An approach to lexical development for inflectional languages

Davide Turcato*, Janine Toole*, Stavroula Tsiplakou†, Trude Heift†, Paul McFetridge*

*Natural Language Laboratory, School of Computing Science
{turk,toole,mcfet}@cs.sfu.ca
†Department of Linguistics
{stavroula_tsiplakou,heift}@sfu.ca
Simon Fraser University
8888 University Drive, Burnaby, British Columbia, V5A 1S6, Canada

Abstract

We describe a method for the semi-automatic development of morphological lexicons. The method aims at using minimal pre-existing resources and only relies upon the existence of a raw text corpus and a database of inflectional classes. No lexicon or list of base forms is assumed. The method is based on a contrastive approach, which generates hypothetical entries based on evidence drawn form a corpus, and selects the best candidates by heuristically comparing the candidate entries. The reliance upon inflectional information and the use of minimal resources make this approach particularly suitable for highly inflectional, lower-density languages. A prototype tool has been developed for Modern Greek.

1. Introduction

We describe a method for the semi-automatic development of morphological lexicons. A prototype tool, implemented in Perl, has been developed for Modern Greek, as part of a project for an Intelligent Language Tutoring System (Turcato et al., 2000).

We minimally define a lexical entry as a pair ⟨*citation-form, inflectional-class*⟩, where the latter is a label uniquely specifying an inflectional class. E.g. the following would be the entry for αγορά (*aghorá* 'market'):

(1) ⟨αγορά,**1Fp_α_1111**⟩

Given such a pair, all the inflected forms for the specified citation form can be generated. In addition to specifying a lexeme's declension, an inflectional-class label implicitly provides other kinds of lexical information, like syntactic category (adjective, noun, etc.), noun gender, etc.

The lexical development process only relies upon the existence of the following resources: (i) a raw text corpus; (ii) a database of inflectional classes. No other resource, like an initial lexicon, a Machine Readable Dictionary, or a simple word list of base forms is assumed.

The input to the lexical development procedure is a raw input text for which lexical entries need to be created. The procedure outputs lexical entries for the word forms found in the input text, drawing evidence from the two resources mentioned above.

2. Resources

This section provides a description of the two resources (text corpus and database of inflectional classes) used by the lexical development procedure.

2.1. Text corpus

A raw text corpus is used as an inventory of inflected words. It provides evidence about the existence of inflected forms generated from candidate lexical entries under consideration. Since the existence of single word forms is currently the only sought information, we turn a corpus into a list of unique instances of word forms, one per line, alphabetically sorted for fast retrieval. Syntactic context and word form frequency could also be useful kinds of information found in a raw text corpus, but they are not currently used. The corpus used for this purpose could also be the same corpus used as input to lexical generation. Also, as long as the corpus is used as a simple inventory of inflected forms, any available word list would be equally appropriate. For instance, in our prototype we used a word list from a publicly available Greek spell checker as both our input list and our inventory of inflected forms. For the sake of clarity, in the rest of the paper we will use the term *corpus* to refer to the corpus used to draw evidence from, and the term *word list* to refer to the input corpus for which lexical entries need to be generated.

2.2. Declension database

A database of inflectional classes is used to analyze inflected forms into stem and suffix, as well as generate hypothetical inflected forms for a given stem. A declension database is a set of ordered pairs ⟨*inflectional-class, suffix-list*⟩, each of which simply specifies a set of suffixes and associates it with an inflectional class label. E.g.

(2) ⟨**1Fp_α_1111**, [α/1, ας/1, ες/1, ων/1]⟩

where the suffixes are to be read [*a, as, es, on*]. As the example shows, each suffix is associated with a number (from 1 to 3) specifying which syllable is stressed in the corresponding inflected form (1 = last syllable, 2 = second to last, 3 = third to last). As already mentioned, what is specified here is simply a set of strings. In particular, suffixes are not associated here with any piece of the morphological information they carry. Therefore, the order of elements is irrelevant and does not stand for any canonical way of ordering a morphological paradigm (by number, case, etc.).

We simply chose the alphabetical order as our canonical way of sorting elements. Accordingly, a given suffix is only present once in an inflectional class, even if it is used more than once in a corresponding complete paradigm. For example, the suffixes $-\alpha$ and $-\epsilon\varsigma$ in (2) are used for nominative, accusative and vocative cases (singular and plural, respectively), but they are only included once in the inflectional class.

The declension database currently contains 93 inflectional classes (for nouns, adjective and verbs), loosely based on the morphological analysis provided in (Mackridge, 1985) and (Holton et al., 1997). The labels we assign to inflectional classes reflect the classification we use (although such labels are entirely conventional and their internal structure is not relied upon at any stage): e.g. the label **1Fp_**α**_1111** in (2) identifies nouns of class 1, feminine, parisyllabic, ending in $-\alpha$, with four oxytones items (i.e. stressed on the last syllable). We take a simple concatenative approach to combining stems with suffixes. Hence, we decided to include in the suffix epenthetic consonants and any other changes that strictly speaking would belong to the stem. This involves in some cases a proliferation of classes, but since such proliferation was limited, we accepted it for the sake of preserving the simplicity of a purely concatenative approach.

Finally, we note that the choice of encoding only word form information in the database, without attaching any morphological information, makes the database free from the assumption of any theoretical framework or specific formalism.

3. Methodology

The procedure comprises three phases. In the first phase candidate lexical entries are automatically generated for a set of related word forms, in the second phase candidate entries are automatically filtered, in the third phase surviving candidates are further filtered by human intervention.

3.1. Phase 1: generation of hypotheses

An input word list is scanned and the following algorithm, comprising three steps, is applied to each word form.

1. Given an input word form, a set of hypothetical lexical entries is created, one for each inflectional class matched by the input word form. Since the input word form is taken from raw text and is therefore inflected, the match can involve any inflected forms in an inflectional class. For instance, the input word form $\alpha\gamma o\rho\acute{\alpha}$, for which we showed the correct entry in (1), triggers the creation of 10 hypothetical entries (5 as adjective, 3 as noun, 2 as verb). All the candidates involve breaking the input word form into the stem $\alpha\gamma o\rho-$ and the suffix $-\alpha$/1, although any other split would be theoretically possible, as long as some suitable suffix is found in the declension database.

2. For each hypothetical lexical entry, the corpus is looked up for further evidence in support of the entry, i.e. for further inflected forms covered by the entry. A set of attested inflected forms (i.e. the original input word form plus all the new word forms found in support of the entry) is associated with each hypothetical entry. We show in (3) the attested forms found for some of the 10 candidates entries that were created on the basis of the input word form (2 as adjective, 2 as noun, 1 as verb, respectively):

   (3) **ADJ_3_1_1x11**: $\alpha\gamma o\rho\acute{\alpha}$, $\alpha\gamma o\rho\acute{\epsilon}\varsigma$, $\alpha\gamma o\rho\acute{\omega}\nu$.
   ADJ_3_2_1x10: $\alpha\gamma o\rho\acute{\alpha}$, $\alpha\gamma o\rho\acute{\alpha}\varsigma$, $\alpha\gamma o\rho\acute{\epsilon}\varsigma$, $\alpha\gamma o\rho\acute{\omega}\nu$.
   1Fp_α**_1111**: $\alpha\gamma o\rho\acute{\alpha}$, $\alpha\gamma o\rho\acute{\alpha}\varsigma$, $\alpha\gamma o\rho\acute{\epsilon}\varsigma$, $\alpha\gamma o\rho\acute{\omega}\nu$.
   2B_1111: $\alpha\gamma o\rho\acute{\alpha}$, $\alpha\gamma o\rho\acute{\omega}\nu$.
   verb2bs1: $\alpha\gamma o\rho\acute{\alpha}$, $\alpha\gamma o\rho\acute{\alpha}\varsigma$.

   The 4 relevant word forms (as listed under either **ADJ_3_2_1x10** or **1Fp_**α**_1111**) are to be read [*aghorá, aghorás, aghorés, aghorón*].

3. We compute the transitive closure of the set of hypothetical lexical entries under the relation $R = \{\langle L', L''\rangle$*: the sets of attested inflected forms associated with* L' *and* L'' *have an element in common*$\}$. In other words, given the overall set of inflected forms collected up to this point, steps 1-2 are recursively repeated for all the newly added inflected forms, and all the hypothetical lexical entries thusly found are clustered together.

   In our example, 3 additional word forms were found in the previous step ($\alpha\gamma o\rho\acute{\alpha}\varsigma$, $\alpha\gamma o\rho\acute{\epsilon}\varsigma$, $\alpha\gamma o\rho\acute{\omega}\nu$). For each of them, steps 1-2 are repeated, and newly found candidates are added to the cluster. For instance $\alpha\gamma o\rho\acute{\epsilon}\varsigma$ triggers the addition of 3 new entries:

   (4) **ADJ_3_8_**η**_111111**: $\alpha\gamma o\rho\acute{\epsilon}\varsigma$, $\alpha\gamma o\rho\acute{\omega}\nu$.
   1Fp_η**_1111**: $\alpha\gamma o\rho\acute{\epsilon}\varsigma$, $\alpha\gamma o\rho\acute{\omega}\nu$.
   1Mp_$\eta\varsigma$**_1111**: $\alpha\gamma o\rho\acute{\epsilon}\varsigma$, $\alpha\gamma o\rho\acute{\omega}\nu$.

   Overall, 13 new candidates are added in this phase for our example. Since none of these new candidates introduces any new word forms, no further iteration is performed. Therefore, the entire phase ends up with 23 candidates.

The resulting cluster contains all the hypothetical lexical entries connected, directly or indirectly, to the initial input word form. For instance, an input word form W_1 may result in a cluster whose entries are associated with the following sets of attested forms:

(5) $\{\{W_1,W_2\}, \{W_1,W_3\}, \{W_3,W_4\}\}$

where the first two elements are directly triggered by the base step on W_1, and the last element is triggered by the recursive step on W_3, and does not contain the input word form. In our example, this is illustrated by the candidates in (4).

The introduction of step 3 is motivated by the radically contrastive approach we take. In our approach we do not rely on any existing resource against which hypotheses can be checked. For instance, a simple resource like a word list of base forms would be sufficient to rule out most of the candidates in our example (all verb candidates, all noun

candidates involving base forms different from αγορά, like αγορή, αγορής, etc.). Since we do not rely on any such resources, all our evidence exclusively comes from comparing the candidates among themselves and retaining the fittest candidates. Before validating a candidate, and the association of a set of attested word forms to it, it is useful to check all other possibile assignments for the involved word forms, and choose a combination of candidates that best covers all attested word forms. For instance, in our hypothetical example (5) there would be no principled way of assigning W_1 to either $\{W_1,W_2\}$ or $\{W_1,W_3\}$, as long as only these two candidates are taken into account. However, taking also into account $\{W_3,W_4\}$ might provide some indirect evidence in favor of $\{W_1,W_2\}$, as the two candidates together would provide a complete coverage of the attested forms at hand. Another reason for this extended approach to clustering is that it provides lexicographers with a better overview of candidates they have to choose from, when manual selection is required.

3.2. Phase 2: hypotheses filtering

A number of heuristics are used to filter out competing lexical entries in a cluster. Currently, such heuristics are limited to either one-to-one comparisons between candidates, or checks on single candidates. The following heuristics have been currently implemented:

1. When a set of attested forms is a proper subset of another one, the former is removed. This move is based on the simple consideration that a candidate properly containing another candidate has a better empirical support. This is a more conservative move than just taking a count of the attested forms covered by a candidate as a measure of its empirical support, both because candidates have different cardinalities (i.e. different sizes for their complete set of inflected forms), and because a candidate with less attested forms in a direct comparison might get priority in the light of evidence coming from further candidates, as discussed for example (5). However, neither of such considerations holds in the case of proper containment.

   In our example concerning αγορά, most of the candidates are filtered out at this stage (namely, 21 out of 23). The two surviving candidates are the following:

   (6) **ADJ_3_2_1x10**: αγορά, αγοράς, αγορές, αγορών .
   1Fp_α_1111: αγορά, αγοράς, αγορές, αγορών .

   It can be checked that all other candidates listed in (3) and (4) are properly contained in the two candidates above.

2. When two sets of attested forms are identical, but only one is complete, the incomplete one is removed. In other words, given two candidates supported by the same amount of empirical evidence, we favor a complete one over an incomplete one, on consideration that the former has gathered all the empirical evidence that it could possibly gather. Obviously, the incompleteness of the removed candidate could be due to the insufficient size of the corpus. However, since the same line of reasoning would hold for any given size of the corpus, the adequacy of the corpus size is simply assumed. Also, a weaker version of this heuristics could perhaps be used, by choosing the candidate that achieves a higher rate of completeness, with respect to its complete set of inflected forms. In this case, the heuristics would apply even in the case of two incomplete candidates. In our example, the first of the two candidates in (6) is removed at this step. Therefore, the following candidate is (correctly) left as the only survivor:

   (7) **1Fp_α_1111**: αγορά, αγοράς, αγορές, αγορών .

3. Hypotheses associated with sets of inflected forms containing a single item are removed. This move is taken in order to cut down the number of generated hypotheses: evidence based on two word forms, at least, is required for a hypothesis to be taken into consideration. This move means removing some correct hypotheses, but currently the gain in terms of filtering out incorrect hypotheses provides sufficient motivation for its introduction.

3.3. Phase 3: hypotheses selection

A lexicographer selects the correct entries from a cluster. At this stage, each candidate is no longer presented as a set of attested forms, but rather as a complete paradigm, where all inflected forms are listed and associated with morphological features. For instance, given the following cluster of candidates:

(8) **2B_2222**: κεφαλαία, κεφαλαίο, κεφαλαίου, κεφαλαίων .
2B_3322: κεφάλαια, κεφάλαιο, κεφαλαίου, κεφαλαίων .

where the two sets of word forms are to be read [*kefaléa, kefaléo, kefaléu, kefaléon*] and [*kefálea, kefáleo, kefaléu, kefaléon*], respectively, lexicographers are presented with the following paradigms:

(9) a.

	Singular	Plural
Nom.	κεφαλαίο	κεφαλαία
Acc.	κεφαλαίο	κεφαλαία
Gen.	κεφαλαίου	κεφαλαίων
Voc.	κεφαλαίο	κεφαλαία

b.

	Singular	Plural
Nom.	κεφάλαιο	κεφάλαια
Acc.	κεφάλαιο	κεφάλαια
Gen.	κεφαλαίου	κεφαλαίων
Voc.	κεφάλαιο	κεφάλαια

We incidentally note that the example above is a special case, in that it presents two overlapping candidates, both

	Phase 1	Phases 1 & 2
Overall # of proposed clusters	771	351
Overall # of proposed entries	13768	1285
# of clusters with correct entry	225	196
# of clusters with just correct entry	0	79
Average # of entries per cluster-with-correct-entry	33	4

Table 1: Test results showing the effect of the filtering phase.

valid. The former refers to κεφαλαίο (*kefaléo* 'capital letter'), the latter refers to κεφάλαιο (*kefáleo* 'capital, fund'). In this case the lexicographer will validate both candidates.

Since morphological paradigms are presented as simple lists of inflected forms, no knowledge of specific formalisms is required on the lexicographer's part.

4. Results and future work

The following experiment was run, to test the effectiveness of the filtering phase. A list of word forms was created from a pre-existing, manually developed Greek lexicon. The lexicon contained 669 entries, 440 of which were potential candidates for automatic generation (nouns, adjectives, verbs). The word list was then used as input to the prototype in two different runs, the first of which only performed phase 1, while the second performed both phases 1 and 2. The resulting clusters were checked against the original lexicon. Table 1 summarizes the results.

Although recall is generally low (51.4% and 44.5%, respectively), mainly due to the incompleteness of the morphological database, the table shows that the filtering phase dramatically reduces the number of candidates (54.5% reduction in terms of clusters, 90.6% in terms of entries), while keeping the loss of recall within acceptable limits (6.6% loss). As a result, precision improves from 29.2% to 55.8%, in terms of clusters, and from 1.6% to 15.3%, in terms of entries. The filtering phase effectively replaces lexical resources in restricting the number of hypotheses.

We also note that multiple solutions are ineliminable in some cases, as long as only internal evidence about word forms is used to filter out candidate entries. This is so because some pairs of inflectional classes are identical in terms of the set of suffixes they use, only differing in the morphological features attached to the suffixes. This happens, for instance, with the following pair of inflectional classes:

(10) a. ⟨**1Fp**_η_**2221**, [η/2, ης/2, ες/2, ων/1]⟩

b. ⟨**1Mp**_ης_**2221**, [η/2, ης/2, ες/2, ων/1]⟩

where the suffixes are to be read [*i, is, es, on*]. The former class covers feminine nouns like κόρη (*kóri* 'daughter', with inflected forms κόρη, κόρης, κόρες, κορών [*kóri, kóris, kóres, korón*]). The latter class covers masculine nouns like κλέφτης (*kléftis* 'thief', with inflecting forms κλέφτη, κλέφτης, κλέφτες, κλεφτών [*klépti, kléftis, kléftes, kleftón*]). Nouns like κόρη have the suffix –η in the nominative singular and –ης in the genitive singular, while nouns like κλέφτης do the reverse. For example, in our second test (with filtering on) 14 of the 17 cases where two solutions were output were due to such systematic ambiguities of the relevant sets of suffixes.

Future work will focus on increasing and refining the filtering phase. We plan to extend our contrastive approach from binary comparisons (two candidates at a time) to n-ary comparisons, where more than two candidates are compared at the same time, in order to select the combination of candidates that provides the best coverage of the set of word forms under consideration. We also plan to extend our filtering criteria from a purely word-form-based approach, which only takes into account internal evidence coming from word forms, to a more context-sensitive approach, which takes into account simple, unambiguous syntactic contexts from the corpus to rule out hypotheses (e.g. a noun vs. a verb, a noun vs. an adjective, a masculine vs. a feminine noun). For instance, a simple inspection of the determiner preceding a noun can solve the ambiguities of the kind described above, every time two inflectional classes for masculine and feminine nouns are involved. Other heuristics include the use of quantitative techniques, for instance frequency counts to detect noise in the corpus due to spelling errors (Quasthoff, 1998).

5. Conclusion

The described lexicon development method is maximally self-contained. Unlike previous methods (ten Hacken et al., 1994) (Quasthoff, 1998) (Tuells, 1998), no lexicon is assumed. The lexicon creation process is only based on contrastive knowledge. Therefore, it works best with inflectional languages, which allow a better application of contrastive methods. In conclusion, this approach is particularly suitable for low-density inflectional languages, for which linguistic resources are rare, but monolingual corpora are available.

6. References

Holton, David, Peter Mackridge, and Irene Philippaki-Warburton, 1997. *Greek: A Comprehensive Grammar of the Modern Language*. London and New York: Routledge.

Mackridge, Peter A., 1985. *The Modern Greek Language: A Descriptive Analysis of Standard Modern Greek*. Oxford and New York: Oxford University Press.

Quasthoff, Uwe, 1998. Tools for automatic lexicon maintenance: Acquisition, error correction, and the generation of missing values. In Antonio Rubio, Natividad Gallardo, Rosa Castro, and Antonio Tejada (eds.), *Proceedings of the First International Conference on Language Resources and Evaluation (LREC-98)*. Granada, Spain.

ten Hacken, Pius, Stephan Bopp, Marc Domenig, Dieter Holz, Alain Hsiung, and Sandro Pedrazzini, 1994. A knowledge acquisition and management system for morphological dictionaries. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*. Kyoto, Japan.

Tuells, Toni, 1998. Constructing and updating the lexicon of a two-level morphological analyzer from a Machine-Readable Dictionary. In Antonio Rubio, Natividad Gallardo, Rosa Castro, and Antonio Tejada (eds.), *Proceedings of the First International Conference on Language Resources and Evaluation (LREC-98)*. Granada, Spain.

Turcato, Davide, Devlan Nicholson, Trude Heift, Janine Toole, and Stavroula Tsiplakou, 2000. A parsing methodology for error detection. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT-2000)*. Trento, Italy.