2,065	28,603	5,983	53,647
Manual filtering (= estimation)*							
N-V pairs	515*	70	1,163*	141*	191*	36	2,060*
Web pages	2,954*	235	9,450*	1,733*	448*	222	14,580*

Table 2: Overview of the filtering process on Exalead Corpus

tion of the web server[8] is a bit frustrating... We also had to deal with recurrent local network dysfunctions until a new firewall made our crawler inoperative and required other modifications.

We gave up the Trifouillette project in 2006 due to a lack of time but continued to use the tools we designed as a basis for developing new specific applications.

5.3 Working with Web professionals: using Exalead's corpus

Taking advantage of a research collaboration with the Exalead company,[9] we got access in 2008 to a ready-to-use corpus of French web pages. Founded in 2000, Exalead is a software provider in Web search markets that launched in year 2006 a public search engine which indexes today 8 billion pages and is a keystone of the Quaero program.[10] The company provided us with a sample corpus made of 2.5 million pages identified as French, handling the language detection, the character encoding and the conversion into raw text. The 20GB of text pages contain 3.3 billion words, that we tokenized and indexed in a database.

Our method followed the same principles as the late Webaffix program (§5.1): we first selected word forms ending with either *-age*, *-ion* or *-ment* (or their plural counterparts) which did not appear in our French dictionary, nor in the Verbaction database. This gave us 340,213 word forms. Table 2 shows the breakdown between the 6 different word endings and the number of different web pages used to find the candidate word forms.

We then applied our filtering methods (described in §3.2), base verb prediction, and search for cooccurrence between noun and verb. This led to 4,116 new noun-verb pairs. Manual filtering on a sample of 1,197 couples by three different judges led to 599 valid pairs. The overall ratio of correct pairs is 51%, with important variations between suffixes, as explained in §3.2. Although the entire list has currently not been manually validated, it gives us a good insight at both the expected results and the general process.

First, it shows that the selected suffixes continue to provide a seemingly endless stream of new words. If our estimation is correct, the Verbaction database (currently containing 9,393 pairs) will grow by 22% with these results. Almost all new words we identified correspond to recent technical or social evolutions, as shown by these few selected examples:

- *wiitage* - *wiiter*: playing the Wii console (*i.e. wiiing*). The Wii was commercially launched in 2006.
- *sarkoïsation* - *sarkoïser*: being influenced by Nicolas Sarkozy (now French president). The word was coined by a French football player in 2006 and has been frequently used since.
- *télédéclaration* - *télédéclarer*: declaring one's income online. This has been made possible by the French tax office in 2001.
- *wambement* - *wamber*: using the social networking website Wamba (launched in 2007).

Second, it clearly shows the amount of raw data needed to extract useful information. Our estimation is that one web page out of 200 contains a new valid word pair. However, automatic filtering is quite effective in reducing the amount of data that needs to be examined manually.

6 Conclusion

As shown in these last results, we have been successfully searching for new French derived words in an ever-evolving Web. We now have the most extensive collection of French deverbal nouns available in the community. Starting 8 years ago with the opportunity to submit sophisticated queries to a compliant search engine, we tried to get along without it when it disappeared, before realising what a difficult task web-crawling is, and

[8] the pages were generated with Perl (`pl`) and the administrator probably misunderstood the role of the `Content-Language` header.

[9] `www.exalead.com`

[10] `www.quaero.org`

how it needed an industrial approach, which can only be provided by commercial search engines.

Along these different stages, our method has remained the same, our main effort being the filtering out of the erroneous contexts found in web pages. However, this evolution takes us back to a more traditional corpus approach. This has several benefits: we are less constrained in our searching (for example, the AltaVista method could not have found *wiitage*, because *wii-* is not plausible as a French word beginning), and we can now have an estimation of the huge amount of raw data necessary to get some useful linguistic material. The only visible counterpart is the bulk of data to be processed (dozens of GB and a dedicated database), while the original Webaffix program was lightweight.

This evolution also raises many methodological questions: we now are in the position to perform more sophisticated corpus linguistics inquiries on our data, such as studying more thoroughly the contexts.

Acknowledgements

The authors would like to thank Exalead for kindly letting us use their corpus and thus unburdening us of a significant part of the work.

References

R. H. Baayen and A. Neijt. 1997. Productivity in context: a case study of a Dutch suffix. *Linguistics*, 35:565–587.

R. H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. The CELEX lexical database (release 2). CD-ROM. Linguistic Data Consortium, University of Pennsylvania, Pennsylvania, USA.

M. Baroni, F. Chantree, A. Kilgarriff, and S. Sharoff. 2008. Cleaneval: a Competition for Cleaning Web Pages. In *Proceedings of LREC*, Marrakech.

D. Bourigault and C. Fabre. 2000. Approche linguistique pour l'analyse linguistique de corpus. *Cahiers de Grammaire*, 25:131–151.

J. L. Bybee. 1985. *Morphology. A Study of the Relation between Meaning and Form*, volume 9. John Benjamins Publishing Company, Amsterdam.

C. Castillo. 2004. *Effective Web Crawling*. PhD Thesis, Dpt of Computer Science, University of Chile.

W. B. Cavnar and J. M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of SDAIR*, pages 161–175, Las Vegas.

W. H. Fletcher. 2007. Implementing a BNC-compareable Web Corpus. In *Building and Exploring Web Corpora, Proceedings of WAC3*, Louvain-la-Neuve.

N. Habash and B. Dorr. 2003. A categorial variation database for English. In *Proceedings of NAACL/HLT*, pages 96–102, Edmonton. ACL.

N. Hathout and L. Tanguy. 2002. Webaffix: a tool for finding and validating morphological links on the WWW. In *Proceedings of LREC*, Las Palmas.

N. Hathout, F. Namer, and G. Dal. 2002. An Experimental Constructional Database: The MorTAL Project. In Paul Boucher, editor, *Many Morphologies*, pages 178–209. Cascadilla, Somerville, Mass.

N. Hathout, M. Plénat, and L. Tanguy. 2003. Enquête sur les dérivés en *-able*. *Cahiers de grammaire*, 28:49–90.

N. Hathout, F. Montermini, and L. Tanguy. 2008. Extensive data for morphology: using the World Wide Web. *Journal of French Language Studies*, 18(1):67–85.

H. Jing and E. Tzoukerman. 1999. Information retrieval based on context distance and morphology. In *Proceedings of SIGIR*, pages 90–96, Berkeley, CA. ACM.

F. Keller and M. Lapata. 2002. Using the web to overcome data sparseness. In *Proceedings of EMNLP-02*, pages 230–237.

A. Kilgarriff and G. Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29:333–347.

K. Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439.

A. Lüdeling, S. Evert, and M. Baroni. 2007. Using Web data for linguistic purposes. In Hundt, Nesselhaut, and Biewer, editors, *Corpus Linguistics and the Web*, pages 7–24. Rodopi, Amsterdam.

M. Plénat. 2000. Quelques thèmes de recherche actuels en morphophonologie française. *Cahiers de lexicologie*, 77:27–62.

B. Ranaivo-Malançon. 2006. Automatic identification of close languages - Case study: Malay and Indonesian. *ECTI Transaction on Computer and Information Technology*, 2(2):126–133.

K. P. Scannell. 2007. The Crúbadán Project: Corpus building for under-resourced languages. In *Building and Exploring Web Corpora, Proceedings of WAC3*, Louvain-la-Neuve.

S. Sharoff. 2006. Creating general-purpose corpora using automated search engine queries. In Baroni and Bernardini, editors, *Wacky! Working Papers on the Web as Corpus*. GEDIT.

J. Xu and W. B. Croft. 1998. Corpus-based stemming using co-occurrence of word variants. *ACM Transaction on Information Systems*, 16(1):61–81.