Processing Mutations in Breton with Finite-State Transducers

Thierry Poibeau
Laboratoire LATTICE (UMR8094)
CNRS & Ecole normale supérieure & Université Paris 3 Sorbonne Nouvelle
1 rue Maurice Arnoux 92120 Montrouge France
thierry.poibeau@ens.fr

Abstract

One characteristic feature of Celtic languages is mutation, i.e. the fact that the initial consonant of words may change according to the context. We provide a quick description of this linguistic phenomenon for Breton along with a formalization using finite state transducers. This approach allows an exact and compact description of mutations. The result can be used in various contexts, especially for spell checking and language teaching.

1 Introduction

Celtic languages (Welsh, Cornish, Irish, Scottish-Gaelic, Manx, etc.) are known to support a common feature: the initial consonant of different types of words (esp. nouns, adjectives and verbs) is modified in certain contexts and after certain function words (e.g. prepositions and determiners for nouns and adjectives; auxiliaries for verbs). This phenomenon known as "mutation" has been largely studied and described from a linguistic point of view. Formal descriptions have even been proposed, especially Mittendorf and Sadler (2006) for Welsh.

In this paper, we investigate mutations in Breton.[1] Our study is largely inspired by the previous study by Mittendorf and Sadler for Welsh: We share with these authors the idea that "initial mutation is close to inflection in nature and is essentially a morphosyntactic phenomenon". We propose to process this phenomenon with finite state transducers. In fact, we propose two formalizations: in the first one, mutations are processed by directly storing the lexical forms with mutations in a dictionary of inflected forms; in the second one, local rules encoded using finite state transducers are applied dynamically, depending on the context. We show that this strategy allows for an exact and compact description of the phenomenon, since transducers directly encode grammar rules.

The paper is organized as follows: we first propose a linguistic description of this phenomenon. We then explore the two strategies exposed in the previous paragraph: a dictionary of inflected form vs local grammars encoded using finite state machines. We conclude with a discussion and an overview of the practical use of this implementation.

1.1 A Quick Description of Mutations in Breton

As said in Wikipedia (http://en.wikipedia.org/wiki/Breton_mutations), "Breton is characterized by initial consonant mutations, which are changes to the initial sound of a word caused by certain syntactic or morphological environments. In addition Breton, like French, has a number of purely phonological sandhi features caused when certain sounds come into contact with others." The following details are then added: "the mutations are divided into four main groups, according to the changes they cause: soft mutation (in Breton: *kemmadurioù dre vlotaat*), hard mutation (*kemmadurioù dre galetaat*), spirant mutation (*kemmadurioù c'hwezhadenniñ*) and mixed mutation (*kemmadurioù mesket*). There are also a number of defective (or incomplete) mutations which affect only certain words or certain letters." A

[1] Breton is a Celtic language spoken in Brittany (Western France) According to recent studies, 200,000 persons understand the language but only 35,000 practice it on a daily basis.

Proceedings of the First Celtic Language Technology Workshop, pages 28–32,
Dublin, Ireland, August 23 2014.

same word can thus appear differently depending on these classes of changes (for example the noun *tad – father* – becomes *da dad – your father; he zad – her father*; etc. because of the possessive pronouns *da* and *he* that entail different kinds of mutation).

The best approach to give an idea of mutations is to consider some examples. "Soft mutations" refer to the fact that after the definite article *ar* (and its variant *an*) or the indefinite article *ur* (or *un*) the initial consonant of singular feminine nouns is subject to the following changes:

- K → G, ex. Kador (*chair*) → Ur gador
- T → D, ex. Taol (*table*) → Un daol
- P → B, ex. Paner (*basket*) → Ur baner
- G → C'H, ex. Gavr (*goat*) → Ur c'havr
- GW → W, ex. Gwern (*mast*) → Ur wern

Note that in Breton nouns referring to objects and abstract notions can be either masculine or feminine (there is no neuter case).

Although the phenomenon is well known, its description is not straightforward since it involves a large number of parameters and different types of information (lexical, morphological, semantic). For example, plural masculine nouns referring to male persons have the same behavior as singular feminine nouns (but this is only true for plural masculine nouns referring to people, not for all plural nouns). It is therefore necessary to distinguish different categories of nouns.

- K → G, ex. Kigerien (*butchers*) → Ar gigerien
- T → D, ex. Tud (*people*) → An dud
- P → B, ex. Pesketaerien (*fishermen*) → Ar besketaerien
- G → C'H, ex. Gellaoued (*French*) → Ar C'hallaoued
- GW → W, ex. Gwerzherien (*sellers*) → Ar werzherien

These mutations also affect adjectives, provided that the noun preceding the adjective ends with a vowel or with the consonant l, m, n, or r.

- K → G, ex. Kaer (*nice*) → Ur gador gaer (a nice chair)
- T → D, ex. Tev (*thick*) → Ur wern dev (a thick mast)
- P → B, ex. Paour (*poor*) → Ur vamm baour (a poor mother)

There are different series of mutations depending on the functional word preceding the noun (and the adjectives if any). It is one of the main difficulties of the language since this phenomenon changes the initial of the words: after mutation, words cannot be found anymore directly in the dictionary.

A comprehensive description of this phenomenon can be found in traditional grammars of the language: see especially Kervella (1976), Hemon (1984) and Stump (1984) for a formal description of agreement in Breton.

1.2 Automatic Processing of Mutations in Breton

Two approaches are possible:

- store all the inflected lexical forms and their mutations in a dictionary. Mutation is then considered as a case of word variation (like the alternation singular/plural);
- compute on the fly the lexical form in context, which is an interesting strategy for text generation or, more directly, in the context of text authoring (for example to assist students producing texts in Breton).

In this paper, we consider both approaches since they are both relevant depending on the context.

2 Two Competing / Complementary Solutions for Mutations in Breton

The following section describes two ways of processing mutations in Breton. We discuss their interest and their applicability to the problem.

2.1 A Comprehensive Dictionary of Inflected Forms

This solution is the simplest one: all inflected forms including those with modified initial letters are included in the dictionary. The dictionary remains manageable and ambiguity introduced by the new lexical forms is limited. Below is a short extract of a dictionary of inflected forms including lexical forms after mutation:

```
kador,kador.N:fs          taol,taol.N:fs
gador,kador.N:fs          daol,taol.N:fs
c'hador,kador.N:fs        zaol,taol.N:fs
```

The format of the dictionary is the one defined by LADL (Courtois and Silberztein, 1990): inflected forms are followed by a lemma (separated by a comma). The category of the word can then be found (N for noun) as well as morphosyntactic features (fs: feminine singular).

However, this solution is not fully satisfactory since it does not explain why a given form is used in a given context. It would be relevant to provide a more dynamic description of the process taking into account the different constraints we have seen in the previous section.

2.2 A Finite State Approach

We have seen in the introduction that mutations refer to a simple change in the first letter of certain words in certain contexts. This phenomenon is essentially local (it does not require to take into account a large context) so finite state transducers seem to be a relevant choice. These transducers will directly encode the rules described in the grammar of Breton that just need to be made more formal.

Below (Figure 1) is an example of such a finite state transducer.

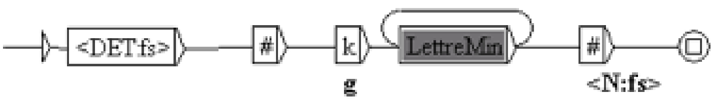

Figure 1: Graph MUT-Detfs-K-G

This graph directly encodes all the constraints involved in the process. Elements that appear in boxes describe a linguistic sequence while elements appearing under the boxes correspond to the rewriting part of the transducer (i.e. the transduction). Here is a description of the different elements that can be used for the linguistic description:

- Tags between < and > refer to morphosyntactic categories (DET for determiner, N for noun, A for adjective, etc.);
- The elements after the colon are morphological features (f: feminine , s: singular...);
- The # sign indicates a separator between words (e.g. blank spaces between words);
- A gray box refers to a subgraph (here LettreMin refers to all lowercase letters; please note that the sequence described here corresponds to any sequence of letters between separators, i.e. tokens, because of the recursion on the state itself);
- Other items appearing in a box correspond to characters (or lexical forms);

Here, we see clearly that K becomes G if the sequence is a fem. sing. noun appearing immediately after a determiner. Notations correspond to the ones defined by the LADL team, see Silberztein (1993) and Paumier (2011) – other frameworks could of course be used like the Xerox FST toolbox (Beesley and Karttunen, 2003).

Transducers provide a unified view of the different contraints along with a rewriting process. Recursive transducers (RTN) make it possible to obtain a concise and effective formalization. Different linguistic phenomena can be processed using a cascade of automata applied one after the other. For example, it seems relevant to re-use the graph encoding noun mutations to process adjectives. If all the mutations for nouns have been encoded and compiled in a single graph called MUT, it is then possible to write the fllowing transducer (figure 2) for adjectives.

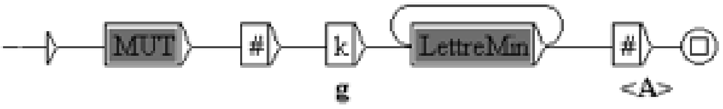

Figure 2: Graph MUT-Adj-K-G

MUT also encodes the constraints on the last vowel of the previous word (only adjectives following a noun ending with a vowel or with l, m, n or r are subject to this mutation).

2.3 Implementation and evaluation

Local contextual grammars can be encoded using various techniques but finite state transducers seem to be the most effective and readable way to encode these rules. This is in line with previous work: for example Mittendorf and Sadler (2006) use the Xerox finite state transducer toolbox to implement mutations in Welsh. Our proposal is very close to theirs.

Various other platforms allow the manipulation of finite state transducers for local grammars. Scripting languages (like perl or python) also offer a good solution but these languages are made to manipulate strings. However for mutations we need to have different information on the words themselves, hence using a linguistic toolbox seems more appropriate.

The implementation of this linguistic phenomenon using finite state transducers produce a compact and accurate description. Grammars are easy to modify and maintain. Additionally different grammars could be developed to take into account local variations and dialects.

3 Discussion

We have presented a practical approach to process mutations in breton. The approach is based on well known techniques (finite state transducers) that provide an accurate and efficient description of the phenomenon. The technique used reveal the fact that mutation is essentially a morphosyntactic phenomenon, as said in the introduction.

However, the main challenge does not lie in the proposed formalization. Endangered languages are generally not well supported (lack of resources and automatic tools) and we think this kind of contribution could have a positive impact on the evolution of the language. If relevant tools exist, it could be a way to attract new attention and help language students acquire a good command of the language. Since a large part of language learners study at home, having dynamic tools assisting text production would be a real plus.

Adding explanation to the above rules would make it possible to generate suggestions during text production or text revision. From this perspective, the description we provide could serve as a basis for a spell checker of the language.[2] Detailed explanations would make the whole system usable for assisting people during language learning (e.g. to explain why a given sequence in not fully correct in case a word should appear with a mutation, etc.). This strategy could easily be re-used for other languages and other linguistic phenomena.

[2] Note that different initiatives exist to develop natural language tools for processing Breton. We should cite more specifically the association Drouizig http://www.drouizig.org/ that has developed a spell checker independently of this study.

References

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford.

Blandine Courtois and Max Silberztein, editors. 1990. *Dictionnaires lectroniques du français*, volume 87, Paris. Larousse.

Roparz Hemon. 1984. *Grammaire bretonne*. Al Liamm, Brest.

Fransez Kervella. 1976. *Yezhadur bras ar brezhoneg*. Al Liamm, Brest.

Ingo Mittendorf and Louisa Sadler. 2006. A treatment of welsh initial mutations. In *Proceedings of the LFG06 Conference*, Universität Konstanz. CSLI.

Sébastien Paumier. 2011. *Unitex 3.0 User Manual*. Universit de Marne la Vallée, Marne la Vallée.

Max Silberztein. 1993. *Dictionnaires lectroniques et analyse automatique de textes : le systme INTEX*. Masson, Paris.

Gregory Stump. 1984. Agreement vs. incorporation in breton. *Natural Language and Linguistic Theory*, 2:289–348.