Generation of exercises for derivational morphology using the Démonette database

Nabil Hathout, Fiammetta Namer, Olivier Bonami, Georgette Dal, Stéphanie Lignon

Received: March 2023 / Accepted: June 2023
Published on line: December 2023

Generation of exercises for derivational morphology using the Démonette database

Démonette-2, une base de données dérivationnelle du français à large couverture lexicale munie de descriptions morphologiques détaillées

Nabil Hathout
CLLE – Université de Toulouse
nabil.hathout@univ-tlse2.fr

Fiammetta Namer
ATILF – Université de Lorraine
fiammetta.namer@univ-lorraine.fr

Olivier Bonami
LLF CNRS – Université Paris-Cité
olivier.bonami@linguist.univ-paris-diderot.fr

Georgette Dal
STL –Université de Lille
georgette.dal@univ-lille.fr

Stéphanie Lignon
ATILF – Université de Lorraine
stephanie.lignon@univ-lorraine.fr

Résumé

Cet article présente un démonstrateur qui permet de créer facilement des énoncés d'exercices en très grand nombre à partir des données présentes dans la base de données Démonette. Dans le cas présent, ces exercices s'adressent plus spécifiquement aux étudiants de licence. Les énoncés peuvent contenir trois types de questions : sur des lexèmes (par exemple, des dérivés), sur une famille de mots dont il faut décrire les relations, ou encore sur des items provenant de différentes familles, à classer relativement à une ou des propriétés. Ces propriétés peuvent être phonémiques, graphémiques, catégorielles, morphologiques ou sémantiques.

Mots-clés : dérivation, base de données morphologique du français, exercices pour étudiants de Licence, génération automatique.

Abstract

This article presents a demonstrator that lets teachers easily create large numbers of exercises from data in the Démonette database. These exercises are intended for undergraduate students. They may include three types of questions: on individual lexemes (e.g., derived words), on sets of words from the same family whose relations need to be described, or on items coming from different families that need to be classified with respect to one or more properties. These properties may be phonemic, graphemic, categorical, morphological or semantic.

Keywords: derivation, morphological database for French, exercises for undergraduate students, automatic generation.

Funding: The Demonext project has been funded by the French National Research Agency (ANR), under the reference ANR-17-CE23-0005. The description of the project, its results and publications are available at https://www.demonext.xyz/

1. Introduction

Online courses and assessments have become familiar to both teachers and students. This trend has accelerated greatly since the health crisis of 2020. One of the advantages of online courses is the possibility of generating content automatically. In particular, a large number of exercises can be created on a particular question, provided that sufficient suitable examples are available. This article presents a "proof of concept" (relatively rudimentary and to be adapted for use by the target audience of this work) that illustrates one of the possible uses of the Démonette database, namely the generation of large numbers of derivational morphology exercises. The database is large enough to provide a sufficient number of examples. In addition, the quality and detail of its annotations enable the production of exercises to train students on a wide range of phenomena, and the creation of tests to assess students on a large range of skills.

Users of the tool we are proposing are mainly teachers of derivational morphology who have a stock of exercises that they propose as part of their teaching and who wish to "massify" them by producing from each one a large number of instances obtained by changing the examples presented in the statement and the data to be processed. This work is of interest to teachers who want to transform classroom content into distance learning. We have designed the demonstrator considering the practices of several linguistics teachers. However, nothing would prevent the production of similar exercises for other audiences, such as students learning French as a foreign language.

Automatic exercise generation has been the subject of numerous studies, including (Wilson, 2014; Baptista et al., 2016; Chinkina & Meurers, 2017; Kledilson et al., 2018; Agirrezabal et

al., 2019; Bodnar, 2022; Heck & Meurers, 2022). The main contribution of the demonstrator we present here is to illustrate the quality of the Démonette database and to give an idea of the range of exercises it can produce. In addition, some technical choices made in this work are relevant to other linguistic exercise generation projects. Overall, the demonstrator uses annotations described in tabular form in a database. The sample exercises illustrate possible uses of these annotations. All the exercises presented relate to derivational morphology, but it would be possible to adapt them to other phenomena in inflectional morphology, phonology, lexical semantics, mono- or bilingual lexicography, etc. by using different resources.

From a technical point of view, our demonstrator was implemented in python in the form of a Jupyter notebook. This format is well suited for prototyping. The interpreted nature of python makes it easy to test the individual parts of the program. The notebook format also makes it easy to modify, debug, and rerun the program. Python provides many libraries, including the pandas data analysis library. Another technical choice we have made is to use LaTeX. LaTeX is a programming language that produces PDF (or PostScript or HTML) documents from a program. In this way, one can produce a LaTeX document using a program and then compile it in the python script to obtain a PDF document. LaTeX provides a rich and comprehensive font environment that includes numerous Latin and non-Latin fonts, in particular fonts with all IPA phonetic symbols and diacritics. It also allows the creation and inclusion of diagrams and figures.

2. A first example

Before going into more detail about how the exercise generator works and the range of questions and phenomena that can be addressed using the Démonette database, we present an initial example of an exercise that will help us understand the purpose of this work. A graphic illustration of the general mechanism is provided in Figure 3 at the end of the section.

The exercise consists in identifying the similarities and differences between six lexemes (i.e., six lexical units) whose citation form ends in *aire* but are morphologically heterogeneous (i.e., created using different Word Formation processes). More specifically, the exercise involves observing that some lexemes in *aire* are adjectives (*légendaire* ‘legendary’), nouns (*commentaire* ‘comment’), or both (*signataire* ‘signatory’), that some are verb-based (*commenter* ‘to comment’→ *commentaire*, *signer* ‘to sign’ → *signataire*), others noun-based (*aéroport*$_{N}$ ‘airport’→ *aéroportuaire*$_{A}$ ‘airport’), and determining the form and meaning relations they have with their possible bases.

For example, *commentaire* refers to an act of communication or its concrete result, whereas *signataire* refers to a person. We also note that, formally, *commentaire* is formed by simply adding *-aire* to the imperfect stem (*comment-*) of *commenter*, whereas the *-at-* sequence in *signataire* belongs neither to the inflectional stem of the verb base nor to the suffix exponent,

and that *signat-* can therefore be analyzed as a suppletive learned root inherited from the supine theme of the Latin verb *signare* (on the distinction between base, stem, root and theme, see Roché, 2010; for a discussion of learned verb roots, see Bonami, Boyé & Kerleroux, 2009). Other lexemes in the list are morphologically simple verbs where *aire* is not an inflection exponent (*plaire* 'like$_{INF}$' *vs plait* 'like$_{PRS.3SG}$'*; extraire* 'extract$_{INF}$' *vs extrait* 'extract$_{PST.PTCP}$'). The resulting statement can take the following form:

One process or several processes?
The following 6 lexemes all end with the graphic sequence *aire*.
Propose a derivational analysis for each of them. In particular, answer the following questions:
- Are they all formed by derivation?
- Are they derived by the same process?

aéroportuaire, commentaire, dentaire, déplaire, signataire, taire
'airport', 'comment', 'dental', 'to not like', 'signatory', 'to keep quiet'

Figure 1. Example of exercise statement.

A variety of other examples can be substituted for the ones given at the end of this exercise, such as:

- *contestataire, distraire, embryonnaire, parasitaire, parfaire, protestataire*
  'anti-establishment', 'to distract', 'embryonic', 'parasitic', 'to perfect', 'protestor'
- *commanditaire, exemplaire, génocidaire, plaire, signataire, soustraire*
  'silent partner', 'copy', 'genocidal', 'to please', 'signatory', 'to substract'
- etc.

The idea is therefore to have, on the one hand, a gap-filling text to be filled in with a list of examples, and, on the other hand, a set of sextuplets of examples. Each sextuplet consists of three pairs of lexemes ending in *aire*: two nouns or adjectives derived from verbs; two adjectives derived from nouns; two verbs whose infinitive form ends in *aire*. The lexemes in these three pairs are randomly selected from three sets extracted from Démonette's tables (for a presentation of Démonette's tables, see Namer et al's general article in this volume).

1/ Verb-based nouns and adjectives derived with the *-aire* suffix. Démonette contains 79 such lexemes, among which:

commentaire 'commentary', *contestataire* 'anti-establishment', *contresignataire* 'counter-signatory', *fermentaire* 'relative to fermentation', *stipendiaire* 'who is on someone's payroll', *revendicataire* 'demanding', *dépositaire* 'keeper', *protestataire*, 'protestor', *rétributaire* 'relative to

remuneration', *retardataire* 'latecomer', *adjudicataire* 'successful bidder', *baptistaire* 'who records a baptism', *manifestaire* 'relative to a manifesto', *gesticulaire* 'relative to gesticulation', *narrataire* 'reader addressed by the narrator', *cosignataire* 'cosignatory', *intercalaire* 'divider', *distributaire* 'who receives a share in a distribution'

2/ Adjectives derived from nouns by suffixation in *-aire*. Démonette contains 634 adjectives of this type, among which:

lagonaire 'relative to lagoon', *bolaire* 'of a clayey nature', *tumultuaire* 'tumultuous', *plébiscitaire* 'relative to referendum', *villositaire* 'who suffers from atrophy of the vilosities', *myofibrillaire* 'who suffers from myofibril degradation', *oviductaire* 'relative to the oviduct', *sédimentaire* 'sedimentary', *obituaire* 'obituary', *aréolaire* 'relative to the areola', *utilitaire* 'utilitary', *subventionnaire* 'who has to pay subsidies', *transactionnaire* 'specialist in pharmacy transactions', *fracturaire* 'fracture-related', *ostensionnaire* 'who participates in ostensions (religious ceremonies)', *rubanaire* 'shaped like a ribbon', *simplicitaire* 'proponent of voluntary simplicity', *cémentaire* 'relative to the cementum', *fusionnaire* 'relative to merging', *vestibulaire* 'vestibular'

3/ Verbs with a lemma ending in *aire*. Démonette contains 63 verbs of this type, among which:

refaire 'to redo', *déplaire* 'to not like', *méplaire* 'to not like'*, resatisfaire 'to satisfy again', reparfaire 'to perfect again', raire* 'to roar', *abstraire* 'to abstract', *autosoustraire* 'to remove oneself', *plaire* 'to please', *mésatisfaire* 'to displease', *taire* 'to keep quiet', *entre-bienfaire* 'to do each other good', *liquéfaire* 'to liquify', *redistraire* 'to distract again', *parfaire* 'to perfect', *malfaire* 'to do ill', *redéplaire* 'to displease', *distraire* 'to distract'

Creating an exercise instance therefore involves selecting three lexeme pairs from the three sets and including them in the gap-filling text in place of the <EXAMPLES> tag (Figure 2). Creating a text with holes is not particularly difficult in LaTeX. All you have to do is include a variable in the text and assign it a string corresponding to the examples you want to include in the statement. This part of the task is made all the simpler by the fact that it can be carried out by adapting an existing statement.

One process or several processes?
The following 6 lexemes all end with the graphic sequence *aire*.
Suggest a derivational analysis for each of them. In particular, answer the following questions:
• Are they all formed by derivation?
• Are they derived by the same process?

< EXAMPLES >

Figure 2: Template used to generate the exercise shown in Figure 1.

Setting up the three sets of examples is more complex, as it relies on the way analyses are described in Démonette. First, let's describe how the third set can be obtained. Since the only condition the verbs must satisfy is that their lemma ends in *aire*, they can be extracted directly from the lexicon of the database, i.e., from the table of lexemes.[1] These verbs are retrieved by constraining the part-of-speech of the lexeme to be V (for "verb") and the written form (in the field GRAPHIE) to end in *aire*. The second constraint is specified by the regular expression `^.+aire$` (where `^` represents the beginning of the lemma, `$` the end of the lemma, and `.+` any sequence of characters):[2] the expression matches any string that begins with any sequence of characters, followed by the sequence *aire*, and such that *aire* is at the end of the string.

The second set is slightly more complex to create because the adjectives it contains are derived from nouns. The table of relations must be used for this set.[3] First, we extract from this table the entries where the first lexeme is an adjective (i.e. with the feature CAT_1 = `Adj`), the second lexeme is a noun (CAT_2 = `N`), and such that the first lexeme is derived from the second one by suffixation in *-aire* (CSTR_1 = `Xaire`), the relation goes from the most complex ('descending') to the simplest ('ascending') (stated by the feature ORIENTATION = `des2as`), and where there is only one derivational step between the two lexemes (COMPLEXITE = `simple`).

The first set is set up in the same way as the second with only two differences: the first lexeme can be a noun or an adjective, and the second must be a verb.

The creation of the three sets corresponds to the first step in Figure 3. It is done automatically. Once the three sets have been created, the script randomly selects two lexemes

[1] See: https://demonette.fr/demonext/vues/table_lexemes.php

[2] It is possible to search for entries in Démonette tables using regular expressions. See Namer et al. (2023) in this volume, and the overview on regular expressions at: https://demonette.fr/demonext/vues/aide_lexemes.php.

[3] See: https://demonette.fr/demonext/vues/table_relations.php; for a comprehensive description of the fields of this table, see the documentation at: https://www.demonext.xyz/en/download-the-base/

in each set, then sorts the six lexemes by alphabetic order so that their position in the list does not indicate the set they belong to (second step in Figure 3).

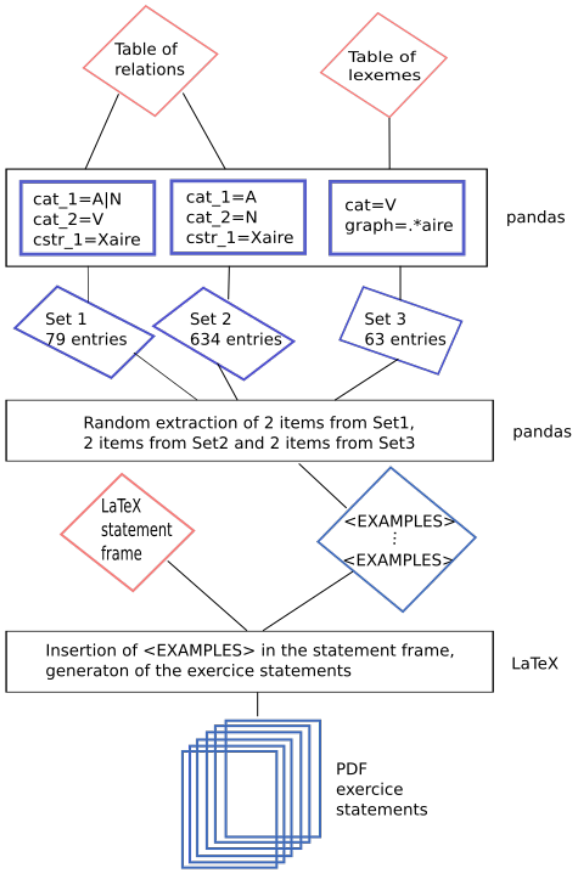

Figure 3. Main steps performed to generate the statement shown in Figure 1. Three sets of lexemes are extracted in the first step. Two examples are extracted from each set in the second step. In the third step, the list of examples is inserted into the template (Figure 2) and the LaTeX document is compiled.

3. Three types of questions

3.1. Processing a set of independent lexemes

Many exercises involve the analysis of a set of independent items, where the analysis of one lexeme in the set does not depend on that of the others. The question may also concern pairs of lexemes that are morphologically related to each other, either directly (one of them being the derivative of the other) or indirectly. The exercise outlined in the previous section is of this type: all the lexemes to be analyzed are independent. As we have seen, it deals with the difference between sequences, exponents and processes. The same is true for the following

exercise. Before going any further, let's clarify what these three terms correspond to, in the context of the analysis of lexemes ending in *-ier*.

In the following, we call **sequence** a string of characters (or of phonemes) that occurs at the beginning or the end of a word, whether or not it has a linguistic function. For example, the sequence *ier* is not a suffix in *cri* 'cry' → *crier* 'to cry', but it is one in *poisson* 'fish' → *poissonnier* 'fishmonger$_{N}$'.

The **exponent** of a word-formation operation is whatever aspect of the output form results from the application of the operation. In simple cases, the exponent will be an affix: for example, we could say that in *lavage* 'washing' the final sequence *age* is the exponent of the suffixation that forms the action noun from the verb *laver* 'to wash'. Word Formation processes may not have an exponent: this is the case of conversion (see e.g., Tribout, 2010, 2012), e.g., in the relation between *danse* 'dance' and *danser* 'to dance', or between *pêche* 'fishing' and *pêcher* 'to fish'.

The notion of **process** considered here assumes that each derivational operation is characterized by three distinct relations: a formal relation, a categorical one, and a semantic one. For example, the formal relation in *laver* → *lavage* is a relation X → Xage, where X represents the verb stem; the categorical relation is V → N; the semantic relation can be glossed as "@ → action of @", where @ represents the verb meaning. Two processes are different if one of the three component relations is different. For example, the *-age* suffixation that forms stative nouns from adjectives (*veuf* 'widow' → *veuvage* 'widowhood') is considered to be a different process from the one that forms verb-based nouns (*laver* → *lavage*).

Exercise 3.1.

Having clarified these three concepts, let's return to the exercise of analyzing lexemes ending with the string *ier*, starting with a set of lexeme pairs as in (1):

(1) *levier*$_{Nm}$/*lever*$_{V}$ '*lever*/raise', *yaourtier*$_{A}$/*yaourt*$_{Nm}$ 'relative to yoghurt/yoghurt', *ailier*$_{Nm}$/*aile*$_{Nf}$ 'winger/wing', *papier*$_{Nm}$/*papetier*$_{Nm}$ 'paper/papermaker', *apparier*$_{V}$/*paire*$_{Nf}$ 'to match/couple', *crier*$_{V}$/*cri*$_{Nm}$ 'to scream/scream'

For example, students could be asked to distinguish between pairs in which *ier* is a sequence, as in *crier*, *papier*, *apparier* and those in which it is a suffix, as in *yaourtier*. Next, the students are asked to identify the processes used in these pairs. The aim is to distinguish deverbal nouns like *levier* from a denominative adjective like *yaourtier*.

How can we create a list of examples for this kind of exercise? The procedure is the same as for the previous exercise. First, we need to identify which of the lexemes ending in *ier* are word-formed by *-ier* suffixation and which are not. Note that the descriptions in Démonette depend on the resources used to feed the database, so there is no guarantee they are exhaustive: the fact that a lexeme is not described as being derivationally complex does not

mean that it actually is a simplex word. Consequently, we can't rely on the contents of the table of relations to retrieve simplex lexemes ending in *ier*. The only option left is to rely on part of speech as we know that *-ier* does not form verbs. Therefore, *ier* is not a derivational exponent in the verbs with a lemma ending in this string, and we can use them as examples of lexemes where *ier* is a sequence.

The set of lexemes formed by suffixation in *-ier* can be created from the table of relations: we just need to extract the entries which contain a lexeme annotated as an instance of the pattern Xier. Next, we need to sort out the lexemes formed by different processes. Because the current version of Démonette does not provide a semantic characterization of derivational relations, we can only rely on part of speech[4] to distinguish between denominative adjectives (*laitier* 'dairy'), denominative nouns (*poissonnier*) and deverbal nouns (*levier*). Then two lexemes are randomly selected from each of the four sets and create a list assigned to the variable that stands for the examples. As you can see, this second example is very similar to the first one. It illustrates how Démonette can be used to vary the nature of the ending (exponent or non affixal sequence), the semantic relation between the pair of lexemes, their grammatical categories, etc.

Exercise 3.2.

Another exercise of the same type would be to list the lexemes that make up the derivational history going from a derived lexeme to one of its distant ascendants, as in (2).

(2) *militer*$_V$/*militantisme*$_{Nm}$ 'to campain/activism', *exclure*$_V$/*exclusionnaire* 'to exclude/ exclusionary', *poule*$_{Nf}$/*poulailler*$_{Nm}$ 'chicken/henhouse'

For example, the derivational histories of the first two examples are: *militer* → *militant* 'activist' → *militantisme* and *exclure* → *exclusion* 'exclusion' → *exclusionnaire*, respectively. Selecting pairs of lexemes as in (2) presents no difficulty. One just has to select the lemmas (GRAPH_1 and GRAPH_2) in the entries in the table of relations where the feature COMPLEXITE has the value `complexe` (this value indicates that the derivational path between the two lexemes includes more than one step) and the feature ORIENTATION has the value `as2des` (this value indicates that the first lexeme is an ascendant of the second one). Démonette contains 153 such pairs.

Exercise 3.3.

A variant of the same exercise involves analyzing pairs of lexemes, where one of them results from a parasynthetic formation (for a discussion of the notion of parasynthesis, see Hathout, 2011; Heyna, 2014; Hathout & Namer, 2018). In these pairs, form and meaning do not match.

[4] We may use additional resources to approximate the semantic classes of a pair of lexemes from their ontological categories, as illustrated in the exercise in (5).

In (3), the meaning of the adjectives is built from that of the noun, but its form is not. For instance, the meaning of *antigrippal* is built from that of *grippe* (anti-influenza medicine cures the flu), while its form is coined from that of the relational adjective of *grippe*, namely *grippal* 'relative to flu'. This type of discrepancy is described in the field COMPLEXITE in Démonette by the value `motiv-sem`, which specifies that the meaning of the first lexeme only is built on that of the second lexeme.

Selection of the examples in (3) is just as easy as selection of the examples in (2): We just need to retrieve the lemmas (GRAPH_1 and GRAPH_2) in the entries in the table of relations where COMPLEXITE = `motiv-sem`.

(3) *préglaciaire*$_{A}$/*glaciation*$_{Nf}$ 'pre Ice Age/glaciation', *antigrippal*$_{A}$/*grippe*$_{Nf}$ 'anti-influenza/flu', *biculturel*$_{A}$/*culture*$_{Nf}$ 'bicultural/culture', *contreterroriste*$_{A}$/*terrorisme*$_{Nm}$ 'counter-terrorist/terrorism'

Exercise 3.4.

The same type of exercise can be used to explore the notion of morphological families and their alignment in derivational paradigms (Bonami & Strnadová, 2019; Hathout & Namer, 2019, 2022). For example, the pair *contestataire/contestation* can be completed by lexemes such as *contester* 'to contest$_{V}$' and *contestable* 'contestable'. Similarly, *protestataire/protestation* belongs to a family that also includes *protester* 'to protest' and *protestable* 'protestable', and *dérogataire/dérogation* belongs to a family that includes *déroger* 'to derogate' and *dérogeable* 'derogable'.

(4) *contestataire*$_{A}$/*contestation*$_{Nf}$ 'anti-establishment/protest', *protestataire*$_{A}$/*protestation*$_{Nf}$ 'protestor /protest', *dérogataire*$_{A}$/*dérogation*$_{Nf}$ 'derogatory/derogation'

Here again, the selection of examples in (4) is straightforward. The pairs are characterized by the fact that their ORIENTATION feature has the value `indirect` (indicating that neither lexeme is derived from the other), and their COMPLEXITE feature has the value `simple` (since both lexemes are derived from the same base).

Exercise 3.5.

The examples used in the exercises can also be selected using external resources, and in particular semantic annotations not yet included in Démonette. In the following exercise, for example, the aim is to show the diversity of processes that can be used to construct human nouns (5).

(5) *actionnaire*$_{N}$/*action*$_{Nf}$ 'shareholder/share', *albanais*$_{Nm}$/*Albanie*$_{Npx}$ 'Albanian/Albany', *chasseur*$_{Nm}$/*chasser*$_{V}$ 'hunter/to hunt', *antipape*$_{N}$/*pape*$_{Nm}$ 'antipope/pope', *barbu*$_{Nm}$/*barbu*$_{A}$ 'beared/bear', *américaine*$_{Nf}$/*Amérique* 'American/America', *castrat*$_{Nm}$/*castrer*$_{V}$ 'castrato/to castrate'

To identify these nouns, we use the annotations of Huguin et al. (2023, this volume), which give the ontological class of a subset of the nouns of the table of lexemes. To collect the examples used in this exercise, we first perform a join between the semantic annotation table and the lexeme table to select the nouns encoded as human (i.e., of the *Person* class). We then project these annotations onto the table of relations using a second join. Finally, we extract the entries describing simple direct relations from the second join to only keep the word pairs where the human noun is derived from the other lexeme.

Exercise 3.6.

A variant of the same exercise involves extracting noun/verb conversion pairs from the table of relations (without condition on the orientation of the relation), and then selecting pairs where the noun is semantically annotated as an action (Action), an artifact (Artefact), or a human being (Person) in the table of semantic annotations. Examples of such pairs are given in (6). The variant focuses on the impact of semantic properties on the cross-definition of noun and verb that are in a conversion relation. For example, in (6), when the noun refers to a person (*ventriloque*), the associated verb refers to an act of imitation (*ventriloquer*). If it refers to an artifact (*bague*, *vitriol*), the verb can be instrumental (*vitrioler*) or ornamental (*baguer*). Finally, an action noun (*zigzag, caricature*) can be interpreted as the result of the activity denoted by the verb: a zigzag or a caricature is what you do when you zigzag or caricature somebody. This result may be concrete, as in *caricature*.

(6) *zigzag*$_{Nm}$/*zigzaguer*$_V$ ‘zigzag/to zigzag’, *vitriol*$_{Nm}$/*vitrioler*$_V$ ‘vitriol/to use vitriol’, *bague*$_{Nf}$/*baguer*$_V$ ‘ring/to attach a ring to’, *caricature*$_{Nf}$/*caricaturer*$_V$ ‘caricature/to caricature’, *ventriloque*$_N$/*ventriloquer*$_V$ ‘ventriloquist/to project one’s voice without moving the lips’

3.2. Process a list of morphologically related lexemes

The second type of exercise involves one or more lists or tuples of lexemes from the same family (i.e., morphologically related lexemes), as in (7). The example consists of a noun followed by a number of possible forms of *-at* derivatives obtained by using several stems of the noun, in this case *directeur* and *director*. In addition, derivatives include either the expected suffix *-at* (*directeurat*), or the variant *-iat* (*secretariat*), or the variant *-ariat* (*vedettariat*). The exercise consists of noting the variations in the stem base (/diʁɛktœʁ/ is realized as /diʁɛktɔʁ/), and in the exponent (/a/, /ja/, /aʁja/) and explaining them (on these variations, see Plénat & Roché, 2014).

(7) a. *directeur*$_{Nm}$ ‘director’, *directeurat*$_{Nm}$ ‘directorate’, *directorat*$_{Nm}$ ‘directorate’, *directoriat*$_{Nm}$ ‘directorate’

b. *sous-chef*$_{Nm}$ ‘underchef’, *sous-chefat*$_{Nm}$ ‘function of underchef’, *sous-cheffariat*$_{Nm}$ ‘function of underchef’, *sous-chefiat*$_{Nm}$ ‘function of underchef’

The main difference with the previous type of exercise is that we need to compile lists of lexemes. In this case, we have to:

- select the entries in the table of relations that correspond to a *-at* derivation (CSTR_1 = `Xat`),
- group these entries relatively to the second lexeme (GRAPH_2),
- collect the GRAPH_1 of each group in a list,
- select lists longer than 2.

These operations must be carried out using a data analysis library such as pandas (Section 6). In a similar type of exercise, students may be asked to reconstruct the graph of relations between one lexeme and other members of its derivational family.

3.3. Group lexemes from a list into subclasses

The third type of exercise is a variation of the exercises presented in Section 3.1. In these exercises, the students are asked to find remarkable properties (semantic, categorical, morphophonological, etc.) of the units under study by comparing the proposed examples, with the added difficulty that the examples may show these differences more or less clearly. The idea here is to include in the statement examples that illustrate these properties, grouped into classes of interest, then to propose a new list of lexemes and ask the students to determine the class each lexeme belongs to.

For example, we may ask them to distinguish the following four types of nouns in *-ité* (Koehl, 2012; Koehl & Lignon, 2014). Examples of each type are shown in (8):

(8) a. *absurdité*$_{Nf}$ ‘absurdity’, *acerbité*$_{Nf}$ ‘acerbity’, *aridité*$_{Nf}$ ‘aridity’, *débilité*$_{Nf}$ ‘debility’

b. *portabilité*$_{Nf}$ ‘portability’, *respectabilité*$_{Nf}$ ‘respectability’, *sociabilité*$_{Nf}$ ‘sociability’, *variabilité*$_{Nf}$ ‘variability’

c. *complémentarité*$_{Nf}$ ‘complementarity’, *scolarité*$_{Nf}$ ‘schooling’, *polarité*$_{Nf}$ ‘polarity’, *similarité*$_{Nf}$ ‘similarity’

d. *sévérité*$_{Nf}$ ‘severity’, *obscénité*$_{Nf}$ ‘obscenity’, *prospérité*$_{Nf}$ ‘prosperity’, *obésité*$_{Nf}$ ‘obesity’

The proposed exercise consists of determining the type of *-ité* derivatives of a list of nouns as in (9), explaining the differences between the four types and establishing the lexeme formation rules used to build them, where appropriate.

(9) *austérité*$_{Nf}$ ‘austerity’, *bestialité*$_{Nf}$ ‘bestiality’, *commodité*$_{Nf}$ ‘convenience’, *exemplarité*$_{Nf}$ ‘exemplarity’, *fidélité*$_{Nf}$ ‘loyalty’, *maniabilité*$_{Nf}$ ‘manageability’, *popularité*$_{Nf}$ ‘popularity’, *responsabilité*$_{Nf}$ ‘responsibility’

The analysis of (9) includes the following points:

- The derivational base of all nouns in (8) and (9) are syncretic adjectives, i.e. with identical masculine and feminine inflected forms;
- In class (8a) the noun is formed by the simple suffixation of /ite/ to the adjective stem. The nouns *bestialité* ‘bestiality’ and *commodité* ‘convenience’ belong to (8a).
- In (8b), the base adjective is formed by suffixation in *-able*; the sequence /abl/ appears as /abil/ in the noun. Class (8b) includes *maniabilité* ‘manageability’ and *responsabilité* ‘responsibility’.
- In (8c), the base adjective is formed by suffixation in *-aire*; the final sequence /ɛʁ/ of the adjective is realized as /aʁ/ in the noun. Class (8c) includes *exemplarité* ‘examplarity’ and *popularité* ‘popularity’.
- In (8d), the vowel in the last syllable of the root of each base adjective (*sévère* ‘severe’, *obscène* ‘obscene’, *prospère* ‘prosperous’, *obèse* ‘obese’) is /ɛ/. It closes in /e/ on contact with /ite/. The examples *austérité* ‘austerity’ and *fidélité* ‘fidelity’ belong to class (8d).

The examples in the exercise can be selected by combining the descriptions in Démonette’s tables of lexemes and relations. The table of lexemes provides the phonetic transcription of the adjective and noun paradigms. The table of relations provides adjective-noun pairs in which the noun is derived from the adjective by suffixation in *-ité*.

Exercise generation includes the following steps. All steps are done automatically.

- From the table of lexemes, we extract the set SyncreticAdjSet of adjectives (CAT = `Adj`) in *able*, (GRAPH = `^.*able$`), in *aire* (GRAPH = `^.*aire$`) and the adjectives with

identical value in all phonetic transcription cells of their inflectional paradigm (PARA_PHON).

- From the table of lexemes, we extract the set IteNounSet of feminine nouns (CAT = `Nf`) ending in *ité*, (GRAPH = `^.*ité$`).
- From the table of relations, we extract the set XiteRelSet of word pairs formed by an adjective (CAT_1 = `Adj`) and a noun (CAT_2 = `Nom`) such that the noun is derived from the adjective (COMPLEXITE = `simple`, ORIENTATION = `as2des`) by suffixation in *-ité* (CSTR_2 = `Xité`).
- Pairs for which the adjective is not part of SyncreticAdjSet are excluded from XiteRelSet.
- From XiteRelSet, SyncreticAdjSet, and IteNounSet, we create the set ARelSet of adjective-noun pairs of class (8a) by only keeping the pairs where the noun stem does not vary with respect to the adjective stem. To do this, we check that the phonetic representation of the adjective (i.e., the value of the PARA_PHON feature in SyncreticAdjSet) is identical to the phonetic representation of the corresponding noun (i.e., the value of PARA_PHON in IteNounSet), stripped of the final /ite/.
- From XiteRelSet, we create the set BRelSet of adjective-noun pairs of class (8b) by only keeping word pairs with an adjective ending with *able* (GRAPH_1 = `^.*able$`).
- From XiteRelSet, we create the set CRelSet of adjective-noun pairs of class (8c) by only keeping word pairs with an adjective ending with *aire* (GRAPH_1 = `^.*aire$`).
- From XiteRelSet, we create the set DRelSet of adjective-noun pairs of class (8d) by only keeping word pairs with an adjective ending with *èCe*, where *C* stands for a consonant, i.e., such that GRAPH_1 = `^.*è.e$` (actually, this condition allows any character between 'è' and 'e').

Other examples of exercises are included in the demonstrator available on Demonext's git repository.[5]

4. General architecture

The tool we present has been developed in the python programming language (Van Rossum & Drake, 2009). It was implemented in the form of a Jupyter notebook (Kluyver et al., 2016). A notebook is a document containing both formatted text, code, in our case in python, and execution results. Data analysis and example selection are performed in pandas (McKinney et al., 2010), a data manipulation and analysis library. Intermediate LaTeX documents are generated from templates containing the fixed parts of the statements and a file containing

[5] https://github.com/demonext/Exercices

the examples to be included in the template. The template contains the document title, section titles, exercise presentation and questions. It also contains variables whose values are the text of the examples to be included in the statements. A file of examples is created for each statement. Compiling a statement therefore requires two files: a template and a file containing the examples.

For example, the exercise presented in Section 2 is generated from the following LaTeX template.

```
\subsection*{One process or several processes?}

The following 6 lexemes all end with the graphic sequence
\emph{aire}. Suggest a derivational analysis for each of them. In
particular, answer the following questions:
\begin{itemize}
   \item Are they all formed by derivation?
   \item Are they derived by the same process?
\end{itemize}

\begin{center}
   \lexemes_aire
\end{center}
```

Figure 4. LaTeX template of the statement of Figure 1.

The template contains a variable (i.e., a command) `\lexemes_aire` instantiated in a separate file of examples, as shown in Figure 5. The assignment of these variables corresponds to step 3 in Figure 3 (Section 2).

```
\newcommand{\lexemes_aire}{aéroportuaire, commentaire,
dentaire, déplaire, retardataire, taire}
```

Figure 5. Instanciation of the variable `\lexemes_aire` in the file of examples used to produce the statement in Figure 1.

The program also uses a LaTeX header suited to the specificity of the system on which the documents are compiled. It specifies the font of the document and packages to be loaded.

Data selection is carried out by means of relatively simple pandas queries when the conditions concern information present in the fields of the Démonette tables and which need not be manipulated. The pandas library also simplifies the combination of information from

Démonette with other descriptions from external resources, such as the semantic annotations of nouns by Huguin et al. (2023, this volume).

Discussion

The work we have just presented illustrates the coverage and the accuracy of the descriptions encoded in the Démonette database. Note that several other French resources do include morphological annotations, though their coverage and precision vary. This is the case of the Diko, a French lexical database from the JeuxDeMots project (Lafourcade, 2007), which contains an exceptionally rich range of relations, but whose coverage is often limited. For example, we only found 821 occurrences of the morphological derivational relations (r_der_morpho).[6] This figure represents a tiny fraction of the 222 118 pairs in Démonette's table of relations. Other resources, such as the French part of UniMorph (Batsuren et al. 2022), have very extensive coverage, but cannot be used to produce exercises because they lack sufficient annotations. Using tools like ChatGPT is not an option either, as its answers are not always reliable and are therefore not suitable for teaching. For example, ChatGPT wrongly analyzes *adiposité* 'adiposity' as a noun built on the noun *adipose* 'adipose' (which denotes a morbid condition characterized by fatty tissue overload) instead of deriving it from the adjective *adipeux* 'fat'.

As mentioned in Section 1, the tool presented in this paper is a simple demonstrator. It has several limitations and could be improved in several ways. The first concerns the code and its documentation. For now, the design of the statements is completely handmade, and the coding of the tool does not conform to any standard and has no documentation. Improvements to this tool could be made in several directions: make the program more modular and conform to software development standards; produce a documentation; describe the selection of the inputs in an external file in the form of attribute-value pairs; propose outputs in formats that can be integrated into a MOOC. Another limitation is that this work has not been evaluated in terms of the benefits, or lack of, of the generated exercises (condition where students work on statements containing automatically selected examples) compared with conventional exercises (condition where students all work on the same examples). Such an evaluation goes beyond the scope of the present work, which aims at illustrating the possible uses of the Démonette database and is not really a CALL (Computer Assisted Language Learning) project. Another limitation of this work is its lexeme-based approach, and the fact that it cannot be

[6] These word pairs are available at: https://www.jeuxdemots.org/JDM-LEXICALNET-FR/01022020-LEXICALNET-JEUXDEMOTS-R99.txt (accessed on August 23, 2023); JeuxDeMots also includes 9 609 action noun/verb relations and 10 137 verb/action-noun relations.

used to produce exercises for teaching in other framework, notably in morpheme-based approaches, including distributed morphology (Halle & Marantz, 1993). While lexeme-based morphology is the dominant approach to derivation in France, this is not the case in other countries, where generative approaches remain mainstream.

Conclusion

In this paper, we presented a tool for morphology teaching based on the Démonette database. The tool is essentially a demonstrator that generates large numbers of morphology exercises for training and knowledge evaluation.

The tool automatically produces many instances of a set of exercises. These instances differ only in the examples they contain. The examples are selected from the annotations present in Démonette, and possibly from other resources. The article shows that, for many exercises, the information used is explicitly encoded in Démonette tables. But it's also possible to generate exercises that cover relatively complex configurations using the features offered by the pandas data manipulation library. Eventually, we plan to turn this tool into a library of exercises extensive enough to meet the main needs of teachers of undergraduate courses in derivational morphology.

References

Agirrezabal, M., Begona, A., Gil-Vallejo, L., Goikoetxea, J., & Gonzalez-Dios, I. (2019). Creating vocabulary exercises through NLP. In *Digital Humanities in the Nordic Countries. Proceedings 2019* (pp. 18–32). Copenhagen, Denmark.

Baptista, J., Lourenco, S., & Mamede, N. J. (2016). Automatic generation of exercises on passive transformation in Portuguese. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 4965–4972). IEEE Press.

Batsuren, K., Goldman, O., Khalifa, S., Habash, N., et al. (2022). UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 840–855). Marseille, France.

Bodnar, S. (2022). The instructional effectiveness of automatically generated exercises for learning French grammatical gender: preliminary results. In *Swedish Language Technology Conference and NLP4CALL* (pp. 10–22). https://aclanthology.org/2022.nlp4call-1.2.pdf

Bonami, O., Boyé, G., & Kerleroux, F. (2009). L'allomorphie radicale et la relation flexion-construction. In B. Fradin, F. Kerleroux & M. Plénat (Eds), *Aperçus de morphologie du français* (pp. 103–125). Presses Universitaires de Vincennes.

Bonami, O., & Strnadová, J. (2019). Paradigm structure and predictability in derivational morphology. *Morphology*, *29*(2), 167–197.

Chinkina, M., & Meurers, D. (2017). Question Generation for Language Learning: From ensuring texts are read to supporting learning. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 334–344). Copenhagen, Denmark.

Hathout, N. (2011). Une analyse unifiée de la préfixation en *anti-*. In M. Roché, G. Boyé, N. Hathout, S. Lignon & M. Plénat (Eds), *Des Unités Morphologiques au Lexique* (pp. 251–318). Hermès.

Hathout, N., & Namer, F. (2018). La parasynthèse à travers les modèles : des RCL au ParaDis. In O. Bonami, G. Boyé, G. Dal, H. Giraudo & F. Namer (Eds), *The lexeme in descriptive and theoretical morphology* (pp. 365–399). Language Science Press.

Hathout, N., & Namer, F. (2019). Paradigms in word formation: what are we up to? *Morphology*, *29*(2), 153–165.

Hathout, N., & Namer, F. (2022). ParaDis: a Family and Paradigm Model. *Morphology*, *32*(2), 153–195.

Halle, M., & Marantz, A. (1993). Distributed Morphology and the Pieces of Inflection. In K. Hale & S. J. Keyser (Eds), *The View from Building 20* (pp. 111–176). MIT Press.

Heyna, F. (2014). *Etudes morpho-syntaxique des parasynthétiques.* De Boeck Supérieur.

Heck, T., & Meurers, D. (2022). Parametrizable exercise generation from authentic texts: Effectively targeting the language means on the curriculum. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)* (pp. 154–166). Seattle, Washington.

Huguin, M., Barque, L., Hass, P., & Tribout, D. (2023). Typage sémantique des noms en français pour la ressource lexicale morphologique Démonette. *Lexique*, 33.

Koehl, A. (2012). *La construction morphologique des noms désadjectivaux suffixés en français.* Thèse de doctorat. Université de Lorraine.

Koehl, A., & Lignon, S. (2014). Property nouns with *-ité* and *-itude*: formal alternation and morphopragmatics or the sad -itude of the *Aité*$_{N}$. *Morphology*, *24*(4), 351–376.

Kledilson, F., & Pereira, A. R. (2018). Verb Tense Classification And Automatic Exercise Generation. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web (WebMedia '18)* (pp. 105–108). New York, NY, USA.

Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds), *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87–90). IOS Press.

Lafourcade, M. (2007). Making people play for Lexical Acquisition with the JeuxDeMots prototype. In *7th international symposium on natural language processing (SNLP'07)* (pp. 13–15). Pattaya, Thailand.

McKinney, W., et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (pp. 51–56). Austin, Texas, United States. DOI: 10.25080/Majora-92bf1922-00a

Namer, F., Hathout, N., et al. (2023). Démonette-2, a derivational database for French with broad lexical coverage and fine-grained morphological descriptions. *Lexique*, 33.

Roché, M. (2010). Base, thème, radical. *Recherches linguistiques de Vincennes*, *39*, 95-134.

Tribout, D. (2010). *Les conversions de nom à verbe et de verbe à nom en français*. Thèse de doctorat, Université Paris 7.

Tribout, D. (2012). Verbal stem space and verb to noun conversion in French. *Word Structure*, *5*(1), 109–128.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.

Wilson, E. (2014). The automatic generation of CALL exercises from general corpora. In A. Wichmann & S. Fligelstone (Eds), *Teaching and language corpora* (pp. 116-130). Routledge.