Certification and cleaning up of a text corpus: towards an evaluation of the "grammatical" quality of a corpus

Cyril Grouin

LIMSI–CNRS
BP 133, F-91403 Orsay Cedex
cyril.grouin@limsi.fr

Abstract

We present in this article the methods we used for obtaining measures to ensure the quality and well-formedness of a text corpus. These measures allow us to determine the compatibility of a corpus with the treatments we want to apply on it. We called this method "certification of corpus". These measures are based upon the characteristics required by the linguistic treatments we have to apply on the corpus we want to certify. Since the certification of corpus allows us to highlight the errors present in a text, we developed modules to carry out an automatic correction. By applying these modules, we reduced the number of errors. In consequence, it increases the quality of the corpus making it possible to use a corpus that a first certification would not have admitted.

1. Introduction

Due to the development of the Internet, the constitution of large scale corpora is not any longer an obstacle to statistical analysis of linguistic phenomena. The multiplication of personal Web pages, newsgroups, forums, blogs and electronic mail offers possibilities to gather a very large number of documents for corpus linguistics (Habert et al., 1997). In parallel to easy access to the data, the NLP community is provided with a set of powerful tools which make it possible to handle these corpora, whatever the linguistic field concerned and the objective of the study (Habert et al., 1998).

All the corpora we can obtain over the Internet do not necessarily offer a basic quality sufficient for textual analysis. The corpora generally include a variable number of errors – these errors can be of typographical, orthographical, or syntactic nature – which can generate noise when using automatic language processing tools (Hofmann and Weerkamp, 2007). The problem of the quality of the data rises up in the industrial world, in particular in marketing services where cleaning data bases is relevant in order to improve the client relationship and to reduce the costs of maintenance of customer information (Clément and Laboisse, 2007).

In order to measure the quality of a corpus before any application of linguistic processing, we developed a method of certification of corpus allowing to draw up quality indicators. These indicators are related to three text levels.

- At the typographical level, can improbable character sequences reveal typographical errors (a space before a comma or a dot)?
- At the orthographical level, are the words in a lexicon, at least do they have an acceptable form?
- At the grammatical level, is the text correctly structured? Are the links between the words coherent (grammatical agreement rules, verbs conjugation, etc.)?

Faced with the difficult exercice of error detection – and its corollary, their correction – it is important to keep in mind what is the richness of the natural languages, in terms of vocabulary, rules and exceptions. To launch out into this task is a fascinating game which reveals new problems for each achieved improvement...

2. Detecting errors

Various methods exist for error detection. Choosing a method depends on two factors: on the one hand, the type of error we want to underscore, and on the other hand, the type of resources we can obtain for producing data reference.

2.1. Orthographical errors

(Mitton, 1996) explained that the study of orthographical errors consists in checking, for each word from a text, that this word exists in the language. Two methods allow us to check the spelling of a word: the first one based upon n-gram of words, the other based upon dictionaries.

2.1.1. N-gram method

The objective of this method is to check the existence of a word from the analyzed text in a list of word n-grams that we qualify of reference list. This method supposes that the reference list is the most comprehensive possible, language coverage (number of forms and their derivatives) as well as thematic point of view (general vocabulary compared to specialized vocabulary from a domain).
During the constitution of such a reference list, it is important to make sure that the corpus from which this list will be constituted, contains the fewest mistakes possible. Indeed, any error kept in this list will not be detected, thus distorting the error detection. One way to reduce this persistant error rate is to define a threshold of minimal frequency, below which the words will not be included in the reference list. A low threshold generally allows us to eliminate the hapax legomena which are often errors.
Moreover, in order to ensure the compatibility of the errors detection system with a corpus composed of a vocabulary

coming from a specialized domain (eg. medicine), it is important to foresee the possibility that the user can complete the reference list with his own vocabulary.
The application of this method is both simple and efficient, as long as the reference can be juged of good quality.

2.1.2. Method based upon dictionaries

A second method uses existing dictionaries as reference. This method presents the advantage to use references that do not contain any error.
Usually, the entries of a dictionary are reduced to the lemmas. This particularity implies the necessity of defining rules of derivation that will allow us to check the forms presents in corpus, such as the conjugate forms of a verb, or the agreement in gender and number for an adjective.
As for the n-gram method, the question of the coverage of the dictionaries must be kept in mind while choosing the dictionary reference. We give below the number of entries in some "classical" French dictionaries.

DICTIONARY	ENTRIES
Le Petit Larousse (2008)	59 000
Le Petit Robert (2008)	60 000
Le Grand Larousse (2006)	87 000
Le Grand Robert de la langue française (2001)	80 000
Trésor de la Langue Française Informatisé, TLFi (2004)	100 000

Figure 1: Number of entries in some French language dictionaries.

Some electronical dictionaries do not only present lemmas but also the inflected forms. The lexical called "Morphalou", realized by (Romary et al., 2004), contains 95 810 lemmas and 524 725 inflected forms while the monolingual dictionaries realized within the project EuRADic[1] contains such lemmas and inflected forms for 5 european languages. Each lemma is accompanied by its parts of speech and flexional informations. We give below the number of lemmas and inflected forms for each monolingual EuRADic dictionary.

LANGUAGE	LEMMAS	INFLECTED FORMS
English	171 713	365 823
French	112 216	694 673
German	157 813	17 634 834
Italian	70 951	557 204
Spanish	83 952	838 391

Figure 2: Number of lemmas and flexional forms in the monolingual EuRADic dictionaries.

[1]EuRADic: European-Arabic dictionaries. References ELRA-L0049 (French), ELRA-L0050 (English), ELRA-L0051 (German), ELRA-L0052 (Spanish) and ELRA-L0053 (Italian), more details on `www.elra.info`

2.2. Syntactic errors

The detection of syntactic errors depends on the analysis of the relations between units in a sentence, the set of these relations building the whole sense of the sentence.
Two methods exist for detecting syntactic errors: the first one is based upon the comparison of possible syntactic tag combinations, the second one is based upon the set of the syntactic rules used in a language. (Mitton, 1996) underlined that these two types of analysis tend to send back a much too high number of false errors. We will see that a third method exists, based upon n-grams of words, which, however could not be applied.

2.2.1. Method based upon tags

The method based upon syntactic tags consists in comparing combinations of morpho-syntactic tags. In this sense, it is related to a n-gram method where n-grams are tags.
This method rests on three steps. In the first one, the reference corpus and the corpus to analyse are tagged. In a second time, the occurrence frequencies of tag combination are collected. Then, the lower frequencies of tag combinations in the analyzed corpus are put forward and compared to the reference; these lower frequencies could be syntactic errors.

2.2.2. Method based upon rules

An other method to detect syntactic errors is based upon the language rules. This method consists in applying syntactic rules on the corpus to analyze. (Mitton, 1996) recommends, in case of analysis failure, to make another analysis, making the rule more permissive until the analysis be completed. The impossibility of an analysis implies a syntactic error.
This method, more formal, supposes to have all grammatical rules from a language, which could be tedious due to the numerous exceptions that exist in the languages and the differences of range for each rule: as an example, in the case of an agreement in gender and number between an article and a noun, one or more adjectives could be inserted between them, requiring a certain flexibility of the rule system.

2.2.3. Method based upon n-grams of words

A third method, based upon n-grams of words, allows us to make a syntactic error detection. This method supposes to collect bigrams of words in a reference corpus and then, to compare these bigrams of words to the one of the analyzed corpus. This method presents the advantage to not use tagging system but it implies to have a very huge list of reference bigrams of words. Due to the impossibility to have all the possible combinations of words for a language, this method is not enough efficient to be used. Moreover, it requires a very important process time.

3. Corpus certification

Our work has been realized within the framework of the ANR RNTL SEVEN[2] project whose aim is to produce a software for classifying and visualizing textual documents

[2]SEVEN: claSsification Et Visualisation pour l'Exploration et la Navigation.

– principally customers satisfaction surveys – in order to help enterprise decision making.
Because this software would be shared out, we chose not to use external softwares (such as a lemmatizer-tagger system) and producing our own linguistic resources in order to freely distribute them.
These choices led us to adopt the n-grams methods in order to make the certification and automatic corpus correction.

3.1. Methods

We developed two principal resources for checking a text: rules and reference lists of words and n-grams. These resources are used by different methods depending on the concerned analysis level.

3.1.1. Rules

Initially, we used the good usage typographical rules based upon the recommendations established by the French National Printing works (Imp, 2004). These rules make it possible to define the behavior of space with respect to the punctuation marks as well as the use of the letters in capital in a text. The use of these rules enables us to carry out a certification of corpus on the typographical level.

3.1.2. Lists of references

In order to draw up reference n-gram list, we used a set of articles coming from the French newspaper *Le Monde*. The choice of this newspaper is justified by the fact that it covers several thematic parts (international and national events, art, economy, sciences and sports news) and that it contains few orthographical errors. The set of articles we used have been published between February 2006 and June 2007; it represents about 60 000 articles.

Using a newspaper corpus in order to collect n-grams allow us to dispose, at least, of three kinds of resources that are not provided by "classical" dictionaries: the inflected forms of the words of the language (even if some dictionaries tend to propose them, see section 2.1.2.), the acronyms (names of compagnies), and proper names (last name, trademarks, etc). This last kind of data is especially crucial for the marketing objective of the SEVEN project.

We collected unigrams of words and bigrams of characters from this reference corpus, associating the frequency of occurrence of each n-gram in the corpus. Then, we defined a minimal threshold in order to eliminate the few errors present in the reference corpus. This threshold has been defined by an empirical method, based upon the consultation of collected n-grams; we used a threshold of 4. We insist on the fact that this threshold is valid only for this corpus and that it must be defined again if a new collect was made on another corpus, even if we collected n-grams from the same corpus on a larger period. Thus, we eliminated all n-grams whose frequency was less than 4; these low frequency n-grams (hapax legomena) were generally errors (*contumière* instead of *coutumière, conversio* instead of *conversion, juusqu'ici* instead of *jusqu'ici*), correct words whose usage is rare (*chiroptère, orphéon*) or foreign words (*hukukçu*).

Contrary to traditional dictionaries that are only made of canonical forms of the words (lemmas), this list of unigrams of words makes it possible to have, at the same time, the canonical forms and the inflected terms (variations in gender and number for nouns and adjectives, conjugation of a verb) of the words present in the text. The comparison of the words to be checked in the text with the reference unigram words enables us to make a certification on the orthographical level.

LIST	NUMBER
Unigrams of words (from *Le Monde*)	327 113
Unigrams of words (complementary list)	3 762
Bigrams of characters (from *Le Monde*)	1 892
Bigrams of characters (complementary list)	132

Figure 3: Number of elements in the lists of reference.

We also tried to extend this method of statement of the N-grams to syntactic certification by producing lists of bigrams of words from our reference corpus. We realized that these lists are very bulky – increasing therefore the processing time necessary to the checking – but that they also suffer from an absence of exhaustiveness due to the combinative richness of the language. These two reasons led us to give up this method.
Then, we used specific lists for three kinds of entities:

- List of abbreviations used when taking notes that must be transliterated (*càd* > *c'est-à-dire*, *nov.* > *novembre*);
- List of acronyms found in the corpus that must be put in upper case (*adsl* > *ADSL*, *edf* > *EDF*);
- List of 35 887 cities[3] names. In order to efficiently use this list, we made several preliminary treatments: to introduce hyphens between elements of the city name (*Ablon-sur-Seine*), to put in upper case the initial of the name (but keeping in lower case articles and prepositions: *Aboncourt-sur-Seille*), to reintroduce the accents (*Achères-la-Forêt*) and to translitterate the abbreviations used in cities names (*"st"* > *saint*, *"s/"* > *sur*, *"ss"* > *sous*: *Saint-Aubin-sur-Loire*) ;

A method based upon the N-classes model and successfully tested by (Spriet and El-Bèze, 1997) within the framework of an automatic tool of texts reaccentuation seems to be promising for syntax.

3.2. Corpus diagnosis

3.2.1. Quality indicators

By the use of these various methods, we are able to establish a quantitative corpus diagnosis according to the linguistic processing which will be applied to the corpus. Making a corpus certification produces two kinds of results:

[3] We used the list of French cities coming from the ABU (Association des Bibliophiles Universels): `http://abu.cnam.fr/cgi-bin/go?DICO/cites`

- Initially, we return for each level of analysis several rates:

  – Error rates, i.e. percentage of words and bigrams of characters which we find in the corpus but which are not in the reference. We established this rate on the forms (in the sense of an inflected term of a word of the language) and the occurrences (the repetition of these forms).

$$\text{Error rate} = \frac{\text{distinct n-grams number} \times 100}{\text{total n-grams number}}$$

  – Dispersion rate, i.e. percentage of repetition of an error in the corpus. The distinction between forms and occurrences enables us to measure the dispersion of the errors in the corpus. Indeed, if an error is produced only once in the corpus, its distribution will be weak, which will result in lowering the error rate on occurrences compared to the error rate on forms.

$$\text{Dispersion} = 100 - \left(\frac{\text{distinct n-grams number} \times 100}{\text{distinct n-grams frequency}}\right)$$

- In the second place, we also produced lists of elements (unigrams of words and bigrams of characters) which were identified in the corpus but that the program did not find in the lists of references. These lists make it possible to carry out an *a posteriori* control by checking these words considered as errors by our program but that appear to be "false errors", i.e. correct words which are not present in our list of reference. It is important in this case, to manually integrate these false errors in our complementary lists of reference. The manual completion of these lists can prove to be time-consuming at the first time but it will decrease as certifications are established making it possible to notably improve quality of future certifications.

3.2.2. Using the diagnosis to make a decision

The user has to make a choice based upon the success and error rates returned by the corpus certification:

- To preserve all the corpus in the state, by considering the possible use of NLP tools which can work on less clean corpora;
- To preserve only the passages of the corpus that include less errors;
- To apply automatic correction processing to the corpus;
- To eliminate this corpus to choose a cleaner corpus.

Within the framework of corpus constitution from the Internet, (Ringlstetter et al., 2006) proposed a threshold of 5‰ of errors (that is to say a maximum of 5 errors for 1000 elements considered) as acceptable threshold of conservation of a corpus. We insist on the fact that it is important to keep in mind that any corpus is established from a particular point of view and that the threshold of conservation of a corpus varies according to the process to apply thereafter to this corpus.
On the Seven project, as documents classification is based upon lemmas, it is important to ensure the lowest error rate possible. For this reason, we kept the threshold of 5‰. This rate only allows a very small number of errors, but it ensures a high quality at the orthographical level for the analyzed text.

3.2.3. Choosing a threshold to keep corpora

In order to determine the best threshold to use, we launched corpus certification on four kinds of corpus:

1. Parliament debates trancription (40 710 words). This corpus has been anonymized[4]; in consequence, there is no names of members of Parliament;
2. Articles from the newspaper *Le Monde* (38 829 words). While our reference lists are based upon the newspaper *Le Monde* (articles published between February of 2006 and June of 2007), we choose a different period of publication for this test (a set of articles published between January and March of 2008);
3. Articles from *Wikipedia* (40 133 words);
4. Messages published on a forum over the Internet[5] (22 451 words).

We give below the results of the corpus certification, after having completed the complementary lists of reference with the false errors, before any automatic correction:

	With UC		Without UC	
Corpus	Error	Disp.	Error	Disp.
Parliament			0.02%	0.00%
Le Monde	0.49%	20.21%	0.03%	36.36%
Wikipedia	1.54%	18.28%	0.48%	7.83%
Forum	5.99%	41.65%	5.18%	47.83%

Figure 4: Error rates on occurrences and error dispersion rates on four corpora, taking into account the words whose initial is in upper case (With UC) and not taking into account these words (Without UC).

We made two certifications on the three last corpora, the first one taking into account the words whose initial is in upper case (words which are potentially proper names) and the second one not taking into account this kind of words. In this second case, the corpora are closed to the Parliament debates corpus which has been anonymized.

The error rate on occurrences for the Parliament debates transcription is very low because all transcriptions are proofread. The errors that have been detected are real errors

[4]The Parliament debates corpus has been produced for DEFT'07 *(Défi Fouille de Textes)* campaign on automatic opinion recognition: `http://deft07.limsi.fr/` This corpus is freely available at `http://deft.limsi.fr/`

[5]Short reactions about radio frequencies lists coming from `www.mixture.fr`

(*Bostwana* instead of *Botswana*) or unknown abbreviations (*netcet*).
The error rate for the newspaper corpus is basically equal to the threshold we defined (4.971‰). A lot of errors detected refer to proper names or to foreign words (*Wanding, Orihuela, Eladio, Dunya,* etc.) that are not in the reference list. These must be considered as false errors. If we launch a new certification without taking into account the words whose initial is in upper case (which concerns a lot of these false errors), the error rate is only of 0.03% which is very close to the one of the Parliament debates. In both cases, the corpus could be used without any automatic correction.
The Wikipedia corpus provides an error rate of 1.54% but, as for *Le Monde* corpus, a lot of errors are due to proper names and foreign words (*Yardbirds, Rodham, Toepffer*). A new certification without taking account of words whose initial is in upper case returns an error rate of 0.48% which allows us to use this corpus.
Conversely, the corpus extracted from a forum over the Internet returned error rates higher than the conservation threshold of 5‰. This rate is due to the use of abbreviations (*lol, qq*) and to real errors (accents forgotten: *frequence* instead of *fréquence*, phonetic script : *malorosemen, avouar* instead of *malheureusement, avoir*). This kind of errors seems to be difficult to automatically correct. A certification without taking into account the upper case begining words do not provide an error rate (5.18%) lower than the defined threshold.

In accordance with (Ringlstetter et al., 2006), the threshold of 5‰ seems to be a good limit as long as we want to keep a corpus of quality.

4. Automatic correction

The methods of certification of corpus, because they make it possible to highlight the errors present in a text, led us to work out modules of automatic correction.

For automatic correction, we used the same methods developed for certification of corpus:

4.1. Typographical corrections

We carry out the corrections of typography by using the good usage rules. This method presents the advantage of quickly obtaining good results of typographical rewriting but it remains strongly related to the way in which the text was written and encoded (Fletcher, 2007). We will reconsider the case of certain typographical ambiguities which cannot be raised.

4.2. Orthographical corrections

4.2.1. Basic modules

We correct the orthographical errors by connecting several modules. Each one of these modules carries out a correction and tests the existence of the word thus corrected in the list of unigrams of words of reference. In the event of identification of a form corrected in this list, we stop the correcting process and preserve the identified corrected form. These modules are connected as follows:

- Rewriting of character strings integrating of the phonetic notations (the letter "k" instead of the bigramme "qu" in the error *m'enkikiner*). This first module allows us to transform a *hapax legomenon* into an existing word, even if there are other corrections to make;
- Addition and suppression of accents and diacritic (the error *plutot* is corrected in *plutôt* while the error *nécéssaire* will be corrected in *nécessaire*);
- Reduction of geminated consonants (the error *grossse* is corrected in *grosse*);
- Insertion of missing letters or removal of letters in excess (the error *gourvernement* is corrected in *gouvernement*);
- Calculation of distance between two textual chains founded on the Levenshtein's distance.

4.2.2. Levenshtein distance

Levenshtein distance (Levenshtein, 1965) is a measure of similarity between two strings. This measure is based upon the number of operations (deletion, insertion, substitution) needed to transform a string into another string.

OPERATION	EXAMPLE	DISTANCE
Deletion	*bonjours* > *bonjour*	1
Insertion	*bojour* > *bonjour*	1
Substitution	*bonkour* > *bonjour*	1
Permutation	*bonjuor* > *bonjour*	2

Figure 5: Levenshtein distance by operations.

We consider that the correction is the nearest Levenshtein distance word. In case of several words for the same minimal distance, we choose the nearest alphabetical order word from the erroneous word.
The processing time for this module is more important than the one of the other modules. During our tests, we measured that the Levenshtein distance time of calculation between an erroneous word and each one of the 327 129 reference unigrams of words was of 35 to 40 seconds.

In order to reduce processing time for this module without any loss of corrections, we made two changes.
First of all, we reported the use of this module after all other correcting modules. Consequently, the Levenshtein distance calculation is no more applicated until the failure of the previous modules. This first measure allows us to divide treatment time by three[6].
Secondly, we restricted calculation to the sole reference words begining by the same letter than the erroneous word. Indeed, we noticed that 100% of the erroneous words that have been corrected by Levenshtein distance started by the same letter as the expected correction. Thanks to this second changement, we divided again the processing time by two[7].

[6]For a corpus composed of 859 errors, the total processing time passed from 4 hours 18 minutes to 1 hour 27 minutes.

[7]For the same corpus than previously, total processing time passed from 1 hour 27 minutes to only 39 minutes!

4.2.3. Limit of this method

A short evaluation based upon 50 errors returned precisions of 100% for the geminated reduction module, 94% for the accent adding or removal module, 71% for Levenshtein distance and only 37% for the insertion or removal letters module.

We noticed that the context is needed where the errors have not been correctly corrected. The context allows us to choose the relevant correction. As an example, the error *releve* can be corrected either into *relevé* (noun and past participle) or *relève* (noun and verb). The unigram of words method is not effective for producing the right correction.

A second limit of this method is due to the segmentation of the corrected text into words. We made a correction, word after word. This kind of process does not treat the compound words where hyphen or apostrophe are forgotten: the error "*lorsqu on*" is corrected into "*Lorsqu on*" instead of "*lorsqu'on*". Combination of words with hyphen or apostrophe must be taken into account in order to correct this kind of words.

4.2.4. Preserving errors

In every automatic process, it is important to keep an execution trace. Within the framework of automatic correction, it is essential to come back to the errors, at least for three reasons:

The first reason concerns the system evaluation. In order to measure an automatic correction system (principally in terms of recall and precision), we have to highlight the detected errors and the produced corrections.

The second reason refers to the checking of the results produced by the different process. In our system, we ask the user to check up the corrections made by the Levenshtein distance module. By checking these corrections, the user can decide to keep the automatic correction, to propose an alternative correction, or to come back to the original word (in the case of the "false errors", it is to say words which are correct from the orthographical point of view but which are not in the reference list; consequently, it matters to introduce these words in the complementary reference lists).

A last motivation in favour of preserving detected errors concerns the processing time. For each successfull correction, we save the erroneous form and the associated corrected form. This memory of corrections allows us, in case of identification of the same error in the rest of the corpus, to recover the correction produced previously. By applying this functionnality before any other correcting module (see section 4.2.1.), the system do not make calculations and comparisons already made. Thanks to this memory of corrections, we estimate that the processing time has been divided again, at least, by ten.

We keep a trace of the produced corrections under two distinct ways:

- Stand-off informations in a log tabular file;
- And embeded informations in the corpus file, using XML tag (see figure 6).

In the two cases, we indicate: the original error, the produced correction, the module which made the correction, and if it is the Levenshtein distance module, the calculated distance between the two words.

La section "conséquences de l'inflation" en montre les effets de manière assez détaillée en <erreur token="focntion" module="similarite" dist="2">fonction</erreur> des différents groupes.

Figure 6: Preserving errors in corpus. The error *"focntion"* has been corrected in *"fonction"* by the module of similarity where the Levenshtein distance is of 2 operations (a first substitution of "c" by "n", *fonntion*, and a substitution of the second "n" by "c", *fonction*).

4.2.5. Interface of validation

Because of the higher number of bad corrections returned by the module of Levenshtein distance, we developed a graphic interface allowing the user to validate the corrections suggested or to enter an alternative correction if necessary.

This interface takes back all the errors that have been corrected by the Levenshtein distance module (and only the corrections produced by this module in order to avoid the checking of a too high number of corrections) and shows: the erroneous word, the correction produced by the module, and the sentence where this error appears.

4.3. Grammatical corrections

We made the choice to not use external plugins, in order to easily redistribute this system. In this way, we resorted to the use of rules for producing grammatical corrections.

After a thorough study of the types of grammatical errors, we realized that the errors of number agreement were more frequent (96,9%) than the errors of gender agreement (3,1%). We produced a syntactic module of correction based upon a contextual study of the vicinity of the words between them (Pinot, 2005). We thus test the treating of any word in the plural, when such words follow a determiner (in fact, a list of articles or prepositions contracted in the plural *aux, les, des*). However, let us specify that we drew aside this last method for two reasons: first, errors of labelling, and secondly, methodological problems.

4.4. Other corrections

Apart from typographical, orthographical and syntactic errors, there is one type of errors which is less evident to detect and to correct: the case of the words which are correct on the three previous levels but which do not make sense in the context of the analyzed sentence.

This kind of errors is called "malapropisms" by (Hirst and St-Onge, 1998) who developed a method based upon the synset in WordNet for detecting these malapropisms (*an ingenuous machine* instead of *an ingenious machine*). This

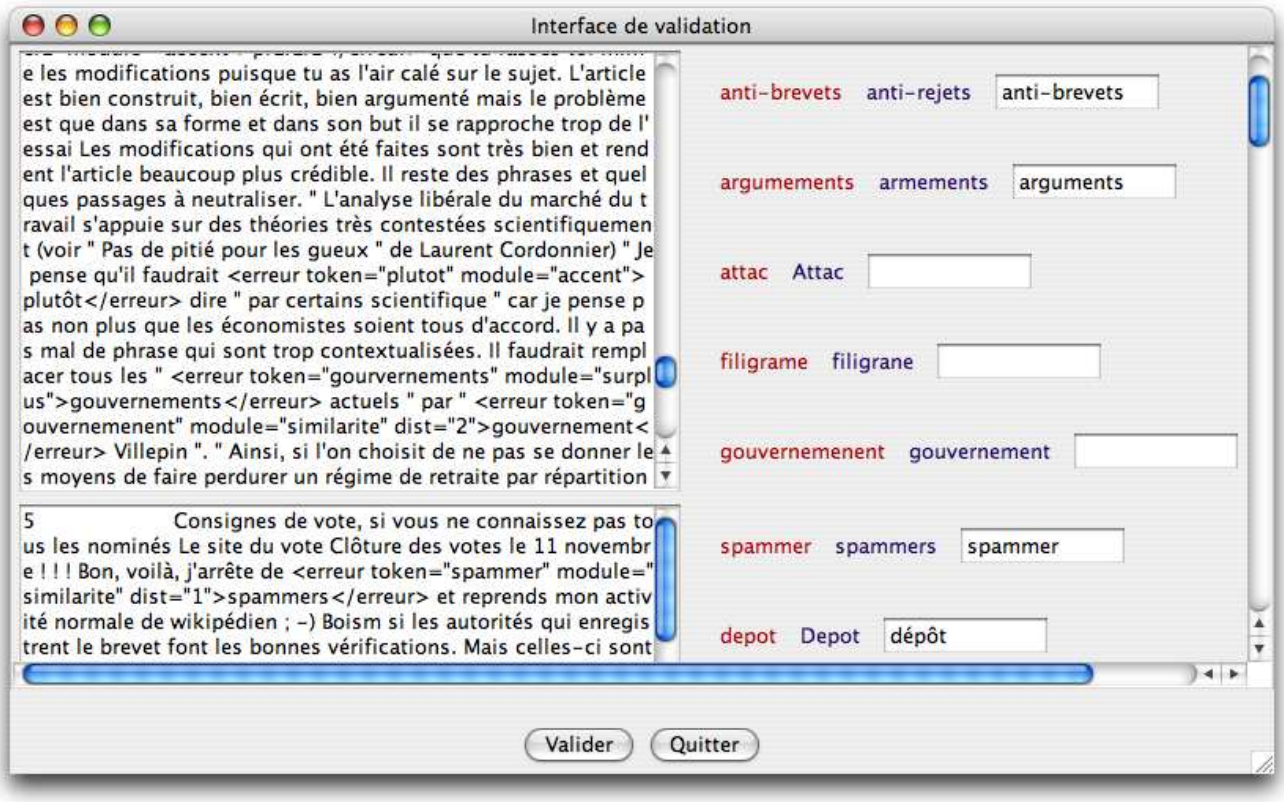

Figure 7: Interface validation of the automatic corrections. In red, raised errors, in blue automatic corrections. If the user accepts the correction suggested, he leaves the box empty, else, the user enters an alternative correction in this box. The left-hand column displays the sentence from which the errors are extracted.

method depends on the identification of possible semantic relations (hyponym, superordinate, association of ideas) between malapropism and real word.
We did not find such kind of malapropism words in our corpus when checking the list of erroneous words and the corrected corpus. In this way, we did not try to apply this method. But it constitutes a track to explore in order to improve the results of the system.

4.5. Results

We launched the corpus certification on a corpus of 509 discussion pages coming from Wikipedia. This corpus is composed of 22 509 different forms (in the sense of inflected forms) and 303 949 occurrences (ie. the total number of forms used). We made two corpus certifications: the first one on the basic corpus, and the second one after having made the automatic correction.
The basic corpus contains 19,17% of forms errors and 2,11% of occurrences errors (6 424/303 949). This error rate on occurrences is higher (21,1‰) than the threshold of 5‰ we defined as a threshold to keep corpus (see section 3.2.2.). In this way, the corpus could not be used for other natural language processing tools.
We made an automatic correction on this corpus and then, we launched a new corpus certification. This new certification returned an error rate on forms of 3,49% and an error rate on occurrences of 0,28% (839/303 832), which is now lower (2,8‰) than the threshold of 5‰. This new rate allows us to use the corrected corpus in NLP tools.

	CORPUS CERTIFICATION	
	BEFORE CORRECTION	AFTER CORRECTION
FORMS		
– TOTAL NUMBER	22 509	20 535
– ERRONEOUS	4 314	717
OCCURRENCES		
– TOTAL NUMBER	303 949	303 832
– ERRONEOUS	6 424	839

Figure 8: Results of corpus certification on Wikipedia corpus, before and after automatic correction.

The errors that have not been corrected in the Wikipedia corpus are of different kinds:

- Real errors not corrected: *bouc-emissaire* (“bouc émissaire”), *Mastrirt* (“Maastricht”), *non-consomptible* (“non consommable”);
- Hapax legomena that do not exist in our reference list: *chercheur-PTT-fonctionnaire, ONUsement, zajoutez, zarticles*;
- False errors which must be integrated in the complementary reference lists (see section 3): *contre-*

propagande, Evidences;

- User name in Wikipedia discussion pages: *Fredcoach, Pgreenfinch, Powermonger*;

- Wikipedia specific terms: *Wikilove, Wikiquote, Wikinews, Wiktionnaire*. In this case, the specific terms from a domain must be added to the reference complementary list of unigrams of words;

- Foreign words: *Bagadou-Strolladou, sonnerezh* (from Breton language).

Conclusion

Whereas the constitution of corpora in linguistics is not any longer an obstacle, the problem of the quality of the documents available gives rise to significant linguistic processing upstream. We set up a method of corpus certification that allows us to draw up a qualitative diagnosis of any corpus. The description of the typographical, orthographical and syntactic errors present in the corpus led us to develop modules of automatic correction.

The first evaluations relating to a corpus of about thirty discussion articles published on Wikipedia returns an accuracy of 0,79 and a recall of 0,90. This corpus integrates many references to knowledge of the world (names of particular associations) which is not referred in our lists and which must thus be added. We also raise that this type of corpus is rich in typing errors and omitted letters, increasing the difficulty of the task.

During the development of this system, we have been particurlarly confronted with the problem of the availability of the linguistic resources, notably within the perspective of freely distributing them. This juridical aspect of the data leads us to develop our own reference lists.

For the lists available and that we can freely distribute, a cleaning and data formatting step has been proved essential. This was the case for cities names list for which we put in upper case the initials, we introduced hyphens, we add accents and we translitterated the abbreviations.

This system has for immediate vocation to be integrated inside a huge system of documents classification and visualization; in this way, we put the stress on this final objective. For this reason, we developed the graphical interface of corrections validation for the Levenshtein distance module. Nevertheless, the system and this interface have been designed only from the user's point of view. There is no administration interface for introducing, as an example, the false errors in the complementary reference lists. Actually, this addition must be done manually via a text editor.

New solutions must be achieved and tested in order to improve the grammatical correction (presently a basical treatment based upon a few number of gender and number agreement rules).

5. Acknowledgements

This work has been done within the framework of the SEVEN[8] project, held by the ANR (project number: ANR-05-RNTL-02204 (S0604149W)).

6. References

Delphine Clément and Brigitte Laboisse. 2007. Création d'un référentiel d'indicateurs de mesure de la qualité des données CRM. In *Actes du 3^e atelier Qualité des Données et des Connaissances*, pages 5–14, Namur, Belgique. En conjonction avec EGC 2007.

William H. Fletcher. 2007. Toward cleaner Web corpora: recognizing and repairing problems with hybrid online documents. Corpus Linguistics 2007, Birmingham, 27–30 July.

Benoît Habert, Adeline Nazarenko, and André Salem. 1997. *Les linguistiques de corpus*. Armand Colin/Masson, Paris.

Benoît Habert, C. Fabre, and F. Issac. 1998. *De l'écrit au numérique : constituer, normaliser et exploiter les corpus électroniques*. Masson Éditeur/Interéditions, Paris.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: an electronic lexical database*, Language, Speech and Communication, chapter 13, pages 305–332. The MIT Press, Cambridge, Massachusetts.

Katja Hofmann and Wouter Weerkamp. 2007. Web Corpus Cleaning using Content and Structure. *Cahiers du Cental*, 5.

Imprimerie nationale, Paris, 2004. *Lexique des règles typographiques en usage à l'Imprimerie nationale*, 3ème edition, mai.

Vladimir Levenshtein. 1965. Binary codes capable of correction deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848.

Roger Mitton. 1996. *English spelling and the computer*. Birkbeck ePrints, London. Available at: http://eprints.bbk.ac.uk/archive/00000469.

Guillaume Pinot. 2005. Correction orthographique en contexte. Technical report, LINA. Rapport de stage.

Christoph Ringlstetter, Klaus U. Schulz, and Stoyan Mihov. 2006. Orthographic Errors in Web Pages: Toward Cleaner Web Corpora. *Computational Linguistics*, 32(3):295–340, September.

Laurent Romary, Susanne Salmon-Alt, and Gil Francopoulo. 2004. Standards going concret: from LMF to Morphalou. In *Workshop on Electronic Dictionaries, Coling 2004*, Geneva.

Thierry Spriet and Marc El-Bèze. 1997. Réaccentuation automatique de textes. In *Fractal 97*, Besançon.

[8]SEVEN: claSsification & Visualisation pour l'Exploration et la Navigation.