Collapsing Morphological Information in Lexical Databases for NLP Applications

Juan A. Alonso, Ramón Fanlo, Albert Llorens

Sail Labs S.A.
Roger de Llúria, 50, 1er. – E-08009 Barcelona
Spain
Juan.Alonso@sail-labs.es, Ramon.Fanlo@sail-labs.es, Albert.Llorens@sail-labs.es

Abstract

The morphology of inflectional languages poses specific problems in the processing of morphological alternations. Regular alternations at morpheme boundaries can be elegantly captured by the use of rule formalisms based on the two-level morphology model. Stem alternations and completely irregular alternations at morpheme boundaries, however, need to be captured in some way in the lexicon. This paper presents four possible solutions to the problem and makes a claim in favor of one of them. The proposed approach makes use of feature bundles that contain the necessary linguistic information to uniquely identify allomorphic variations of stems in the lexicon. The proposal is an improvement in that it simplifies the representation of allomorphic variations in the lexicon by avoiding duplication of stem allomorphs to capture cross-combination of several morphosyntactic features in stem+flex sequences.

Keywords

Computational morphology, computational lexicography, morphological alternations, morphographemics, machine translation.

Introduction

Morphological analysis and morphological generation are two tasks shared by many applications in the field of NLP. Closely connected to these tasks, one of the main issues at hand when designing such applications is how to organize and store in the lexicon the morphological information needed to analyze and generate words.

Machine Translation is typically one of the applications that needs to handle analysis and generation processes at the same time. Linguistic databases for MT systems need to be designed so that the knowledge they store is as much process independent as possible. Thus designing declarative lexicon databases in MT applications is a must.

This paper describes work done in Sail Labs in the areas of computational morphology and computational lexicography. The problem addressed is that of morphological alternations in inflectional languages. It is well known that regular alternations at morpheme boundaries can be handled by using morphographemic rules in a two-level morphology model. However, there are still many stem alternations that are completely idiosyncratic and need to be addressed in the lexicon. In the next sections, we will present the problem, the possible solutions, and our proposal.

The Question

Focussing on the Problem

The complexity of morphological processing depends on the type of language being handled. Whereas isolating languages like Chinese or Vietnamese, and languages with very poor inflectional morphology like English present few problems (because there are no morphemes to be concatenated, or there are very few of them), this is not the case with other languages. Again, the approaches to be taken differ depending on the morphological typology of the languages. Agglutinative languages (Turkish, Finnish, Basque, etc.) ask for specific approaches designed to cope with words consisting of one stem plus a (possibly long) sequence of suffixes, whereas inflectional languages (e.g. Semitic Languages[1] and languages belonging to the Indo-European family: Romance languages, Slavic languages, Greek, and to a lesser extend, some Germanic languages) require a different solution.

We are going to focus in this paper on the general case of inflectional languages, and more specifically, on the morphology of the verbal systems in Romance languages.

[1] Most Semitic languages are internal-inflectional languages and they usually need specific morphological approaches which do not apply to other inflectional languages.

The Verbal Morphology of Romance Languages

Morphological verb variation in Romance languages reflects differences in person and number, on the one hand, and tense[2] and mood, on the other. Morphology of non-finite forms is defined separately (infinitive, past participle, present participle, gerund), although the need to tell these forms from the finite ones in the database entries is still questionable[3].

Analytic verb forms are not affected by the problem of how to efficiently handle verb morphology in the coding of entries in a database. A form like "*ha venido*" is simply a compound of a form of the auxiliary verb "*haber*" (present indicative 1st person) and a form of the verb "*venir*" (past participle). This clarification is important because traditional Romance verb-form paradigms include (some[4]) analytic verb forms built on auxiliaries. This is so because they try to classify verb forms into paradigms that are valid for Latin, which has a synthetic verb system. We stray from that view in this paper, which enables a simpler and computationally more adequate use of feature-value information:

"tememos" (present indicative, 1st plural)

VS.

"estamos temiendo" (present indicative, 1st plural + gerund: compound form expressing progressivity)

"hablarán" (future indicative, 3rd plural)

VS

"habrán hablado" (future indicative, 3rd plural + past participle: compound form expressing perfectivity).

In other words, traditional verb-form labels like "future perfect" or "perfect past" are of no use in our approach to verb morphology. In terms of feature-value information, these compound forms are distinguished from the corresponding simple ones just by one additional, boolean feature (e.g. PROGR[essivity], PERF[ectivity]…).[5] [6]

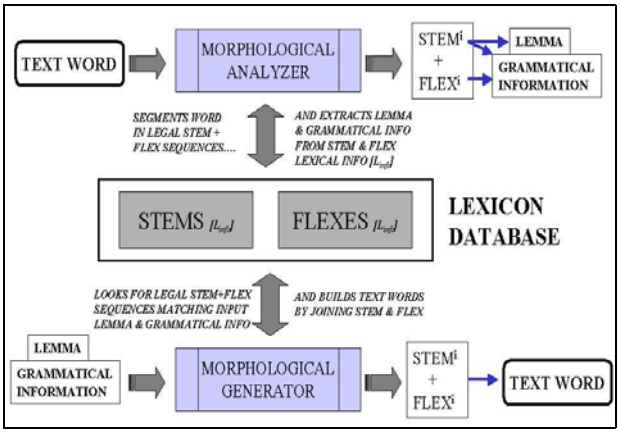
Table of Verbal Forms to be Covered

Predicate Form	Mood	Tense	Number	Person	Form: Example
	INDICATIVE	*PRESENT*	*SINGULAR*	*1*	como
				2	comes
				3	come
			PLURAL	*1*	comemos
				2	coméis
				3	comen
		IMPERFECT	*SINGULAR* …	*1, 2, 3…*	…
		INDEFINITE	*SINGULAR* …	*1, 2, 3…*	…
		FUTURE	*SINGULAR* …	*1, 2, 3…*	…
	CONDITIONAL		*SINGULAR* …	*1, 2, 3…*	…
	SUBJUNCTIVE	*PRESENT*	*SINGULAR* …	*1, 2, 3…*	…
		PAST	*SINGULAR* …	*1, 2, 3…*	…
		FUTURE	*SINGULAR* …	*1, 2, 3…*	…
	IMPERATIVE		*SINGULAR*	*2*	…
			PLURAL	*2*	…
INFINITIVE	X	X	X	X	comer
PAST PARTICIPLE	X	X	X	X	comido/a(s)
PRESENT PARTICIPLE	X	X	X	X	X
GERUND	X	X	X	X	comiendo

The Morphological Processing Scheme

As we said, we are going to restrict the current discussion to the morphological processing of Indo-European inflectional languages, and more specifically, to the verbal systems of Romance languages. The lexical material we are going to deal with will therefore consist of word (verbal) stems on the one hand and word (verbal) inflections on the other[7].

The following figure illustrates the working flow of the two morphological processes at hand, morphological analysis and morphological generation.



In both cases, the software involved (morphological analyzer and morphological generator) accesses the lexical database containing the repository of available stems and flexes. Both stems and flexes have certain

[2] Although perfectivity has also a morphological reflection in the past tense (imperfect "comía" vs. indefinite/simple past tense "comió"), this feature is usually subsumed in the tense classification.

[3] Portuguese "personal infinitives" are non-finite forms that show a special behaviour in that they are marked for person an number.

[4] Usually, traditional verb paradigm descriptions include all perfect forms built on the "haber"-like auxiliary ("he venido", "había venido", "habré venido"). We wonder why progressive forms ("estoy viniendo", "estaba viniendo", "estará viniendo") or their possible combinations ("he estado viniendo", "había estado viniendo") are never mentioned.

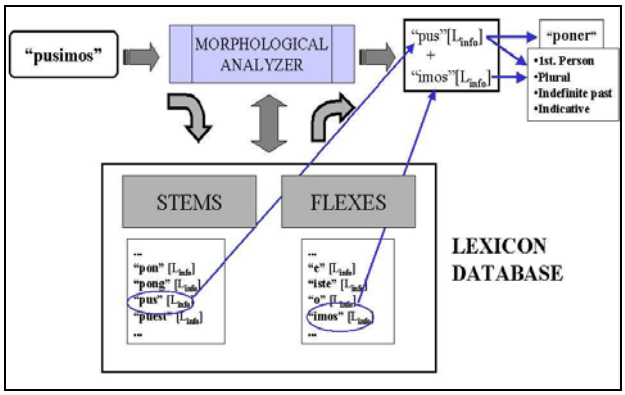
[5] Other languages show the same with future (English auxiliary "will" + infinitive). Therefore, English future tense forms should not be included in a lexicon database.

[6] An implication of this is that "tense" is strictly bound to morphosyntax, not to real-time expression. A form like "he venido" (which expresses past real-time) is morphosyntactically a perfect present.

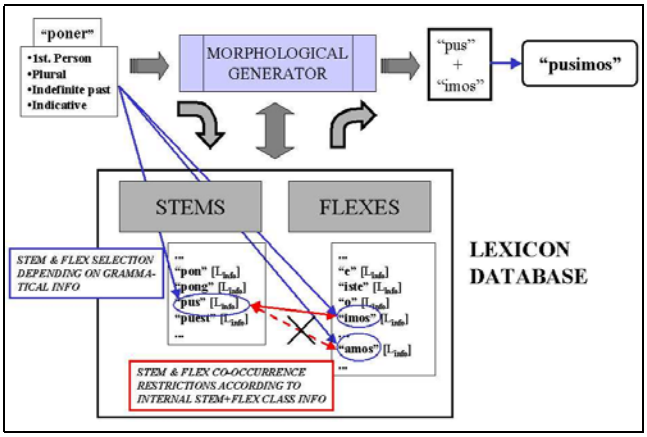
[7] Obviously, the present discussion also applies to similar cases of complex nominal or adjectival morphology (cf. nominal paradigms in Slavic languages or Greek)

lexical information [L_{info}] associated. This lexical information has to meet at least two conditions:

1. The relevant morphosyntactic features (verb lemma, person, tense, mood, number, etc.) are either directly conveyed by [L_{info}] or can be automatically calculated from it by some computational machinery. Thus, the lexical information [L_{info}] stored in the stem "***pus***" together with the information stored in the flex "***imos***" should be enough for the Morphological Analyzer to give as the output for the Spanish verb form "*pusimos*" the lemma "*poner*" plus the associated information of this form being a "*1st. person plural indefinite past indicative*".



2. The [L_{info}] stored for each stem and flex must also be enough to ensure that only legal stem+flex sequences are retrieved from the lexical database. So, when generating the right verbal form corresponding to the "*1st person plural indefinite past indicative*" (grammatical information) of the Spanish verb "*poner*" (lemma), the stored [L_{info}] should be enough for the Morphological Generator not only to select the stem and the flexes matching this information but also to ensure that the right flex "*imos*" is attached to the stem "*pus*", yielding the right form "*pusimos*" and blocking the wrong form *"*pusamos*". This type of "co-occurrence" information is typically represented in form of an inflection class feature ("CL", in our system) whose values usually correspond to the different inflectional models.



We will assume that both grammatical information and lexical information are represented and stored in the form of feature-value pairs (e.g. "first person singular indicative present" = [(PS 1) (NU SG) (TN PR) (MD IND)]

The aim of this paper is to present the possible approaches for organizing verbal stems and flexes and designing their associated lexical information in the lexical database.

Possible Approaches

Storing Full Forms

Description of the Approach

The first obvious "brute-force" approach is to store in the lexicon all the possible inflected forms for every word in the language. Thus, for a Spanish verb "poner", the lexicon would include (at least) all 54 possible inflected forms for this verb, together with their associated lexical information ("*pongo*", "*pones*", … "*ponía*", …"*puse*", …"*pondríamos*", …"*poniendo*", …).

In this approach, the lexical database would contain full word forms, with no stems or flexes whatsoever. Each full form in the lexical database would have associated the corresponding lexical information [L_{info}] in the form of linguistically meaningful feature-value pairs:

LEXICAL DATABASE
...
"pongo" = (LEMMA "poner") (PS 1) (NU SG) (TN PR) (MD IND)
"pones" = (LEMMA "poner") (PS 2) (NU SG) (TN PR) (MD IND)
"pone" = (LEMMA "poner") (PS 3) (NU SG) (TN PR) (MD IND)
"ponemos" = (LEMMA "poner") (PS 1) (NU PL) (TN PR) (MD IND)
"ponéis" = (LEMMA "poner") (PS 2) (NU PL) (TN PR) (MD IND)
"ponen" = (LEMMA "poner") (PS 3) (NU PL) (TN PR) (MD IND)
...
"puse" = (LEMMA "poner") (PS 1) (NU SG) (TN INDF) (MD IND)
....

Advantages of the Approach

This approach has two main advantages:

- The computational machinery needed to access the lexicon is kept to a minimum.
- The process of morphological analysis/generation is very fast.

Disadvantages of the Approach

- A huge amount of disk and memory space is required to store the lexicon.
- An enormous coding effort would be required to enter all the forms in the lexicon, unless some "intelligent" coding tool was used for this task[8].
- There would be serious maintenance problems for lexicographers to change and/or add new entries.
- The handling of deviant forms with proclitic elements ("*vámonos*", "*dámelo*", etc.) would pose a serious problem. For instance, Spanish has three types of such cases:
  - Loss of the stem final consonant: "*amad*" ("love!") → "*amaos*" ("love each other!")
  - Suprasegmental changes (graphic accent): "*da*" ("give") → "*dámelo*" ("give it to me")
  - Both of the above: "*vamos*" ("we go") → "*vámonos*" ("let us go")

Automatic Handling

Description of the Approach

In this approach, the lexical information associated to the stems and flexes in the lexicon database is kept to a minimum, and the bulk of the stem/flex selection task is left to the application software. The following table illustrates the contents of a typical lexicon database built according to this approach:

LEXICAL DATABASE	
STEMS	**FLEXES**
...	...
"pon" (CL X17)	"o" (CL X10 X15 X17) (PS 1) (NU SG) (TN PR) (MD IND)
"pong" (CL X17)	"es" (CL X10 X15 X17) (PS 2) (NU SG) (TN PR) (MD IND)
"pus" (CL X35)	"e" (CL X10 X15 X17) (PS 2) (NU SG) (TN PR) (MD IND)
"pond" (CL X42)	"emos" (CL X10 X15 X17) (PS 2) (NU SG) (TN PR) (MD IND)
"puest" (CL X70)	"éis" (CL X10 X15 X17) (PS 2) (NU SG) (TN PR) (MD IND)
...	"en" (CL X10 X15 X17) (PS 2) (NU SG) (TN PR) (MD IND)
	...
	"e" (CL X30 X35) (PS 1) (NU SG) (TN INDF) (MD IND)
	"iste" (CL X30 X35) (PS 2) (NU SG) (TN INDF) (MD IND)
	"e" (CL X30 X35) (PS 3) (NU SG) (TN INDF) (MD IND)
	...

[8] In which case the intelligent tool itself would have to incorporate the knowledge to generate all the verbal forms for at least the regular verbs.

In this scheme, the application software contains built-in knowledge about the grammatical contexts where each stem/flex can be selected (e.g., it "knows" that the value X17 for CL corresponds to "*second/third person singular and all persons plural present indicative*").

Advantages of the Approach

- The lexical information of stems is kept to a minimum (typically, an inflectional class code)

Disadvantages of the Approach

- The linguistic knowledge stored in the lexicon is extremely opaque. The lexicographers that code and maintain the lexicon only have a list of class codes available, often devoid of any overt linguistic meaning.
- The two processes usually involved in morphological generation (stem selection and inflection selection) become more complex. Backtracking may be necessary if stem selection is ruled out by inflection selection. In some cases, this process may even lead to incorrect word form generation.

Using Morphosyntactic Features as Lexical Information

Description of the Approach

A third solution is to use explicit grammatical features as the lexical information associated to the stems and flexes of the lexical database. In this way, the output of the morphological analysis is the grammatical information obtained from the unification of the lexical information associated to the selected stem and its corresponding class-matching flex. For example "*pusimos*" would be segmented into the stem "***pus***"[**(LEM "poner") (CL INDF-ST-2) (PS 1 2 3) (NU SG PL) (TN INDF) (MD IND SUB) (PF FIN)**] plus the flex "***imos***" [**(CL INDF-ST-1 INDF-ST-2 ER IR) (PS 1) (NU PL) (TN PR INDF) (MD IND) (PF FIN)**], yielding the output ***[(LEM "poner") (PS 1) (NU PL) (TN INDF) (MD IND) (PF FIN)].*** No mapping is needed to convert lexical information into grammatical information.

The following table shows a sample of the contents of a lexical database coded according to this scheme.

LEXICAL DATABASE	
STEMS	**FLEXES**
...	...
"pong" (CL ER) (PS 1) (NU SG) (TN PR) (MD IND) (PFORM FIN) (LEM "poner")	"o" (CL AR ER IR) (PS 1) (NU SG) (TN PR) (MD IND) (PFORM FIN)
"pong" (CL ER) (PS 1) (NU SG PL) (TN PR) (MD SUB) (PFORM FIN) (LEM "poner")	"es" (CL ER IR) (PS 2) (NU SG) (TN PR) (MD IND) (PFORM FIN)
"pon" (CL ER) (PS 2 3) (NU SG) (TN PR) (MD IND) (PFORM FIN) (LEM "poner")	"e" (CL ER IR) (PS 2) (NU SG) (TN PR) (MD IND) (PFORM FIN)
"pon" (CL ER) (PS 1 2 3) (NU PL) (TN PR) (MD IND) (PFORM FIN) (LEM "poner")	"emos" (CL ER) (PS 2) (NU SG) (TN PR) (MD IND) (PFORM FIN)
"pon" (CL ER) (PS 1 2 3) (NU SG PL) (TN IMP) (MD IND) (PFORM FIN) (LEM "poner")	"éis" (CL ER) (PS 2) (NU SG) (TN PR) (MD IND) (PFORM FIN)
"pon" (PFORM INF GER) (LEM "poner")	"en" (CL ER IR) (PS 2) (NU SG) (TN PR) (MD IND) (PFORM FIN)
"pus" (CL INDF-ST-2) (PS 1 2 3) (NU SG PL) (TN INDF) (MD IND SUB) (PFORM FIN) (LEM "poner")	...
"pondr" (CL ER) (PS 1 2 3) (NU SG PL) (TN FUT) (MD IND) (PFORM FIN) (LEM "poner")	"e" (CL INDF-ST-1 INDF-ST-2) (PS 1) (NU SG) (TN INDF) (MD IND) (PFORM FIN)
"puest" (PFORM PAPL) (LEM "poner")	"iste" (CL ER IR INDF-ST-1 INDF-ST-2) (PS 2) (NU SG) (TN INDF) (MD IND) (PFORM FIN)
...	"o" (CL INDF-ST-1 INDF-ST-2) (PS 3) (NU SG) (TN INDF) (MD IND) (PFORM FIN)
	...
	"ieron" (CL ER IR INDF-ST-2) (PS 3) (NU SG) (TN INDF) (MD IND) (PFORM FIN)
	...

Advantages of the Approach

- The lexical information is linguistically overt nad meaningful for lexicographers.
- No conversion from morphological to grammatical features is required for grammar processing.

Disadvantages of the Approach

The main problem of this approach is that when coding irregular verbs, whenever a verb stem is shared by all of the persons in one of the two numbers (say for example by all the plural persons) **and** by some but not all of the persons of the other number (say for example by 2nd and 3rd singular, but not by 1st) in a given tense-mood combination, then the same verb stem has to be coded twice with person and number values not only different but also, and necessarily, mutually exclusive.

An example of this is the present indicative forms of the Spanish verb "poner":

Present Indicative	SINGULAR	PLURAL
1st person	pong-o	pon-emos
2nd person	pon-es	pon-éis
3rd person	pon-e	pon-en

Given the need for a stem "***pong***" to be marked as [**(PS 1) (NU SG)** (TN PR) (MD IND)], a second stem "***pon***" is needed for [**(PS 2 3) (NU SG)** (TN PR) (MD IND)] and still a third stem "***pon***" is needed for [**(PS 1 2 3) (NU PL)** (TN PR) (MD IND)]. The wrong, non-existent form *"pon_o", as [(PS 1) (NU SG) (TN PR) (MD IND)], would be both analyzed and generated if one single stem "***pon***" was coded, because it should have all values for number (NU) and person (PS) with no possibility of ruling out the combination 1st person singular. The situation becomes worse as the stem "***pon***" is also needed, with all possible person and number combinations, for other tense and mood combinations (e.g. the imperfect: "pon-ía", "pon-ías", etc.).

To make things even more complicated, the stem "***pong***" is in turn also needed for all subjunctive present forms, in all persons and numbers:

Present Subjunctive	SINGULAR	PLURAL
1st person	pong-a	pong-amos
2nd person	pong-as	pong-áis
3rd person	pong-a	pong-an

Therefore, the same problem arises whenever a verb stem is shared by <u>all</u> persons and numbers in a given tense and mood combination (like all the present-subjunctive forms with "***pong***") and by <u>some but not all</u> of the person and number combinations of another mood and tense combination (like the 1st singular of present indicative "***pong***").

This multiplication of identical stems in the lexicon is a disadvantage in terms or space efficiency and ease of coding and maintenance.

It is important to note that, in this approach, the introduction of lexical features indicating "transformations of allomorph" (TAL[9]) would effectively reduce the coding problem. However, our experience so far shows that:

- Lexical coding of TAL can only be used for "regular irregularities", i.e. for very productive stem alternations as for example "o"/"ue" in Spanish verbs like "encontrar". It would not be adequate to define a TAL value for a completely idiosyncratic alternation as that of "pus"/"pon for the lemma "poner". For these cases the storage of multiple stems in the lexicon is still needed.
- Although lexical coding of TAL may be of undoubted interest, it means having non-declarative morphological knowledge in the lexicon database. This can for sure be regarded as a design flaw in a lexicon database for NLP applications[10].

Using Mixed Feature Bundles as Lexical Information

Description of the Approach

This final approach avoids the problem of the duplicate stem coding described in the previous section. The basic idea is to use, as linguistic information, features representing possible tense+mood combinations in the verbal inflection paradigm, and taking as values the person+number combinations where the corresponding stem and flex apply. The following table reflects the possible values for person+number:

VALUE	MEANING
1	1st person singular
2	2nd person singular
3	3rd person singular
4	1st person plural
5	2nd person plural
6	3rd person plural
0	all persons in all numbers

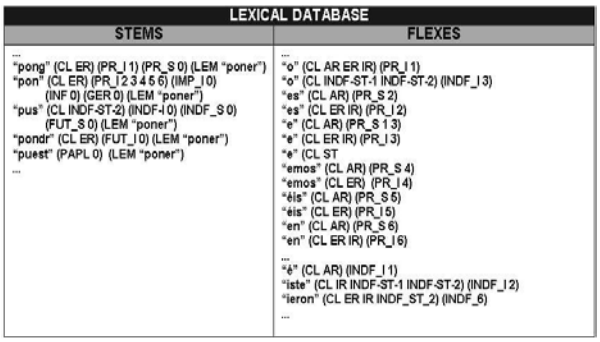
Thus, for example:

(PR_I 1 2 3): "all singular persons in the present indicative"
(IMP_I 0): "all persons in the imperfect past indicative"
(INDF_S 4): "1st person plural in the indefinite past subjunctive"

The following table shows a sample of the contents of a lexical database coded according to this scheme.

[9] TAL coding in the lexicon captures "regular" idiosyncratic allomorph alternations that cannot be captured by a pure two-level morphology model.

[10] Especially in analysis, TAL coding may easily lead to over-recognition.



LEXICAL DATABASE	
STEMS	**FLEXES**
... "pong" (CL ER) (PR_I 1) (PR_S 0) (LEM "poner") "pon" (CL ER) (PR_I 2 3 4 5 6) (IMP_I 0) (INF 0) (GER 0) (LEM "poner") "pus" (CL INDF-ST-2) (INDF-I 0) (INDF_S 0) (FUT_S 0) (LEM "poner") "pondr" (CL ER) (FUT_I 0) (LEM "poner") "puest" (PAPL 0) (LEM "poner") ...	... "o" (CL AR ER IR) (PR_I 1) "o" (CL INDF-ST-1 INDF-ST-2) (INDF_I 3) "es" (CL AR) (PR_S 2) "es" (CL ER IR) (PR_I 2) "e" (CL AR) (PR_S 1 3) "e" (CL ER IR) (PR_I 3) "e" (CL ST "emos" (CL AR) (PR_S 4) "emos" (CL ER) (PR_I 4) "éis" (CL AR) (PR_S 5) "éis" (CL ER) (PR_I 5) "en" (CL AR) (PR_S 6) "en" (CL ER IR) (PR_I 6) ... "é" (CL AR) (INDF_I 1) "iste" (CL IR INDF-ST-1 INDF-ST-2) (INDF_I 2) "ieron" (CL ER IR INDF_ST_2) (INDF_6) ...

Advantages of the Approach

The main advantage of this approach is that, while preserving the transparency of the linguistic information associated to stems and flexes, it avoids having to code several times the same stem for different tense+mood/ person+number combinations.

Thus, the irregular Spanish verb "*poner*", which has four different inflectional stems ("*pon*", "*pong*", "*pus*" and "*puest*") only needs four entries with this approach (i.e., one entry per stem) while it needed nine entries with the previous solution.

This approach is also compatible with the TAL coding (described above), which provides a means of handling regular idiosyncrasies in the lexicon.

Disadvantages of the Approach

This approach requires that either the morphological or the grammatical processors include some machinery that converts morphological feature bundles into grammatical features (person, number, tense and mood). This requirement, however, cannot be regarded as a true disadvantage.

The main problem with the feature bundles as they have been presented, is that they cannot be arranged in a clear hierarchy, as morphosyntactic features do. Feature bundles can only be arranged in a flat structure, which may lead to the need to use several feature bundles to make two entries mutually exclusive, where one single morphosyntactic feature would suffice.

Conclusion

This paper has described four different solutions for the problem of representation of the morphological alternations in the lexicon. For each of these solutions, the advantages and disadvantages have been listed.

In our monolingual MT lexicons, we are currently using the second approach (cf. "Automatic Handling" above) for English and German, and the third approach (cf. "Using Morphosyntactic Features as Lexical Information" above) for Romance languages. Our objective is to change our lexicon coding for all languages to the fourth approach: using grammatical feature bundles for the unique identification of morphemes in the lexicon. We do believe this is the best approach, since it avoids all the disadvantages of the other approaches. However, we still need to investigate how to overcome its main disadvantage, described in the previous section. Future work in the next few months will certainly lead us to an adequate solution to this problem.

Acknowledgements

We would like to thank Anna Civil and Eugènia Torrella for their comments on earlier drafts of this paper.

References

Beesley, K.R. & Karttunen, L. (2000). Finite-State Non-Concatenative Morphotactics. In Proceedings of the Fifth Workshop of the ACL Special Interest Group in Computational Phonology, (pp. 1-12).

Black, A.W. & Ritchie, G.D. & Pulman, S.G. and Russell, G.J. (1987). Formalisms for morphographemic description. In ACL Proceedings, Third European Conference, (pp. 11-18).

Bosque, I. & Demonte, V. (1999). Gramática descriptiva de la lengua española (vol 3). Espasa.

Cahill, L. (1990). Morphonology in the lexicon. In Proceedings of Sixth Conference of the European Chapter of the Association for Computational Linguistics, (pp. 87-96).

Evans, R. & Gazdar, G. (1996). DATR : A language for lexical knowledge representation. In Computational Linguistics, 22.2, 167-216.

Karttunen, L. (1991). Finite-Sate Constraints. In Proceedings of the International Conference on Current Issues in Computational Linguistics.

Gerber, L. & Yang, J. (1997). SYSTRAN MT Dictionary Development. In Machine Translation: Past, Present and Future: Proceedings of Machine Translation Summit VI, pp.211-218.

Hildebrandt, L. (1996). Spanish morphological analysis (2nd approach). Sail Labs internal manuscript.

Loomis, T. (1993). Metal ++ Morphological Analysis. Sail Labs internal manuscript.

Loomis, T. (1991). The METAL 3.0 Lexicon. Sail Labs internal manuscript.

Loomis, T. (1998). T1 Lingware (Version 2.0). Sail Labs internal manuscript.