MULTIDIMENSIONAL FRENCH

Bill Wadge
Computer Science Department
University of Victoria

Christine Wadge
French Department
University of Victoria

Abstract

We describe a multiversion web page (implemented with IHTML) which allows a French language student to browse a fragment of the French language. The sentences of this fragment (there are over four million) illustrate some of the more important aspects of French grammar - including word order, conjugation of verbs, agreement of adjectives, direct and indirect objects, and negation.

The site allows the student to specify the basic vocabulary and other parameters of the sentence, simply by clicking hypertext links in a set of tables. These links take the student to a version of the page that displays the sentence in the fragment determined by the student's choices.

As a result, beginning students with only a basic grasp of grammar and vocabulary can express themselves in sentences which idiomatically correct, and often at a level of sophistication beyond the unaided efforts of students several years' more advanced.

The Learning Atmosphere

Almost everyone agrees that the key to learning a natural language is exposure - to hear and read as many examples of correct utterances. This is how babies learn their native language(s). Other forms of immersion have proved the most effective with older individuals as well.

Currently, however, immersion is simply not available to most students of French. They are learning the language in an environment which, as far as French is concerned, is linguistically impoverished; typically, as one course out of several at an anglophone school or college in an anglophone community.

Technology, of course can, help enrich this environment. In Canada, every community has at least one French language radio and television station. Most schools have language labs that help student's practice reading writing and listening. And now students can find up to date French language newspapers and radio broadcasts (real audio) on the internet, as well as French language newsgroups and chat rooms.

Yet for most students this is not enough. The problem is that in almost all of these contexts the student is a passive listener or reader who has no real control over the discourse. This kind of immersion may not help the students to express themselves, for example, writing a short
elementary essay about a trip to a restaurant. They cannot answer specific questions of the form, "how do you say .... ".

**But grammar, what big tables you have!**

In principle, a student can calculate the correct form of an utterance using the rules of formal grammar. The grammar of French in particular has been studied for hundreds of years and the rules and exceptions are documented in any number of well-written books. Moreover, the rules (from a computational view) are actually fairly simple: most can be formulated on a case-by-case basis, with each case expressed as a relatively small table.

As a result, many upper level language courses are actually grammar courses. Their goal is to upload the formal rules of grammar into the students' brains so that they can generate utterances by mechanical calculation.

The formal grammar approach is ideally suited to modern educational institutions, based as they are on the specialized/individualistic mode. Each student can (supposedly) learn the grammar rules in isolation, without the need to interact much with other students or the instructor.

The formal grammar approach is a typical application of the philosophy of logical atomism[1], according to which knowledge is built up from small, simple facts which whose interpretation requires no context. In the atomistic approach to language/grammar learning, the entire grammar is first divided up into one or two dozen topics to be studied, as far as possible, in isolation. Sentences are broken down into words (with associated functions) which in turn are studied in isolation. Finally the individual words are broken down into their separate forms, so that a typical atomic fact would be that the first person plural imperfect form of *apprendre* is *apprentions*.

Logical atomism is also well suited to automation. It is relatively straightforward to produce generic 'educational' software which takes (or generates) a collection of atomic facts and uses them as the basis for various kinds of exercises.

Unfortunately, humans do not work like computers and cannot explicitly calculate grammar rules reliably and in real time. The different rules can interact in complicated ways, and often the simplest, most common expressions require the more complex and obscure rules (the exceptions). The formalistic approach may teach the students some grammar but it doesn't necessarily teach them how to speak and understand.

The failure of the grammar approach is often covered up by the nature of the assessment: the students are tested on their knowledge of grammar, not on their practical competency in the language. The testing can also be easily automated, since it involves requiring the student to reproduce a discrete selected set of atomic rules or facts. In fact most software based on the atomistic approach makes no distinction between learning and testing.
Most language educators are well aware of the limitations of the formal grammar approach but many feel they have no choice because they lack the resources for the enriched environment required by the alternate approach. Furthermore, the formalist approach seems also to be the modern, high-tech, information age approach, because almost all current educational software is based on it.

**Swapping roles with the computer**

The computer is supposed to be the student's assistant but for many language students, the experience is exactly the reverse. A student being drilled by conventional software feels that the computer is the master and that the machine understands French better than the student ever will.

In fact computers (even with sophisticated software) are poor at linguistic competence measured in any practical way. They are, however, logical atomists par excellence. They can store huge numbers of atomic grammar rules and facts and apply them almost instantaneously. The problem is how to use this ability to help the student gain linguistic competence - a skill that resists a hierarchical, atomistic analysis.

The obvious approach is to use the computer to generate grammatically correct sentences for the student. The problem, however, is to provide the student with input or control over the sentences generated; ideally, control over their semantics, so that the students can express themselves in French. Control presupposes input; how can the students tell the computer what to say (in French) without saying it already in French, or in their own language.

Stated in this general form, the problem is currently unsolvable. Natural language semantics itself appears immune to an atomistic analysis, and so at present there is no general translation algorithm. For the same reason, there is no general formal logical system for expressing meanings (whose expressions could then be translated to French).

**Multidimensional Fragments**

In this situation (common in AI), there are two possibilities: to develop general algorithms (heuristics) which always given an answer but not necessarily a correct one; or restrict the problem to one for which there exists an algorithm which always gives a correct answer.

Heuristics can be useful, but for educational software they may not be appropriate, because we want the student to have complete confidence in the information presented to them.

Instead, we make the problem algorithmically solvable by restricting it: by attempting to generate only sentences in a (relatively) small and well-defined fragment of French. In particular, we choose fragments that can be indexed using a simple multidimensional scheme. If we choose the dimensions to be grammatically or semantically meaningful, the student can specify an individual sentence in the fragment without being able to generate it.
It seems that French (and, as far as we know, all natural languages) lend themselves quite easily to multidimensional fragmentation. As a simple example (in English), consider the following sentence:

*The student reads the book.*

We can use this sentence as the basis for a fragment by identifying parts that can be varied more or less independently. For example, we can allow choices other than *the student* for the subject; verbs other than *to give*; and direct objects other than *the book*.

If we restrict the choice of alternatives to just a few in each dimension, we can present them in tabular form:

<table>
<thead>
<tr>
<th>Subject</th>
<th>Verb</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>student</td>
<td>read</td>
<td>book</td>
</tr>
<tr>
<td>teacher</td>
<td>write</td>
<td>exam</td>
</tr>
</tbody>
</table>

Then any combination of choices, one per column, gives us a sentence in the fragment (there are eight different sentences). Furthermore, each such sentence can be easily constructed from its parameters: it consists of the word *the*, the subject word, the verb word with an *s* on the end, the word *the*, and the object word.

The situation becomes far more interesting if we add three more dimensions, Number (allowing plural as well as singular subjects), Form (allowing negative as well as affirmative sentences) and Tense (allowing simple past, as well as present, tense).

<table>
<thead>
<tr>
<th>Subject</th>
<th>Verb</th>
<th>Object</th>
<th>Number</th>
<th>Form</th>
<th>Tense</th>
</tr>
</thead>
<tbody>
<tr>
<td>student</td>
<td>read</td>
<td>book</td>
<td>singular</td>
<td>aff.</td>
<td>present</td>
</tr>
<tr>
<td>teacher</td>
<td>write</td>
<td>exam</td>
<td>plural</td>
<td>neg.</td>
<td>past</td>
</tr>
</tbody>
</table>

For example, given the following choices
The choices are laid out in a condensed table with the following format
(Choices are listed horizontally).

<table>
<thead>
<tr>
<th>SUJET</th>
<th>nom</th>
<th>pronom</th>
<th>fille</th>
<th>garçon</th>
<th>professeur</th>
<th>étudiant</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>singulier</td>
<td>pluriel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADJECTIF</td>
<td></td>
<td></td>
<td>jeune</td>
<td>vieux</td>
<td>canadien</td>
<td>français</td>
</tr>
<tr>
<td>VERBE</td>
<td>affirmatif</td>
<td>négatif</td>
<td>donner</td>
<td>rendre</td>
<td>envoyer</td>
<td>apporter</td>
</tr>
<tr>
<td>ADVERBE</td>
<td></td>
<td></td>
<td>souvent</td>
<td>toujours</td>
<td>vite</td>
<td>lentement</td>
</tr>
<tr>
<td>OBJET DIRECT</td>
<td>nom</td>
<td>pronom</td>
<td>livre</td>
<td>papier</td>
<td>examen</td>
<td>lettre</td>
</tr>
<tr>
<td></td>
<td>singulier</td>
<td>pluriel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADJECTIF</td>
<td></td>
<td></td>
<td>lourd</td>
<td>petit</td>
<td>important</td>
<td>vieux</td>
</tr>
<tr>
<td>OBJET INDIRECT</td>
<td>nom</td>
<td>pronom</td>
<td>dame</td>
<td>homme</td>
<td>professeur</td>
<td>étudiant</td>
</tr>
<tr>
<td></td>
<td>singulier</td>
<td>pluriel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADJECTIF</td>
<td></td>
<td></td>
<td>grand</td>
<td>petit</td>
<td>sage</td>
<td>ancien</td>
</tr>
</tbody>
</table>

The selected choices are underlined here; on the web, they are highlighted using
different cell and text background colors.

The cells containing unselected choices are links; clicking the link takes the user to a new
version of the page in which the new choice is selected but the other choices are unchanged.

The sentence defined by the selected choices is displayed at the bottom of the page; it
changes every time the user changes the value selected in one of the dimensions. For example, the
choices underlined above correspond to the sentence

La fille canadienne ne donne pas toujours les petits livres a l’étudiant.

(The Canadian girl doesn’t always give the student the little books).

Here are just some of the grammatical facts and rules required to generate this sentence:

- In the absence of pronouns, the word order for a declarative sentence
  is subject-verb-direct object - indirect object
- The noun fille is feminine
- La is the feminine singular form of the definite article
- The adjective canadien is regular in that it is placed after the noun it modifies.
- The feminine singular of adjectives ending in ien ends in ienne.
- A verb is negated by placing ne directly before and pas directly after
  (in the is subject of pronouns)
- Donner is a regular first conjugation verb, and therefore its third person
  singular form is donne.
- Adverbs like toujours come after the verb, and after pas if it is present
- Livre is masculine
- Livre is regular in that its plural is livres.
- Les is the plural form of the definite article
<table>
<thead>
<tr>
<th>Subject</th>
<th>Verb</th>
<th>Object</th>
<th>Number</th>
<th>Form</th>
<th>Tense</th>
</tr>
</thead>
<tbody>
<tr>
<td>student</td>
<td>read</td>
<td>book</td>
<td>singular</td>
<td>aff.</td>
<td>present</td>
</tr>
<tr>
<td>teacher</td>
<td>write</td>
<td>exam</td>
<td>plural</td>
<td>neg.</td>
<td>past</td>
</tr>
</tbody>
</table>

The resulting sentence is

*The students did not write the book.*

The fragment is not only larger (64 sentences), its sentences illustrate many more grammatical rules. For example, here are some of the basic facts needed to produce the sentence just quoted:

- The past tense word order is subject-verb-object
- *The* is invariable for number
- The plural of *student* is *students*
- The simple past is formed using the past of *to do* and the infinitive
- The past plural of *to do* is *did* (irregular)
- The past negative is formed by placing "not" before the infinitive

Clearly the tables could be much longer, allowing more choices for the in some of the dimensions (for example, exercise for the Object or future for the Tense). We can also add more dimensions, for example, for adverbs or subject and object adjectives. As we show below, even tables that are only slightly larger can produce huge numbers of sentences many of are beyond the learner's competence.

**The French language Browser**

We have implemented a browsergenerator for a fragment of French big enough to be useful for first and second year students. It allows students to generate sentences involving four different verbs that expect both direct and indirect object. The subjects and both objects can each be singular or plural, nouns (with an optional adjective) or pronouns.

There are a total of 12 user-settable dimensions: three for the subject (subject itself, subject number, subject adjective), three analogous dimensions for the direct and indirect objects, and a verb, adverb, and form dimension.

The generator was implemented in IHTML and is accessible as a multi-version web page, currently at

http://lucy.uvic.ca/~wwadge/intra32/home.html
- *petit* is regular in that its masculine plural form is *petits*.
- *petit* is irregular in that it is placed before the noun it modifies
- when the indirect object is a noun phrase, it is preceded by a
- the article *le* (but not the pronoun *le*) contracts to *l ’* before a vowel
- this last rule has precedence over the rule that a followed by *le* contracts to au

Many of the rules used to generate this and other sentences in the fragment involve topics typically not discussed (and certainly not mastered) until well into the second year of university level French.

The IHTML Implementation

The generator described above was implemented using IHTML 2 [2]. The source is structured as 30 (multiversion) include files, with about twice as many actual files (IHTML allows multiple sources for a single include file). There are slightly more than 1000 total lines of IHTML source. (Note that there are more than four million different versions of the page, each displaying a different sentence).

These lines are “pure” IHTML - there are no calls on shell scripts. However, we needed 30 lines of Javascript to do the final processing of the displayed sentence, mainly to implement the contraction rules that are orthographic rather than grammatical.

The choice tables were implemented using transversion (vmod) links - each cell containing an unselected choice is a link to a new version of the page in which the coordinate in one dimension is altered to record the choice made.

The display sentence is computed using case (iselect) statements and versioned includes. Several include files (for example, the one for noun phrases) are generic - are used to generate the subject and both object phrases. These generic files have extra dimensions that are used as formal parameters by the code invoking them. In several places we used a series of include files, each include calculating the values for new dimensions from those already available.

Here, for example, is the topmost source for the sentence itself:

```
<!--#include virtual=subj.html -->
<!--#include virtual=ne.html -->
<!--#include virtual=proi2.html -->
<!--#include virtual=prod.html -->
<!--#include virtual=proi3.html -->
<!--#include virtual=vphr.html -->
<!--#include virtual=nmod.html -->
<!--#include virtual=nnoi.html -->
```

This file encapsulates the knowledge of the word order (vphr stands for verb phrase, prod for direct object pronoun, proi2 for first or second direct object pronoun, and so on. The context consists of the user defined dimension values and the values of other dimensions computed from
them. Since IHTML is an intensional language, the context is implicit and the names of the
dimensions do not appear if the values are not directly required.

Notice that in every context some of these include files will be empty. For example, no
sentence will have both a direct object pronoun and a direct object noun phrase. Intentionality
allows us to postpone decisions based on the presence or absence of these parts - there is only
one version of the source for the sentence.

Here is the code that analyzes the verb infinitive into stem and root prior to producing the
correct form:

    <iselect>
      <icase version="vb:donner">
        <!--#include virtual=conjb.html
        vmod="vbr:donn,vbe:er" --></icase>
      
      <icase version="vb:apporter">
        <!--#include virtual=conjb.html
        vmod="vbr:apport,vbe:er" --></icase>
      
      <icase version="vb:rendre">
        <!--#include virtual=conjb.html
        vmod="vbr:rend,vbe:re" --></icase>
      
      <icase version="vb:envoyer">
        <!--#include virtual=conjb.html
        vmod="vbr:envo,vbe:yer" --></icase>
    </iselect>

Future Development

The one generator described covers some but by now means all of the major grammatical
topics of French. Missing, for example, are tenses, prepositional phrases, pronominal verbs,
subordinate clauses, and so on.

One way to proceed is to produce more and more complex generators that incorporate more
and more grammar and vocabulary. At the moment, however, this strategy is impractical. More
features means more dimensions and more choices and the user interface (not to mention the
implementation) could become bewilderingly complex. Also, more choices mean more
possibilities for nonsensical combinations - such as “the house reads the car”. These sentences
may be grammatically correct but they are confusing to the learner and must be avoided.

An alternative approach is to develop a whole family of more modest generators, each
covering a fragment of the grammar. Most textbooks already divide the grammar up into chapters,
and the suite of generators could be designed to coordinate with the textbook.
The generators could also be used as composition aids by people who already know some French and need to express themselves but don’t necessarily want to deepen their grammatical knowledge (for example, to write business letters).

These people clearly might prefer an English-language interface. Even better, for them we could incorporate an English generator - one that calculates an English sentence from the user-selected parameters. In this form, the generator could be used by people with no knowledge of French. They can make choices until the English sentence expresses the desired thought, then read off the corresponding equivalent (and grammatically impeccable) French version.

References


Acknowledgement

This work was supported in part by a grant from the Canadian Natural Sciences and Engineering Research Council (NSERC).