AUTOMATISATION OF THE ACTIVITY OF TERM COLLECTION IN DIFFERENT LANGUAGES

Bruno Cartoni*, Pierrette Bouillon*, Yalina Alphonse* and Sabine Lehmann
bruno.cartoni@eti.unige.ch, {pierrette.bouillon, yalina.alphonse}@issco.unige.ch, sabine@acrolinx.com
ISSCO / TIM*, Ecole de traduction et d'interprétation, Université de Genève, Boulevard du Pont-d'Arve, CH-1211 Genève 4, http://www.issco.unige.ch
acrolinx GmbH, Novalisstr. 12, D-10115 Berlin, http://www.acrolinx.com

Abstract

This article describes the use and development of a tool for grammar and terminology control (FLAG), for the purposes of automating the verification of terminology for a large-scale user of multilingual terminology. It describes the various advantages of the tool and shows a process for transforming a traditional terminology list into a list of inflected forms as well as patterns which can be used to find possible morpho-syntactic derivations of terms.

1. INTRODUCTION

This article describes the use and development of the tool for grammar and terminology control FLAG (Alphonse et al., 2002) for automating the process of terminology checking in a major multilingual Swiss organisation. The aim here is not to find new term candidates but rather to show existing terms in new documents to ease the task of term collection. This means showing all the variations of the terms in question in the three national languages, French, German and Italian (see also Jacquemin, 1999). Since FLAG allows a user to find not only strings of characters (i.e. terms) but also to tag and chunk a text and subsequently search for complex patterns, the system was suited very well for this kind of task.

For this application, we have developed additional tools that take as input a list of terms and generate two resources which can be directly used in FLAG: (1) a list of correctly inflected terms, and (2) a set of all the patterns which together allow FLAG to find other structural and derivational variants of the terms. In the following paper, we present FLAG and these two components. Finally we discuss briefly their integration in the terminologist's workflow.

2. FLAG

The FLAG system has been developed in the project of the same name by ISSCO, DFKI (*Deutsches Forschungszentrum für Künstliche Intelligenz*) and its spin-off acrolinx. The main aim was to design a general platform for building language control applications including terminological, stylistic and grammatical aspects.

2.1 Terminology Control

The FLAG component for controlling terminology allows the use of terminology in documents to be checked against termbases. Currently 10 termbases can be used at the same time. As shown in example (1), each entry contains four elements: the term to be recognized, the reference term (which could be e.g. the lemma or the reference form) and the path of the reference files (e.g. the terminological file or the translation).

(1) copie de sauvegarde copie de sauvegarde
flag/fiche02.html flag/trad02.html

The content of the fields is left open to the developer or the user. The structure of these lists is therefore very flexible.

The terms can for example be divided into two categories, one for approved terms and one for non-approved terms. In the case of the non-approved term, the second word corresponds to the correct form (see example 2):

(2) backup copie de sauvegarde
flag/fiche02.html flag/trad02.html

The terms can be imported easily from most of the current terminological databases. It is also possible to extract semi-automatically terms from texts.

2.2 Language and Style Control

The basis for this module is a robust and powerful natural language processing system which uses a morphological analyzer (Mmorph[1], [Petitpierre et Russell, 1995]), a statistical morphosyntactic tagger (TnT[2], [Brants, 1996]) and a phrasal chunker (Chunkie, [Skut et Brants, 1998]). The advantage of this combination lies in the fact that known words are analyzed with their full morphological information while unknown words are assigned the most probable tag, based on a statistical model, trained on large corpora. It is thus possible to identify errors even in words which are not in the lexicon. This point is essential for the processing of technical texts containing a high density of technical terms.

This shallow approach has a number of advantages over classical approaches to grammar control (for example the Caterpillar tool from CMU [Mitamura et al., 2001]). In traditional systems, the input text is analysed and where the analysis fails the system assumes an error has occurred. In the FLAG system on the other hand the text is only analyzed

[1] Mmorph is a two-level morphology which has been developped at ISSCO (see http://www.issco.unige.ch/tools/) .

[2] TnT has been developed at the Univeristy of the Saarland (see http://www.coli.uni-sb.de/~thorsten/tnt/)

for errors which are predefined in the system. This approach makes FLAG much more robust and configurable than analysis-based systems. In particular, it can be used for a number of different tasks, as shown in this paper.
The error description formalism was designed specifically for this purpose. Rules can be written using linguistic objects (described using feature structures) which then be combined to form rules. Rule (3) for example checks subject-verb agreement.

(3)
```
# OBJS
@singSubj ::=      [POS "NN"    numb "s"]
@singVerb ::=      [POS "VVFIN" numb "s"]
@plurVerb ::=      [POS "VVFIN" numb "pl"]
#RULES
Trigger(80) ==     @singSubj^1  []*
@plurVerb^2 →  $singSubj^1,  $plurVerb^2
NegEv(40) ==       $singSubj  []*  @singVerb
[]*  $plurVerb
```

This rule contains different elements :

- **A definition of linguistic objects** (in #OBJS): the rule above, for example, defines three objects: *singular subject* (`singSubj`), *singular verb* (`singVerb`) and *plural verb* (`plurVerb`). These are defined here on the basis of their syntactic category *POS* (`NN=noun, VVFIN=inflected verb`) and their number *numb* (`s(ingular)), pl(ural)`).
- A so-called **trigger rule**: this trigger is designed to identify the range of potential errors. The confidence level here is set to 80, which means that if the rule is fired, the sentence has an 80% chance of containing an error of the type "subject-verb agreement". The left-hand side of the rule is a regular expression over linguistic objects: the rule looks for a match for the definition of "`@singSubj`" followed by any number of words then a corresponding occurance matching "`@plurVerb`". If the rule is fired, the position of the object is saved with a coindexation label (marked by the "hat" symbol "^"). These positions can then be assigned to variables declared on the right-hand side, which are indicated with dollar-signs. (such as "`$singSubj`") and can be used in other rules. The variables are thus an interface between the trigger and confirmation rules.
- **Confirmation rules (optional)** (NegEv or PosEv): the (positive or negative) confirmation rules are used to weaken or strengthen the hypothesis about an error. There is no technical limit to the number of confirmation rules which can be defined. In the above example, the "negative evidence" rule looks for a singular verb between the two objects identified by the trigger rule (`$singSubj, $plurVerb`). The confidence level is set to 40, which means that if this rule fires, it will weaken the confidence level of the trigger. In our example the probability would be reduced to 40% (80-40). The positive evidence rules are used to strengthen a hypothesis, in which case their confidence level is added to the index of the trigger.

Each rule also contains a URL for a help file which the system can show the user on demand. This file contains a short description of the error and some positive and negative examples. This file is stored as XML and can be rendered by an XSL stylesheet for viewing in a browser. FLAG also contains a plug-in for Microsoft Word (acrocheck[3], [Bredenkamp et al., 2002]) which allows style, grammar and terminology control directly during the editing process.
For our purposes, the interest of FLAG lay in the richness of the control mechanism and the flexibility in the way terminology databases and grammar rules could be used. In this project, the main task was to build tools for creating different resources on the basis of the users' terminology lists, which could be then used in the FLAG system. The first of these resources is a list of inflected terms which contains only correctly inflected forms and which is used for terminology control; the second is a set of grammar rules for checking other possible variants of these terms (combinations and structural and derivational variations). We present these tools in the following sections and discuss their integration in the FLAG system.

3. GENERATION OF MORPHOLOGICAL VARIANTS OF TERMS

The aim of this module is to generate the different inflected forms of a list of terms in the three languages. In order to produce only correct forms, we inflect the head of the term and any modifiers, limiting ourselves to plural (for all languages), feminine (for French and Italian) and nominal case terms (for German):

assuré obligatoire (m. s.) (obligatorily insured person)
--> assurée obligatoire (f. s.)
--> assurés obligatoires (m. pl.)
--> assurées obligatoires (f. pl.)

and to the different conjugations of verbs in the case of verbal terms:

avoir la garde d'un enfant (looking after a child)
--> a la garde d'un enfant
--> avaient la garde d'un enfant
--> auront la garde d'un enfant

The module is designed with the following goals: to make use of the simplified structure of terms and to develop a very simple process, basically a set of perl scripts linked together. The generation is performed in five distinct phases: (a) segmentation of terms using "ISEG", the ISSCO segmentizer, (b) morphological analysis with Mmorph (Petitpierre et Russel, 1995), (c) syntactic disambiguation and assignment of one or more syntactic patterns indicating which parts of the term will vary, namely the head and the modifiers, (d) automatic morphological classification of words to be inflected and generation of the inflected forms, both steps being performed by Mmorph and finally, (e) generation of the different inflected forms of the terms in

[3] For further information about acrocheck, see http://www.acrolinx.de/acrocheckOverview_en.html.

the FLAG format. This approach provides a means to reliably generate only correct variations of the terms.
In the table below we show the results of the evaluation for the three languages.

	Terms correctly generated	Terms correctly generated with some overgeneration	Terms not completely generated
FR	83 %	10 %	7 %
IT	89 %	5 %	6 %
GE	94 %	0 %	6 %

Table 1

This table shows that the accuracy is more than 90% in each of the three languages. The second column shows the cases where the algorithm overgenerates. It should be noted that this happens where there are attachment ambiguities (German has no overgeneration at that level, due to the fact that most of the terms are compound nouns). For example for "*demande de référendum valable*" our tool will generate all the possible forms: "*demande de référendum valable", "demandes de référendum valables", "demandes de référendum valable*". The cases of failure in the third column can be explained largely by the absence of the relevant syntactic patterns in the list. If these turn out to be common they can of course be added to the list.
Using simple and language independent methods, we have thus been able to produce the correct inflected forms for the terms and to distinguish them from other potential variations which are detected by the grammar checking component, which we will discuss in the following section.

4. GENERATION OF PATTERNS FOR TERM VARIATION EXTRACTION

For other possible variations of terms, such as synapses (*adaptation des salaires --> adaptation importante des salaires*), structural variations (for example in case of a passive form) and derivational changes (*accepter (V) des titres --> acceptation (N) des titres*), we take advantage of the fact that FLAG provides a robust analysis of the input text to be checked and a powerful mechanism of rules. A Perl program produces from an input terminology list a set of rules in the FLAG format. For example, the rule (1) was generated from the input term "*mise au concours de poste*". Very briefly, it specifies that the program will highlight the lemma of the noun «*poste*» if it is followed, in the same sentence, by «*mise*» and then by «*concours*» at any distance. In this case, the system will produce a message stated in the #HELP section.

(1)
```
#HELP : "This may be a variation of the term: mise au concours de postes"

#OBJS
@noun_mise ::= [MORPH.LEMMA "^mise$" POS "noun"]
@noun_concours::= [MORPH.LEMMA "^concours$" POS "noun" ]
@noun_poste ::= [MORPH.LEMMA "^poste$" POS "noun"]

#RULES Trigger == @noun_poste^1 []* @noun_mise^2 []* @noun_concours^3
-> $noun_poste^1 $noun_mise^2 $noun_concours^3
```

This rule will find, for example, variation of structure as in (2) and (3):

(2) *les postes vacants font l'objet d'une mise au concours publique*.
(3) *Si l'accès à un poste est limité, l'autorité compétente le signale dans la mise au concours*.

To evaluate this module, we manually annotated in a text every occurrence of 50 French terms that should have been found by our rules. We then compared these results with the occurrences automatically extracted by FLAG. Out of 103 occurrences found manually, 86 were found by the system, which represent a recall ratio of 83%. Mistakes come from two main causes: a bug in the segmenting module of FLAG which produce wrong cut in ellipsis article (7 mistakes out of 17), and 4 unknown lemmas in the FLAG tagger TnT (Brants, 1996). Simple corrections of the segmenting module will allows us to reach a recall level of 90%, value that is considered as acceptable by our users.

5. INTEGRATION IN THE TERMINOLOGIST WORKFLOW

FLAG is perfectly integrated in the terminologist's workflow. On the one hand, the terminologist can generate for each new terminology list the resources described below by using a simple script in Perl. On the other hand, FLAG is completely integrated into MS-Word; that means users can check the terminology directly inside a MS-Word document. FLAG is also very flexible and customizable. Different colours can be assigned to each list, so the terminologist can see from which list the highlighted term comes from. For the term extraction variation, the user can choose the types of rule he may want to apply. Finally, any other lists of terms or rules can be easily added, making possible to use the tool for other terminological purposes like: comparing terminology lists, searching for a set of candidate terms, controlling terminology, etc. The screenshots (1) and (2) show some of the functionality.

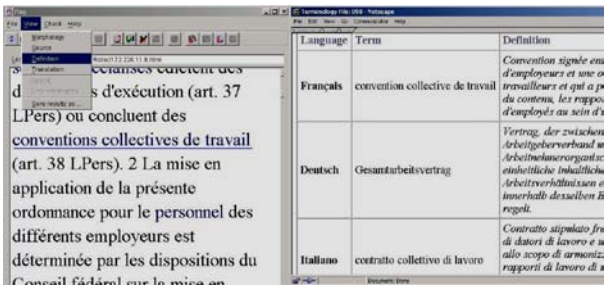

Figure 1

In (1) the user has started a terminology check which uses the termbanks with inflected forms: the inflected term

conventions collectives de travail is shown in the text, showing that it has been found in the termbank. The user can view on demand the information associated with this term (in this case the terminology file from which it was extracted).

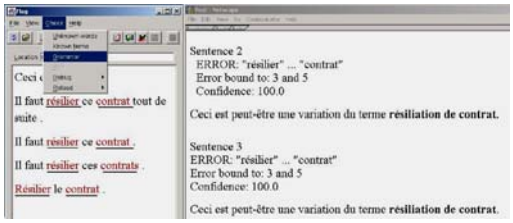

Figure 2

In (2), the user has applied grammar checking (*check grammar* in the drop-down menu). In this artificial example, a single rule has been applied which searches for the verb *résilier* followed by the nominal object *contrat.* This search captures one of the derivational variants of the base term *résiliation de contrat*, such as *résilier ce contrat*, *résilier un contrat*, etc. as shown in the screenshot. It is of course possible to combine these two types of search and mark the terms and their derivations at the same time, but with different colours. This option allows the user to very easily distinguish the correct variations from the ones found as (unwanted) derivations. The tool is currently being tested by our users.

6. Conclusion

In this paper we have tried to show the value of a checking tool based on robust NLP for verifying terminology in several languages. On the one hand, this approach provides a robust analysis of the input text combining linguistic and statistical methods in a language-independent way; on the other hand it provides a powerful rule mechanism for finding particular phenomena in texts; finally it offers a great deal of flexibility for both users and developers to configure the tool to meet their own requirements. The power of the tool is such that it can be used for other task besides terminology checking, such as searching for anglicisms or unwanted expressions, etc.

REFERENCES

- Alphonse et al. (2002) Automatisation de l'activité de vérification terminologique : FLAG, In Proceedings of TIA-2003, Strasbourg, 31 avril et 1er avril 2003.
- Brants T. (1996), TnT – A Statistical Part-of-Speech Tagger, Technical Report, Saarbrücken, Saarland University, Computational Linguistics,
- Jacquemin Chr. et Tzoukermann e. (1999), NPL for term variant extraction, in: T. Strzalkowski, Natural Language Information Retrieval, London, Kluwer
- Petitpierre D. et Russell G. (1995), Mmorph, The Multext Morphology, Technical Report, ISSCO, Genève.
- Becker M., Bredenkamp A.., Crysmann B. et Klein J. (1999), Annotation of Error Types for German USENET News Corpus, Proceedings of the ATALA workshop on Treebanks, Paris.
- Bredenkamp A., Crysmann B. et Petrea M. (1999), Looking for Errors: A declarative formalism for resource-adaptive language checking, Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens.
- Bredenkamp A., Lieske C., Thielen C. et Wells M. (2002), Controlled Authoring at SAP, Proceedings of ASLIB, Translating & the Computer 24, London.
- Daille B., Fabre C. et Sebillot P (2002), Applications of Computational Morphology, in: P. Boucher, Many Morphologies, Somerville, Cascadilla Press.
- Mitamura T., Nyberg E., Baker K., Svoboda D., Torrejon E. et Duggan M. (2002), The KANTOO MT System: Controlled Language Checker and Knowledge Maintenance Tool, NAACL 2001.
- Skut W. et Brants T. (1998), Chunk Tagger - Statistical Recognition of Noun Phrases, Proceedings of the ESSLLI Workshop on Automated Acquisition of Syntax and Parsing, Saarbrücken, Germany.