Computer-Aided Inflection for Lexicography Controlled Lexica

Maarten Janssen
IULA - UPF
Roc Boronat 138, 08018 Barcelona
E-mail: Maarten.Janssen@upf.edu

Abstract

This article describes the design of a computational system for the development and maintenance of inflected lexica, developed as part of the Open Source Lexical Information Network (OLSIN). The system is built as a tool for lexicographers, and is flexible enough for the lexicographers to deal with any irregularities in the language, and transparent enough for the lexicographers to understand the rules used for the automatically generated inflections. It furthermore allows lexicographers to create and modify paradigm rules by themselves, making it easy to implement the system for any language, including less-resources languages. Apart from the system itself, this article describes some of the challenges and obstacles the design of such a system has to face, and the solutions adopted for them in the OSLIN framework.

Keywords: inflectional morphology; paradigm system; full-form lexicon

1. Introduction

Dictionaries have always included inflectional information, though be it only in limited amounts. Inflectional information is traditionally limited to the key word-forms of irregularly inflected words only, due to restriction on the size of dictionaries. With the rise of electronic dictionaries and the loosening of size restrictions they imply, this situation has changed: more and more dictionaries include the full inflection for all words, or at least for all words of heavily inflecting word-classes: in modern Spanish dictionaries, all verbs are provided with their full inflection, although nouns and adjectives are not. This information is included for two reasons: firstly, it is lexical information that many people are interested in, and therefore information that is useful to include when possible. Secondly, it allows the user to find words in the dictionary without knowing the citation form: if you do not know that *fue* is an inflected form of *ser* (to be) in Spanish, it is difficult to find in a dictionary that does not include inflected forms. With inflection moving into a prominent place in dictionaries, their quality should match up with the rest of the dictionary.

To add inflection to dictionaries, most dictionaries rely on computational tools. Yet with respect to the creation of the inflected forms there is a tension between automation and freedom: it is hard at best to created a large-scale inflected lexicon without the use of computational tools, yet computational tools tend to limit the lexicographer in defining precisely the inflection he deems correct.

This article describes a computational system that aims to overcome this tension by allowing the lexicographer full control over the inflected forms, while at the same time automating the process of inflection as much as possible. This tool, which forms part of the Open Source Lexical Information Network (OSLIN) framework (Janssen, 2005), does this by using a paradigm-based inflection system where the lexicographer himself can create, apply, and modify the paradigms. It is a tool designed for practical usability for lexicographic purposes, without too much emphasis on computational innovation or efficiency.

The set-up of this article is as follows: the next section describes which requirements a computational tool has to meet in order to allow lexicographer sufficient guidance and freedom to develop a high quality inflectional database. Section 3 describes how these requirements are implemented in the OSLIN environment. Section 4 describes how the OSLIN tool can, and has been used in practice to create a large-scale, high-quality full-form lexicon. And section 5 describes some of the more complex issues that play a role in the task of semi-automatic paradigmatic inflection.

2. Design Requirements

It is possible to create a full-form lexicon by manually inserting all inflectional forms. However, to do so is very labour intensive: a full-form lexicon for a morphologically rich language typically contains many more forms than it does lexical entries: for Portuguese for example, every lexical entry has on average 10 forms, meaning that a medium-sized dictionary contains well over a million inflected forms. Not only is it labour intensive, it is also extremely hard to avoid typographic errors in that many inflected forms, especially since the inflected forms of a word typically differ very little amongst each other. Therefore, the use of computational tools in this task is highly desirable, both to save work, and to prevent errors.

However, inflection can be a complex issue: although for the majority of words, it is clear and undisputed how they inflect, there are many cases where inflection is less clear and where information has to be taken into account that is not (easily) computationally available, such as normative rules, etymology and pronunciation. Even relying on usage information is not necessarily sufficient:

it is not unseen for a noun that the plural that is considered normatively correct is not the plural that is most frequently used. Another complication is that sometimes, a form other than the standard form is used as the citation form, for instance for noun that are almost exclusively used in the plural.

In the more complex cases of inflection, the computational tools used should not limit the lexicographer in defining exactly the inflected forms he/she considers correct: for instance, it should be a lexicographic decision whether to include the normative forms, the most frequently used forms, or both in those cases in which these two are not the same. The computational mechanisms should therefore suggest an inflectional paradigm to the lexicographer, but always allow changing or overruling the suggestions made.

The easiest way to implement a computational system that produces suggestions that can be modified at will is by having an external system that only intervenes during the creation of a new lexical entry: when creating new word, the lexicographer can choose to either accept the suggestion made by the tool and insert the computationally created forms into the dictionary, or he can ignore the suggestion, and choose to either insert the suggested forms but modify them afterwards, or completely ignore the suggestion and add the forms manually.

Although this is a workable system, and the method used in the OSLIN framework, it has two major drawbacks. Firstly, such a system often works as a black box, which makes it a lot harder for the lexicographer to spot potential errors. And second, the manually corrected or inserted forms are still subject to typographic errors. This means that over time, it is inevitable that errors creep in. And because it is not longer transparent which forms were automatically created, and which were manually altered, it becomes very hard to spot those errors in a large-scale dictionary. This is not merely a theoretical point, but shown by experience in OSLIN before the introduction of the paradigm system described here.

For these reasons, a more reliable set-up is a system in which the inflectional mechanisms are build into the dictionary system in a more integrated way, so that manual correction can be spotted, and ideally, the manual corrections are even done via the same computational tool. Furthermore, it is important that the computational implementation is done in a transparent fashion, so that the lexicographer can understand why the system suggests the inflected forms it does. This makes it much easier for the lexicographer to work with the system, and avoids errors.

The most intuitive way of computationally treating inflection is by using a paradigm-based system: a word is assigned to an inflectional paradigm, and based on that paradigm, it gets assigned a number of inflected forms. Although there are other, more computationally efficient ways of encoding inflection, it is much more difficult to make the inner workings of such systems clear to a lexicographer without a computational background.

An additional advantage of a paradigm-based system is that it is well-known strategy in the lexicographic tradition, used in grammar books and existing (paper) inflection dictionaries for many inflecting languages such as for instance *Els Verbs Conjugats* (Baptista Xuriguera, 2009) for Catalan. In order to explain to the end user how to inflect the Portuguese verb *bailar* (to dance), it is possible list all the (73) inflected forms, but it will in most cases be sufficient to say that it is completely regular, or that it inflects like *amar* (to love). Indicating the inflectional paradigm not only is a convenient way for the user to understand the inflection, but also allows the user to see other words that inflect in the same way.

If paradigms are presented to the end user, the paradigm system not only has to be able to correctly inflect all words in the lexicon, but also correspond to what are traditionally considered to be the paradigms of the language. This seems a trivial issue, but there are many divergences between a computational and a human perspective on inflectional paradigms (see section 5.1).

There is no single unique way to set-up a paradigm system: there are typically several different sets of paradigms, each of which sets can equally be used to define the inflection of a language. The choice of paradigms itself can in such cases itself become a lexicographic or political matter. Therefore, it is very convenient if the system allows the lexicographer to construe the paradigms himself, and ideally do so without having to rely on a computational linguist. Furthermore, it should be possible to modify the paradigms when the need arises, for instance when the orthography changes, or when it is decided that another choice of paradigms is more appropriate.

Finally, a computational tool for the inflection of a lexicon should ideally be, as far as possible, language independent. That is to say, it should be usable for as many languages that have an inflectional morphology as possible, so that the same computational tool can be used for a wide variety of languages.

2.1 Open Source Lexical Information Network

The inflection system described in this article is part of the Open Source Lexical Information Network (OSLIN). OSLIN is a language independent framework for modeling lexical information, with a focus (for the moment) on inflectional and derivational morphology. The system was originally developed at the ILTEC institute in Lisbon for Portuguese, and has since been extended at the IULA institute in Barcelona to several other language in various degrees of detail, including

Spanish, Catalan, Russian, Dutch, and French. OSLIN is a web-based system, and most of the lexicons can be accessed via the project website (www.oslin.org).

The inflectional part of OSLIN consists of a relational database with two tables. The first table contains the lexical entries with their citation form, word-class and other information, including the inflectional paradigm. The second table contains for each lexical entry all the inflected forms related to it, with their orthography and an indication of which inflected form it is.

The paradigm system of OSLIN, described in this article, is used in several ways within the framework. For the lexicographer, the system is used to create and fill the forms table with all inflectional forms for new words, but also to correct words already inflected. And the system can be used to make selections of similarly inflected words to facilitate correction processes. For the end user, the paradigms assigned by the system are used in the online display of the inflectional information for each word in the lexicon, as well as to display each paradigm with its inflected forms, together with a list of all the words that are inflected via that paradigm.

3. OSLIN Paradigm Manager

In a computational paradigm-based inflection system, there are three different aspects to the design and use of paradigms: firstly, there are the rules or mechanisms that define the paradigm itself. Secondly, there is a need for a system to create those paradigmatic rules, and finally, these rules have to be applied to create the actual inflected forms for the words in the lexicon. This section describes the way these three aspects are implemented in the OSLIN inflectional system. Most of the examples given in this section are in Spanish, but there is nothing specific to Spanish in the design, and examples from for instance Russian would work just as well.

3.1 Paradigm Definitions

A paradigm in the OSLIN system is an entity for creating inflection forms for a selection of words in a specific word class. Each paradigm has a unique identifier, which indicates the word-class it relates to, followed by a sequential number. For example, ADJ01 is the first paradigm for the inflection of adjectives. To make it easier to identify the paradigms, each paradigm furthermore has a prototypical word associated with it, which is typically the most recognizable word that belongs to that paradigm. For Spanish, the example word for ADJ01 is the adjective *gordo* (fat), being the paradigm to inflect *gordo* and all words that inflect like it. In this article, paradigms will be identified always by their prototypical word for ease of reading.

The core of the paradigm is a set of string transformation rules: rules that create the orthography for all the inflected forms, starting from the orthography of the citation form, by transforming the string of characters. The reason why the rules start from the citation form is that they lexical entries in dictionaries are identified by their citation form or headword. These string transformation rules define that the inflected for the word *gordo* with paradigm ADJ01 are: *gordo, gorda, gordos,* and *gordas* respectively, and the likewise relate all similarly inflecting words to their respective inflected forms. There are three types of string transformation rules in the OSLIN system: root-creation rules, inflection rules, and root-alternation rules.

For each paradigm, the root-rule defines how to generate the "root", or better the invariant part, of the paradigm based on the citation form. The root in the paradigm system is not necessarily the linguistic root, but simply the part of the citation form that remains invariant throughout the forms in the paradigm. For readability, a hyphen is placed at the end of the root in the examples in this article; this hyphen does not correspond to anything in the actual rules. An example of a root-creation rule is given in (1), which is the root-creation rule for the Spanish adjectival paradigm *gordo*. The rule is a regular expression rule, here formulate in Perl for convenience.

(1) ($root = $citation_form) =~ s/o$//;

The root-creation rule in (1) states that the root for *gordo* is formed by removing the *–o* at the end of the citation form. This rule generates the root *gord-* for *gordo*, *blanc-* for *blanco*, etc.

The inflection rules are rules that create individual inflected forms from the root. Each paradigm has as many inflection rules as there are inflected forms for the paradigm. For the paradigm *gordo*, there are therefore four inflection rules, one for each of the four adjectival forms in Spanish (masculine and feminine singular and plural). The inflection rule for the feminine plural of *gordo* is given in (2).

(2) ($inflection['fem plur'] = $root) =~ /$/as/;

The inflection rule in (2) states that the feminine plural of *gordo* consists of the root, with the suffix *–as* added to the end. Together with the root-creation rule in (1), this defines that the feminine plural form of *gordo* is *gordas*, for *blanco* it is *blancas,* etc.

The reason why paradigms are defined in terms of simple string transformation rules, and not for instance in the more powerful system of Two Level Morphology (Koskenniemi, 1983), is the aforementioned fact that paradigms are the most intuitive way for lexicographer to deal with inflection.

All inflected forms can be defined in terms of string transformation rules from the citation form, although sometimes only in a trivial way: in a paradigm where no letters are shared between all forms, the "root" form has

to be empty, and the word-forms are created by adding the entire word to the empty root.

However, in many cases simple transformation rules lead to paradigms that do not correspond to the traditional paradigms of the language. For example, the Dutch words *jaar* (year) and *boor* (drill) are typically seen as inflecting the same: in their plural form, the double vowel is replaced by a single vowel, leading to the respective plurals *jaren* and *boren*. In a straight-forward string transformation system, these words would, however, end up as having different paradigms: one where the ending *–ar* is replaced by *–ren*, and another where the ending *–or* is replaced by *–ren.*

To implement paradigms in a way that closer resembles traditional paradigms, OSLIN uses root-alternation rules. Root-alternation rules create alternate root forms from the base root form. When using multiple roots, the inflection rules have to indicate which of the root forms is used for each particular form. An example is given in rules (3)-(5). The root-creation rule in (3) defines that the (main) root for *jaar* is identical to the citation form. The root-alternation rule in (4) creates a second form of the root by removing the double vowel in the final syllable of the main root. Finally, the inflection rule in (5) defines that the plural for *jaar* is formed by placing *–en* at the end of the alternate root.

(3) ($root = $citation_form) =~ s/$//;

(4) ($alt_root[1] = $root) =~ s/([aeou])\1([dgklmnprt])$/\1\2/;

(5) ($inflection['plural'] = $alt_root[1]) =~ /$/en/;

The rules (3)-(5) for the word *jaren* first create a root *jaar-*, then create an alternative root *jar-*, and then form the plural *jaren* from the alternative root form. In the case of *boor*, the rules create *boor-*, *bor-* and *boren* respectively.

Root alternation rules can be used to group words together under the same paradigm that are traditionally consider to inflect alike. However, they can also be used to make the "root" of the paradigm resemble the linguistic root more closely. For instance, in the Spanish nominal paradigm *actriz/actrices* (actress), rules without root alternation use a root form *actri*, with endings *–z* and *–ces* in the singular and the plural form. With root alternation, it is possible to create the same form in a more linguistically appropriate way, defining *acriz* as the (main) root, with a plural ending *–es* and a root-alternation rule *z/c*.

3.2 Creating Paradigms

Paradigm-based inflection systems using string transformation or transduction rules are far from new, dating back at least to Matthews (1972). It is not difficult to generate this type of rules by hand, even though exceptions in the inflection of a language can make the set of rules quite complex. However, if we want the lexicographer to be able to create and maintain the paradigm system by himself, without having to possess a lot of computational knowledge or a computation linguist, the system should not rely on manually created rules.

Therefore, in OSLIN the rules for the paradigms are created automatically, based on example data provided by the lexicographer. The way this works is very simple: the lexicographer types in all the inflected forms of an example word by hand, and the system attempts to determine which rules have to be defined in order to generate all the forms that the lexicographer entered, starting from the citation form.

In order to allow adding the inflected forms, the only thing the system needs to know is which forms the word has, and a graphical way to organize these forms to make the data easier to read. In OSLIN, these two things are handled by a template: for each (major) word class, there is a template that defines a graphical display of all the inflected forms for that word class. A template is a simple HTML text file, containing this information. An example of a template for Spanish adjectives is given in figure 1.

```
<table>
     <tr><td><th>masulin<th>feminin
     <tr><th>singular<td>{%ms}<td>{%fs}
     <tr><th>plural<td>{%mp}<td>{%fp}
</table>
```

Figure 1: Paradigm Template for Spanish Adjectives

The template on the one hand defines that adjectives in Spanish have four forms, labels by *ms, fs, mp,* and *fp* respectively. And it defines an HTML table to graphically display these four forms in a convenient way. The template is both used to display the inflection of already inflected words, and to create an HTML form for the insertion of a new paradigm.

When the template is created, the lexicographer can select a word for which he want to enter the inflected forms, say the Spanish adjective *gordo*. The system will then use the Spanish adjectival template in figure 1 to display an HTML form with a text box for each of the four individual forms, which the lexicographer is asked to fill in.

After the form has been submitted, the system will establish the longest sequence of characters that remains invariant throughout all the forms that were entered (the root), and define the root-creation rule necessary to create the root from the citation form. Once the root is established, the system will define which inflection rules are need to generate the inflected forms from the root.

To verify no errors were made, the result is then shown to the lexicographer, with the root in normal type, and the "affixes" in bold face. The result for the Spanish adjective *gordo* (fat) is shown in figure 2. Only after the lexicographer confirms that the paradigm is correct will the system store the new paradigm ADJ01 in the system.

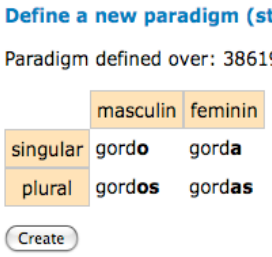

Figure 2: Paradigm definition *gordo* (fat)

While looking for the "root", the system looks for characters, without relying on any type of linguistic knowledge. This makes the process very language independent, and the process works for Spanish just as well as it does languages with another alphabet (like Russian) or languages that have prefixing or circumfixing inflection rules. However, there are paradigms it cannot handle, notably those cases where inflection is not taking place at the beginning and/or the end of the word (see section 5.2).

As mentioned in the previous section, paradigms may contain root alternation rules, and whether or not to use root alternation rules is largely a matter of choice. From only one example, it is impossible for the system to assess what the intended root is, nor what the intended root alternation rules would be. Rather than asking for a large set of examples, the system allows the lexicographer to define the root alternation rules by hand. To facilitate the definition of root-alternation rules, the rules are defined not directly in terms of regular expressions, by in terms of a "from" and a "to" part, separated by a slash, and apply to the end of the root.

To define an alternating pattern for the word *actriz* in Spanish, the lexicographer enters all the inflected forms, plus the (simple) rule *z/c*. With this information, the system automatically compiles all the necessary rules from the example provide, just as in the case without root alternation.

Although simple cases of root alternation rules are easy to define, alternation rules can become rather complex. A somewhat complex rule was given in (4), repeated below, but the rules become even more complex when the root alternation rule adds or removes and accent. Although it would be desirable to facilitate the automatic creation of root alternation rules, root alternation rules currently have to be entered manually. As such, root alternation is the only place in the system where some computational knowledge is required from the part of the lexicographer, but only in cases where the lexicographer wants to use complex root alternation rules.

(4) ($alt_root[1] = $root) =~
s/([aeou])\1([dgklmnprt])$/\1\2/;

3.3 Recognizing Paradigms

When a complete set of paradigms has been defined, it is possible to use the paradigm system to inflect any word of the language: by assigning a word its appropriate paradigm number, all the inflected forms are implicitly defined. However, assigning the paradigm number by hand is a tedious task, and therefore the system should be able to assign the correct paradigm automatically. That is to say, the system should recognize for any new word what the appropriate paradigm is.

In the OSLIN system, this is computationally implemented by constraints that indicate which paradigms are *not* appropriate, until only the correct one(s) remain(s). There are three types of constraints: hard constraints, blocking constraints, and soft constraints.

Hard constraints define which characteristics the word has to have in order for the paradigm to apply. For instance, in Spanish, there is a paradigm for verbs like *actuar* (to act)*:* these verbs get an accent on the *–u–* in, for instance, the first person present indicative: *(yo) actúo*, to make it clear that the accent is on the *–u–* and not on the *–a–* or the *–o–*. This paradigm applies *only* to verbs on *–uar,* which means that the ending *–uar* is a hard constraint for this paradigm.

Blocking constraints state that one paradigm blocks another: in order to for a Spanish noun to inflect like *casa* (house), it has to be a regular noun, which means it should not follow one of the more restrictive paradigms such as the paradigm for words on a *–z* like *actriz*. Words that fall under the paradigm *actriz* do not inflect like *casa*. So meeting the constraints for the paradigm *actriz* means that the word *cannot* inflect like *casa.*

When properly set-up, the combination of hard constraints and blocking constraints define a complete paradigm system, in which the system will only suggest possible paradigms. There can be more than one paradigm suggested though: for instance in the case of a new word on *–olar* in Spanish, there are no formal clues whether the new word should have *–olo* as its first person singular present indicative, like *molar*, or rather *–uelo*, like *volar*. In such cases, it will be up to the lexicographer to choose between the various possible paradigms. In most cases, however, the majority of words will have only one applicable paradigm, which means that the lexicographer will only have to intervene in a limited amount of cases.

Although a paradigm system with only hard constraints and blocking constraints works, it does not always lead to efficient paradigm recognition. An example is the case of invariable nouns in for example Catalan. There are no hard constraints a word needs to meet in order to be invariable: although most of the invariable words end in *–s*, there are quite a few examples of invariables nouns with other endings, especially when counting loanwords as well. Without constraints, the invariable nominal paradigm will be applicable to any new noun in Catalan, and the lexicographer would have to state for every noun that it is not invariable, which is far from efficient. To solve this, it is possible to define soft constraints. Soft constraints work like hard constraints, except that they can be overruled. In the case of Catalan invariable nouns, we can define a soft constraint that such words should end in an *–s*. With such a soft constraint, the system will by default ignore this paradigm for all other nouns, except when explicitly asked to show all available paradigms.

4. The Paradigm Manager in Action

The OSLIN paradigm manager is a fully implemented inflectional system that has been used to generate and manage the full-form lexicon of a variety of languages. When creating a new full-form lexicon, the manager can be used in two different ways, depending on what is available to start from: the system can either be used to inflect a word-list from scratch, or to create a paradigm system from an already inflected wordlist, and then use the system created from this already inflected word-list to inflect words added to the list afterwards. This is useful, for instance, in cases in which a small inflected lexicon is available, and a (much) larger inflected lexicon is needed. Given that the second option is easier to apply, it will be describe first.

4.1 From an Inflected Lexicon

When starting from an already inflected lexicon (however small), the creation of a paradigm system is relatively straightforward. To start the process, it is necessary to load the inflected data into the OSLIN database system. This means that the lexical entries have to be loaded into the table of lexical entries, and the inflected forms into the table of word-forms. Furthermore, a template has to be defined (see section 3.2) using the same codes for the inflected forms as used in the original lexicon.

Once the inflected forms have been added to the system, the lexicographer can start setting-up the paradigm system for each of the major word classes in turn, starting for instance with the adjectives. The process is simple: start by selecting the most regular word you can think of, say the Spanish adjective *gordo*, and ask the system to generate a paradigm for it. The system will look-up all the inflected forms of the adjective *gordo* in the database, and determine the longest sequence of letters common to all of them, in this case *–gord–*. It will then suggest a paradigm as was shown in figure 1, where *–o, –a, –os,* and *–as* are the inflectional suffixes, and the root is created from the citation form by removing the last *–o*.

After creating a paradigm, the system will look through all the adjectives in the database that do not yet have a paradigm assigned to it, and check whether they conform to this newly created paradigm. It does this by applying the paradigm to the citation form of the adjectives (one by one), to generate all the inflected forms that the adjective would have if it inflected via this paradigm. The system then verifies if all forms generated by the paradigm are identical to the forms in the database for the adjective in question. If all the forms match, the word does belong to the paradigm, and the system will automatically assign that adjective the paradigm of *gordo*. To see the progress, it is possible to run this process in verbose mode, in which case the system will also indicate for words that do not match the paradigm, what is the first word-form where the generated forms and the stored forms diverge.

After checking all the adjectives, a lot of them will have been assigned the paradigm *gordo*, yet there will still be a large number of adjectives without a paradigm. The next step is to look at the list of adjectives that do not have a paradigm yet, and select one of them, say *grande* (big), and repeat the process: create a paradigm for it, and have the system mark all adjectives that correspond to the newly created paradigm *grande*. And continue this until all adjectives have been assigned a paradigm. Once all adjectives have a paradigm assigned to them, the paradigm system (for adjectives) is complete.

To help in selecting the next paradigm to create, the system can indicate why words that do not yet have a paradigm assigned to them do not belong to any of the existing paradigms. In figure 3, this is shown for the Catalan adjective *pobre* (poor) after a few paradigms have already been created. In this figure, there are two applicable paradigms, and both are compared to the known forms of *pobre*. The forms in green are those for which the form predicted by the paradigm matches the form stored in the database: the paradigm would (correctly) create the female plural form *pobres* if *pobre* would be assigned the paradigm *asocial* (asocial). The forms in bold red are those where the two forms diverge: if *pobre* would inflect like *asocial*, the female singular form would be *pobre*, whereas it is in fact *pobra*. Therefore, *pobre* in Catalan does not inflect like *asocial,* nor does it inflect like *beix* (beige).

In the comparison in figure 3, the system by default only displays applicable paradigms. Therefore, the Catalan paradigm *blanc* is not shown, since *pobre* violates the hard contraint that all adjective of the paradigm *blanc* have to end on a *–c*. If so desired, it is possible to have the system show all paradigms with violating constraints as well.

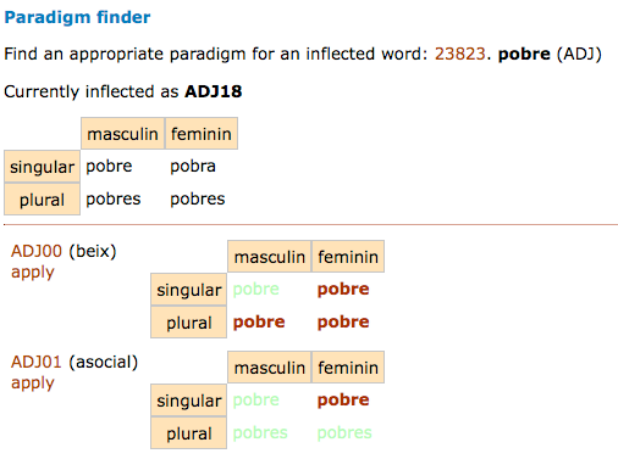

Figure 3: Testing an inflected word

When starting from an inflected lexicon, creating a paradigm system in this way is a rapid process: with each new paradigm, the list of words without a paradigm gets shorter, and it is easy to see which paradigms are still missing. Somewhat more complex is defining the set of constraints to avoid the system from suggesting inappropriate paradigms, but given that for each paradigm, a list of examples will be at hand, it is easy to see which characteristics all of the words of a paradigm share.

When all the paradigms of the language have been defined, the list of words without a paradigm should contain only words that do not inflect like anything else in the language, most of which will be loanwords. Yet if the original database contained errors, a large percentage of the words that were incorrectly inflected in the original database will also remain on that list. This means that applying the paradigm manager to an already inflected lexicon is a quick way to detect errors in a full-form lexicon.

4.2 From a Word List

It is not always possible to start from an inflected lexicon, since an inflected lexicon is not always at hand. Therefore, it is also possible to create a full-form lexicon with OSLIN from only a list of words with their word-classes, provided for instance by a dictionary. The word list should be provided in that case as a simple spreadsheet with two columns: the first containing the citation form and the second the word-class the word belongs to. The system will then help to gradually assign a paradigm to each of the words on the list, and fill the OSLIN tables with the lexical entries and word-forms based on these paradigms.

Without inflected examples, the inflection has to be done interactively, working from the most regular paradigms to the most restrictive ones, and then gradually working back to the regular paradigms. How this process works is illustrated here for Spanish nouns.

Looking at Spanish nouns, the most common plural is the word with a *–s* placed at the end, as in the case of *casa* (house). Since the system does not know the plural of *casa*, we have to add it as a new word, and manually add the singular and plural form. After the word with its inflected forms has been added manually, it can be used to create a paradigm as described in the previous section.

Once the paradigm is in place, the system will display all nouns that could potentially belong to this paradigm; since there are no restrictions (yet), that will be the complete list of all nouns. Looking through that list, there are obvious words that do no inflect like *casa*. For instance, in words ending in a consonant, the plural *–s* is not added directly to the singular, but a linking vowel *–e–* is inserted: the plural of *afinidad* (affinity) is *afinidades* and not *afinidads*.

For each such "exceptionally inflecting" class, a paradigm has to be created, with the restriction that apply to that class. In this case by manually inflecting *afinidad* and then creating a paradigm from it for words ending in a consonant, and then verifying if the all the words on the more restrictive list with candidates for the new paradigm inflect indeed with that paradigm, and otherwise repeat the process. In the case of nouns ending in a consonant, nouns ending in a *–z* form an exception, since they get an orthographic root alternation in their plural form: *actriz/actices* (actress) and not *actrizes*.

The paradigm for *actriz* is restrictive enough to apply to (virtually) all nouns on *–z*. Once such a restrictive paradigm is reached, the system can be asked to inflect all words matching the requirements according to that paradigm. That is to say, we can ask the system to inflect all nouns ending on a *–z* in our wordlist via the paradigm *actriz*.

Once the nouns on *–z* are taken care of, it is necessary to return to the more general paradigm (*afinidad*) to verify if there are more exceptions. Once all exceptions to the paradigm *afinidad* have been taken care of, the paradigm *afinidad* can be applied to all remaining nouns ending on a consonant. Once all words ending on a consonant have been inflected, it is time to return to the remaining list of nouns to see if there are other classes of nouns that do not end in a consonant, yet do not inflect like *casa* either, until finally, all remaining nouns can be inflected like *casa*.

Even a restrictive paradigm like *actriz* is not fully without exceptions, although in this case there is only one exception in the *Diccionario de la Lengua Española* (RAE 2001): the word *kibutz* (kibutz) is a foreign loanword and does not change in the plural. When spotting the exception before inflecting all the words on *–z*, it can be inflected by hand, which will mean it will not receive the paradigm *actriz* since it has already been inflected. If it is not spotted before, since there are hundreds of words on *–z* that do follow the paradigm it is

easy to overlook, it can always be changed afterwards (see 4.3).

It can happen that a whole paradigm is overlooked, meaning that a class of words got inflected via the wrong paradigm in the process described above. For instance, it is easy to overlook the paradigm for *virgen* (virgin), which receives an accent in the plural: *vírgenes*. When the words of this paradigm have already incorrectly been inflected via the paradigm *afinidad*, this can be corrected by manually correcting the inflected forms for *virgen*, and subsequently create a paradigm out of the corrected inflection, which can then be applied to all words on *–en*.

Using this strategy, we have managed to create reliable, full-forms lexica with around 50.000 to 100.000 lexical entries (over a million inflected forms) for several languages in a relatively small amount of time, with an estimate of around 500 man-hours.

4.3 Post-Verification and Maintenance

Even when created with the utmost care, a large-scale lexicon with over a million word-forms is never fully correct. Therefore, it will be necessary to correct errors after the original creation of the database. The OSLIN administration environment is not built as a tool usable only for the creation of a full-form lexicon, but as a management tool for the creation and continuous maintenance of lexical resources.

The OSLIN tools easily allow to choose a different paradigm for an already inflected word to correct a wrongly inflected word, or to change the inflected forms manually if it does not belong to any paradigm. The problem is to find errors in a database that large. Using external resources such as traditional grammar books and existing dictionary helps in finding words that are known to have an exceptional inflection, and therefore are the most likely to have gone wrong in the semi-automatic inflection process. But the OSLIN databases are built to have an alternative way of finding and correcting errors: improvement by use.

The OSLIN resources are not intended to be passive word-list, but rather lexical resources to be actively used. The database is set-up to be used as the exclusion lexicon for neologism research, and the system comes with integrated tools for use as a part-of-speech tagger and a spelling checker. The part-of-speech tagger can report on words that look like known words that are inflected differently in the corpus than in the lexicon. It does this by automatically lemmatizing unknown words, and then looking for words with a known citation form and word-class, but an unknown inflected form.

Furthermore, the data of the OSLIN lexica are directly available online in a user-oriented web site with rich search capabilities. Each page showing the inflected forms of a word has a “report” button on it, which allows the users to provide feedback on errors in the database (although the report function can be disabled for a language when it is not desired).

Most of the feedback coming from the tagger or the online users is not an indication of an error in the database, but rather mistakes by the users or the authors of texts in the corpus. However, the occasional error in the database is likely to be found by these methods over time.

4.4 Less-Resources Languages

As explained earlier, the OSLIN paradigm system can be used to inflect a lexicon for a large variety of languages, since there is nothing language-specific in its design. And furthermore, the system can be set-up and used by a lexicographic team, with only a minimum amount of external help, and without the need for trained computational linguists.

These characteristics make the OSLIN paradigm inflection system very well suited for use with smaller languages for which fewer resources exist. For less-resourced languages, lexicographic sources and lexicographers often are available, but finding computational linguists to work out the inflectional system of the language is more problematic. With the OSLIN tools, it is possible for lexicographers to create and maintain a reliable, large-scale lexicon for such languages, using a framework that furthermore facilitates the creation of the tools mentioned in the previous section: a part-of-speech tagger, a spelling checker, a neologism detection tool, and an online language consultation site.

5. Issues

The use of paradigms is a powerful and intuitive way to treat inflection. However, there are cases where the use of paradigms for inflection raises problems. This section describe three cases in which issues with the use of a paradigms in a semi-automatic detection tool arise, and sketches how these issues are, or can be dealt with in the OSLIN framework.

5.1 Computer vs. Human Paradigms

As mentioned before, what humans consider to be words that inflect the same does not always correspond to what a computational system would do. The root alternation rules bring the two closer together, but that does not in all cases conflate the two. Below are some cases where mismatches remain, although most of them can be overcome.

For several languages, traditional grammars include paradigms that are computationally speaking fully redundant. For instance, the *Normes Ortográfiques* for Asturian (ALLA 2005) includes a paradigm for *panaderu* (baker), even though it inflect fully regularly like *llobu* (wolf). The paradigm is included for the sake

of clarity, and not to indicate an irregularly inflecting group. It is possible to define redundant paradigms in the OSLIN paradigm system, but they have to be forced upon the system, since computationally, there is no reason for their existence. However, the system will not always be able to determine which of the paradigms is the intended one in a given word.

It is common to say that certain words follow more than one paradigm: the Dutch word *leraar* (teacher) is often said to inflect both like *blaar/blaren* (blister) and like *makelaar/makelaars* (house broker). Although it is not impossible to implement this computationally, OSLIN follows the more straightforward method to define a third paradigm for *leraar* in such cases, which is a paradigm that allows both plurals.

Computationally, there is a class of nouns in Portuguese that (at least in some sources) have a *–y* in the singular and *–ies* in their plural: *husky, caddy, body,* etc. These are all (English) loanwords since, until recently, the *y* was not even part of the Portuguese alphabet. Lexicographically, it looks odd to say that there is a paradigm for *husky* in Portuguese. For such cases, it is possible in OSLIN to not assign a word a paradigm at all, but only provide it with manually entered inflected forms.

The most problematic case of mismatch are those cases where for a human, two words clearly inflect the same, whereas for a computer, they do not. An example is the Catalan verb *prevenir* (to prevent). It is a prefixed version of the verb *venir* (to come), and hence inflects like it, as do *sobrevenir* (to overcome) and several other verbs. The third person singular present indicative of *prevenir* is *prevé*, with a accent on the last *é* to indicate the stressed syllable. This stress mark is present in all prefixed verbs from *venir* but it is *not* present in the verb *venir* itself. Since the same form for *venir* (*ve*) is monosyllabic, there is no need for the stress mark. Although it is not fully impossible to define a set of transformation rules that correctly inflect both *prevenir* and *venir*, it is very awkward at best, and definitely not something that can be achieved automatically, or manually by someone without sufficient computational know-how.

5.2 Compounds and Paradigms

In a paradigm-based framework, especially one using string-transformation rules as in the case of OSLIN, inflection is mostly taking place at the beginning and/or the end of the word. For that reason, words where inflection does take place in the middle of the word are problematic.

Infixing inflection is for simple words is not common, but it is much more common to find word-internal inflection in the case of compounds. For instance, hyphenated nominal compounds in Portuguese can pluralize on the left, on the right, or on both part: the plural of *guarda-chuva* (umbrella) is *guarda-chuvas,* whereas the plural of *guarda-nocturno* (night guard) is *guardas-nocturnos*. The same holds for multi-word expressions in English.

In such cases, it would be possible to use a string-transformation rule to place an *–s–* before the hyphen in the paradigm. But the problem with that solution is that if the compound is left-inflecting, it does not necessarily pluralize with an *s*, but can pluralize like any normal noun. Therefore, the OSLIN system can assigns such compounds two paradigms: one for the left part, and one for the right part. For instance, the paradigm SUB01[-]SUB01 can be used for the case of *guarda-nocturno*: it indicates a nominal compound, which inflects on the left and the right, where the two parts are separated by a hyphen, and that the left part inflects via the first nominal paradigm, and the right part via the first nominal paradigm as well.

However, the solution of multiple paradigms relies on the fact that there is a graphical indication what the left and the right part of the compound are. Fortunately, languages have a tendency to avoid left-inflecting compounds where no such indication is present, but they do exist. An example is the Spanish word *hijodalgo* (gentleman) which is morphologically a compound (*hijo de algo* – son of somebody) where the left part is inflecing: *hijosdalgo*. In such cases, it is impossible to automatically determine from the citation form where the plural *s* should be inserted.

In Dutch and German, there is a much larger, well-known class of compounds that are problematic in the same way as left-inflecting non-separated compounds: prefixed separable verbs. The past tense of the Dutch verb *overgeven* (to vomit) is *gaf over*, and the past participle is *overgegeven*. In these two forms, the first component of the verb is separated from the rest, either by displacement, or by the insertion of inflectional material can be inserted between the two parts. Getting the inflected forms correct for separable verbs in a rule-based system is always complicated, but solutions have been implemented in the past (see for instance ten Hacken & Bopp 1998), and these solutions can be implanted in terms over string-transformation rules as well. However, such solutions always rely on a manual indication of the prefix. Although most verbal prefixes are prepositions, there are also verbal compounds with adverbs (*weglopen,* to walk away), or even noun (*brandstichten*, to commit arson), and there is no way to reliably predict which part of the verb will be the prefix.

There are only two solutions in the case of compounds without an explicit indication. The first is to resort to manual inflection for such cases, which is the solution most often used in OSLIN. However, it is possible to manually insert a dummy-separator: by changing the input to the paradigm system from *hijodalgo* to

hijo#dalgo, and from *weglopen* to *weg#lopen*, it becomes possible to use the multiple paradigm assignment for compounds as described above.

5.3 Defectiveness and Clitics

Defective paradigms, such as impersonal verbs, are verbs that lack certain inflected forms. Such verbs can be straightforwardly dealt with in terms of normal paradigms, where the paradigm itself misses a number of forms. There are, however, two problems with such an approach. Firstly, impersonal verbs *can* typically be used in the defective forms when the verb is used metaphorically. And secondly, the forms that do exist can follow any of the existing paradigms. This means that there is not just the need for one additional defective paradigm, but that theoretically, every paradigm would need a defective counterpart.

To solve both problems at the same time, OSLIN uses meta-paradigms: an impersonal verb like *atardecer* (to get dark) is assigned a normal paradigm, in this case it inflects like *crecer* (to grow). On top of that, it is assigned a defective paradigm, which specifies which forms do and do not exist. There can be various defective paradigms per word class if needed.

The defective paradigms make a distinction between two different types of defectiveness. On the one hand, defectiveness due to semantic restrictions, which can typically be overruled in metaphoric uses of the word. Such forms are shown in the web-interface, but grayed out. On the other hand, thre are cases where the defectivity is due to normative considerations, as in the case of the so-called *euphonic defective verbs,* where the defective forms are never (normatively) acceptable, not even in metaphoric use or otherwise. Such forms are stored, but in principle completely hidden in the web-interface.

Not only defective paradigms can be treated by meta-paradigms, but also clitics in the inflection, as for instance in the case of pronominal verbs in Portuguese or Spanish. In a system based on string-transformation, a pronominal verb like *aburrarse* (to get bored) would need a special paradigm, meaning that as in the case of defective verbs, all paradigms would need to be duplicated. In the OSLIN system, the verb *aburrarse* is inflected like *amar*, and a meta-paradigm is used to add the pronominal clitics in the right forms.

6. Conclusion

As shown in this article, it is possible to have a computational tool for the semi-automatic inflection of the lexicon, where the lexicographer has all the freedom he needs to provide high-grade inflectional data, while at the same time being guided and helped along by the computational tool. With the inflectional tools provided by the OSLIN framework, it is possible to generate large full-scale, lexicographically controlled full-form lexicons within a reasonable amount of time.

Because the system is language independent, and furthermore allows lexicographers to create and apply the paradigm system for a new language, the OSLIN paradigm tool is particularly useful for less-resources languages.

Inflection in dictionaries is an often-underestimated topic: it is often considered a trivial task that can easily be achieved by computational means. This article only mentions problems that have to do with the creation of inflected forms by means of an inflectional paradigm. But there are many other problems that are beyond the scope of this article: how to establish what the correct inflected forms are, how to deal with the inflection of loanwords, when to consider a word to have a defective paradigm, etc. Although the OSLIN tools do not by themselves solve any of these issues, they do provide a platform in which the lexicographer has the option to implement his solutions for these issues.

7. References

Baptista Xuriguera, J. (2009). *Els Verbs Conjugats*. Barcelona: Claret.

Academia de la Llingua Asturiana (2005). *Normes Ortográfiques*, ed. 6. Oviedo: ÁPEL.

ten Hacken, P., Bopp, S. (1998). Separable Verbs in a Reusable Morphological Dictionary for German. In: *Proceedings of the 36th annual meeting on Association for Computational Linguistics*, pp. 471-475

Janssen, M. (2005). Open Source Lexical Information Network. In: *Proceedings of the Third International Workshop on Generative Approaches to the Lexicon*. Geneva.

Koskenniemi, K. (1983). *Two Level Morphology: A general computational model for word-form recognition and production*. PhD Thesis, University of Helsinki.

Matthews, P.H. (1972). *Inflectional morphology: A theoretical study based on aspects of Latin verb conjugation*. Cambridge: Cambridge University Press.

Real Academia Española. (2001) *Diccionario de la Lengua Española,* ed. 22. Madrid: Espasa-Calpe.