

Lightweight Morphology: A Methodology for Improving Text Search

TAPoR “The Face of Text” conference, Nov. 19-21, 2004,
McMaster University, Hamilton, Ontario, Canada

Mikaël Roussillon Bradford G. Nickerson Stephen Green William A. Woods

Faculty of Computer Science
University of New Brunswick
PO Box 4400
Fredericton, N.B. E3B 5A3, CANADA
{mikael.roussillon, bgn}@unb.ca

Sun Microsystems Laboratories
1 Network Drive
Burlington, MA 01803, USA
{stephen.green,
william.woods}@sun.com

October 29, 2004

Abstract

Lightweight Morphology is a new approach to morphological analysis, creating morphological variants from sets of rules. The rules are intuitive to define and at the same time, offering expressiveness and control. We defined a grammar for Lightweight Morphology. We defined how to generate English and French morphological variants using the grammar. French Language specification required 526 rules, 41 rule sets and 16,842 exception table words, while the English language specification took 123 rules, 17 rule sets and 2,589 exception table words. English and French Lightweight Morphologies were compared with two other techniques extending queries on a collection of 533 documents from the aligned Hansard of the 36th parliament (1997) of Canada. A differential recall comparison among the techniques showed that Lightweight Morphology has more queries (average of 3.9 times more) retrieving fewer irrelevant document for both English and French. The French Lightweight Morphology has more queries (average of 2.5 times more) retrieving more relevant documents.

1 Introduction

Interpretation of a text can be assisted by knowing the location and content of ‘like’ phrases and words using morphological variants. The challenge of using morphological variants is how to generate them. One can (a) have a complete lexicon for each language that includes all morphological variants, or (b) encode morphotactics as rules, so as to generate the variants of recognized morphological patterns [5, 8]. For example, in French, the ending ‘erait’ as in ‘aimerait’ identifies the conditional 3rd person singular of a first group verb, and we can produce the morphological variants corresponding to the first group verb paradigm, such as ‘aimer’, ‘aimes’, ‘aimions’ and many more.

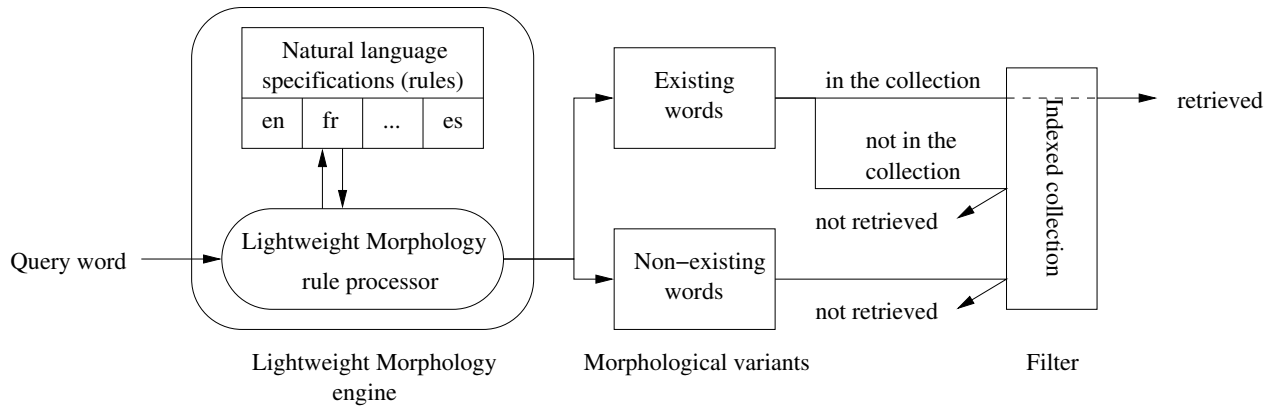


Figure 1: Lightweight Morphology variants production with rule specifications and filtering against the collection (*en* stands for English, *fr* for French, *es* for Spanish).

2 Lightweight Morphology

Lightweight Morphology is a methodology that follows the second approach. This approach is space-efficient and avoids the complexity of establishing a complete lexicon of all morphological variants. As a pre-processing step, Lightweight Morphology tries to expand the word into multiple morphological variants, therefore adding information to the word (see Figure 1). Lightweight Morphology consists of three components which are (1) a set of pattern-matching rules enclosed in rule sets that will produce morphological variants, (2) an escape mechanism to write rules in JavaTM programming language and (3) an exception table to handle exceptions that exist in a language. Lightweight Morphology has a modular approach that enables one to define morphologies for different languages or to introduce new approaches on the way the morphological variants are to be produced.

2.1 Pattern-matching rules and rule sets

A pattern-matching rule is made of the following elements:

1. On the left hand side of the rule, a pattern the word has to match in order to apply the rule. The pattern is defined with regular expressions, and some mechanism is provided to interact with the right hand side.
2. On the right hand side of the rules, a list of morphological variations that are to be applied in order to obtain the variants.
3. The left hand side is separated from the right hand side with the ‘->’ production symbol.

The operators of the pattern-matching rules allow to handle different kinds of affixes, like circumfixes and infixes, either by removal from the input word or by addition to the produced variants. Diphthongs and spelling modifications can also be handled. Figure 2 presents a simple example of a pattern-matching rule for English processing words ending with a vowel followed by *less*. The word is stripped from *less* and the list of morphological variation from the right

hand side of the rule are appended to form the morphological variants ('_' stands for the empty morphological variation). Figure 3 shows a more complex example to create the infinitive forms for French verbs like *assiéger*. It is supposed that the input to this rule is a verb stripped from its ending.

```
.aeiou + l e s s -> _,s,er,ers,est,ed,ing,ings,ly,ness,nesses,ment,ments,ful;
```

Figure 2: Pattern-matching rule example for English words ending with *less*.

```
.$Letter <é|è> g + ?e -> <é>/_er;
```

Figure 3: Pattern-matching rule example for French infinitive form creation.

The pattern-matching rules are themselves ordered within a rule set. The first rule that will be successful in the rule set (i.e. the pattern matches the word and the right hand side produces variants) will return the morphological variants obtained and no more rules will be tried. Different kind of rule sets can be used. The first one is the default rule set, the one that will be first tried when performing Lightweight Morphology. The ending rule set offers a different behaviour. It encloses rules that will apply on words having the same ending. Therefore, if an input word has an ending defined by an ending rule set, this specific rule set will be tried instead of the default rule set. Figure 4 presents an ending rule set containing the only rule processing English words ending with *-ical*. The third kind of rule sets are JavaTM programming language rule sets and will be explained in the next section. The fourth kind is a rule set whose sole purpose is to group a set of pattern-matching rules.

```
RULESET icalRules ENDING ical {
  .aeiouy + i c a l -> ic,ics,ically;
  // (e.g., academical, electrical, theatrical)
}
```

Figure 4: Example of ending rule set enclosing a rule for words ending in *ical*.

Each rule set can be called from a pattern-matching rule, the word matched and modified by the pattern-matching rule being the input word to the rule set. The usefulness of rule sets is therefore to group similar processing, either because of their ending which represents a certain morphological feature (e.g. the *-s* ending for plural), or because of the same problem they solve (e.g. creating from a specific kind of verb the present forms).

2.2 User-defined rules using JavaTM programming language

Sometimes, pattern-matching is not enough to describe morphotactics and create morphological variants. For those cases, Lightweight Morphology offers the possibility to write rules directly in JavaTM programming language to create user-defined rules. These rules behave much like rule

sets since pattern-matching rules can call user-defined rules and user-defined rules can call rule sets. Different user-defined rules can co-exist by creating different JavaTM programming language rule sets.

2.3 Exception table

Every language has exceptions or irregularities, and some are so exceptional that they cannot be encoded as rules. Most of the time, those exceptions are intensively used words that earned a very irregular pattern over the centuries. Classical examples that apply to many languages are ‘to be’ and ‘to go’. Lightweight Morphology provides a mechanism to write this kind of exception. The exception table can be seen as a kind of lexicon where morphological inflections and derivations of a word are grouped together as an entry. Figure 5 presents a small portion of the exception table for English words.

```
EXCEPTIONS {
  all;
  also;
  find, finds, found, finding, findings, finder, finders;
  found, founds, founded, founding, foundings, founder, founders;
}
```

Figure 5: Exception table for some English words.

Before processing an input word with rules, Lightweight Morphology first checks if the word appears in the exception table. If so, all entries containing the word are returned and no rules are tried.

The exception table can also be used for words where rules produce incorrect morphological variations (morphological variations that are real words but are not related to the input word), or for words that do not require processing by rules, like adverbs.

3 Lightweight Morphology specification

The essential work of Lightweight Morphology is to define pattern-matching rules for natural languages. Depending on the language and the person creating the specification, various approaches can be taken to construct a Lightweight Morphology specification a natural language. We present two approaches, one for English and one for French. Use of grammatical resources, like *Bescherelle* [1], *Bled* [2] or *Grevisse* [4] for French, is strongly suggested.

3.1 English

The English Lightweight Morphology has 123 rules, 17 rule sets and 2,589 exception table words. The rule sets are ending rule sets, except for the default rule set. The processing is therefore performed depending on the ending of the input word. Example of endings processed are -s,

-less or *-ing*, handling noun and verbs inflection and some derivations. The default rule set only processes endings that are not defined as an ending rule set. Heavy use of rule set calls is made, for example in some pattern-matching rules, words stripped from *-s* will go again through the rules as a new input word and create new morphological variants.

The exception table is an important part of the English Lightweight Morphology. It encodes irregular verbs (e.g. *find*) or words that fail to be processed by the rules (e.g. *firmament*).

On the set of 200 English words processed by the English Lightweight Morphology, 4% produced irrelevant variants (e.g. *billy* from *bill*) and 16% did not produce one or more important variant (e.g. *legislate* is not produced from *legislation* because of an unhandled derivation).

3.2 French

The French Lightweight Morphology has 526 rules, 41 rule sets and 16,842 exception table words. All rule sets are normal rule sets, except for the default rule set. Each normal rule set will perform a specific grammatical feature. For example, a rule set is used to process masculine adjectives and nouns, while another rule set is used to create the infinitive form for 1st group verbs. The default rule set dispatches the words to the appropriate rule set according to a pattern.

Support for derivation for the suffixes *-age*, *-ance*, *-ment*, *-eur*, *-ion*, and *-able* is provided creating morphological variants from and to these suffixes.

The exception table is extensively used to encode irregular verbs (e.g. *aller*, to go) and irregular form of adjectives (e.g. *beau*, beautiful) and nouns (e.g. *carnaval*, carnival).

The pattern-matching rules, along with the exception table, are supposed to handle inflections for verbs, nouns and adjectives. On a set of 200 French words, 5% produced incorrect variants (e.g. *paye* from *pays*), and 7,5% did not produce one or more important morphological variant (e.g. *rappporter* is not produced from *rapport*).

4 Testing

We tested the Lightweight Morphology approach with 200 English query words and 200 French query words in the aligned Hansard of the 36th parliament of Canada [3] which consists of 533 documents (19,999,604 tokens for the English version, 22,801,063 tokens for the French version). The English version contains 55,323 terms and the French one 76,031 terms. By term we mean a token that does not contain a cipher.

We compared Lightweight Morphology with two other search approaches: (a) stemming and (b) wildcards. For example the word ‘academic’ produces ‘academics’, ‘academical’, ‘academically’ for Lightweight Morphology and ‘academ’ for the Porter stemmer. A good wildcard query would be ‘academ*’ or ‘academic*’. We included the number of documents retrieved with exact query (no pre-processing) as a reference measure, because it is common among search engines.

The Porter stemmer [6] algorithm is used for English, a stemmer created by Martin Porter [7] is used for French. For the French Lightweight Morphology, we produced two sets of rules: (1) one that performs inflections and derivations and (2) one that only performs inflections.

For each language, 100 query words are randomly selected from an Ispell dictionary. If a word does not have a hit in the collection with one of the approaches, it is discarded. The remaining 100 query words are frequency selected from the Hansard. We sorted the words of the collection

by frequency and selected the first 100 meaningful words. We consider that a meaningful word is a word that is a uncommon word. Common words can be an article (e.g. *the, a*), a common verb (*be, go*) or any word that would not make much sense in a query as a single word (e.g. *however, first*).

To evaluate Lightweight Morphology against the other approaches, we used a qualitative measure known as differential recall. In differential recall, we compare two methods A and B and calculate:

- $A \cap B$ — The number of relevant documents found with variants from both A and B ;
- $A - B$ (resp. $B - A$) — The number of relevant documents found by variants from A (resp. B) but not from B (resp. A).

We decided that our relevance criterion would be the correctness of the variants found (e.g. for the query ‘tear’, we consider ‘torn’ to be relevant but not ‘tearmann’), without consideration for the context (e.g. we consider ‘tear’ to be relevant both in the context of a teardrop and of something that is torn).

For the 200 words of each language, we manually classified the variants produced by the three approaches as either relevant or irrelevant according to our relevance criterion.

5 Results

The following abbreviations will be used: **LM** (Lightweight Morphology), **LM'** (Lightweight Morphology without derivation processing), **S** (Stemmer), **W** (Wildcard) and **EQ** (Exact Query).

5.1 Relevance classification and differential recall

Some results for three selected English query words are presented in Tables 1, 2 and 3¹.

The results present a good sample of words retrieved by the approaches. The relevance/irrelevance classification is not perfect in part because of the collection that contains typos, spelling errors, neologisms and foreign words (many French words in the English Hansard and vice-versa). Judging a word to be relevant is often subjective, and our classification can be contested for some queries. For those few contestable classifications, we believe the impact on the differential recall is minor.

5.2 Differential recall win-lose score

In order to compare differential recall among all the words and all the approaches, we introduced a win-lose score, based on the ‘difference’ score ($A - B$ and $B - A$) from differential recall measures. For the relevance criterion, if $A - B > B - A$ for a query, A is awarded 1 point (A is retrieving more relevant document than B on the query). For the irrelevance criterion, if $A - B > B - A$ for a query, B is awarded 1 point (B is retrieving less irrelevant documents than A). The results of the win-lose scoring is given in Table 4 for the randomly selected words and in Table 5 for the frequency selected words.

Table 1: Variants found with Lightweight Morphology, Stemming and Wildcard.

Query	Approach	Example of English variants found in Hansard (number of documents found with this variant, relevance)
artist(85) (artist*)	LM	artist(85, rel), artists(136, rel)
	S	artist(85, rel), artists(136, rel), artistes(2, rel), artistic(46, rel)
	W	artist(85, rel), artists(136, rel), artistes(2, rel), artistic(46, rel), artistically(2, rel), artistique(1, rel), artistry(4, rel)
leaf(99) (leaf*)	LM	leaf(99, rel), leafs(21, rel), leafing(2, rel), leaves(299, rel)
	S	leaf(99, rel), leafs(21, rel), leafing(2, rel)
	W	leaf(99, rel), leafs(21, rel), leafing(2, rel), leaflet(3, irrel), leaflets(5, irrel)
tear(85) (tear*)	LM	tear(85, rel), tears(100, rel), tore(17, rel), torn(97, rel), tearing(40, rel)
	S	tear(85, rel), tears(100, rel), teared(1, rel), tearful(2, rel), tearing(40, rel)
	W	tear(85, rel), tears(100, rel), teared(1, rel), tearful(2, rel), tearfully(1, rel), tearing(40, rel), tearmann(1, irrel)

Table 2: Differential recall of Lightweight Morphology vs. Stemming for English.

Query Word	LM – S	LM \cap S	S – LM
artist	0	153	16
leaf	223	110	0
tear	55	182	1

Table 3: Differential recall of Lightweight Morphology vs. Wildcard for English.

Query Word	LM – W	LM \cap W	W – LM
artist (artist*)	0	153	19
leaf (leaf*)	223	110	0
tear (tear*)	55	182	1

Table 4: Differential recall Win-Lose scores for 100 randomly selected words.

Language	Criterion	LM - S	LM' - S	LM - W	LM - EQ	S - EQ
English	Relevant	24 - 24	N - A	8 - 44	82 - 0	81 - 0
English	Irrelevant	4 - 1	N - A	39 - 0	0 - 1	0 - 4
French	Relevant	35 - 13	25 - 36	11 - 33	90 - 0	93 - 0
French	Irrelevant	5 - 2	5 - 1	26 - 1	0 - 2	0 - 5

Table 5: Differential recall Win-Lose scores for 100 frequency selected words.

Language	Criterion	LM - S	LM' - S	LM - W	LM - EQ	S - EQ
English	Relevant	11 - 17	N - A	8 - 33	71 - 0	73 - 0
English	Irrelevant	21 - 4	N - A	63 - 2	0 - 9	0 - 25
French	Relevant	17 - 7	12 - 13	23 - 14	73 - 0	70 - 0
French	Irrelevant	20 - 5	19 - 3	50 - 7	0 - 11	0 - 23

First, we can see the usefulness of morphological analysis by comparing LM to EQ and to W queries. Compared to EQ, LM provides more relevant documents on 70 to 90 queries but introduces irrelevant documents on 1 to 11 queries. Most of the time, performing morphological analysis on terms will introduce relevant documents, and in a few cases, irrelevant documents.

When comparing to W, we can see the advantage of performing morphological analysis: the user does not need to think about the query. Wildcard queries generate comparatively more irrelevant words. For example, *pity* creates the query *pit** so it will retrieve unrelated words like *pitbull*. It is easy to often retrieve more relevant documents with a straightforward wildcard query; it is also easy to retrieve more irrelevant documents. Morphological analysis has a strong advantage since the user does not need to think about putting wildcard operators at the right place to ensure retrieval of many related and few unrelated documents.

Compared to S, English LM is better, not always by retrieving more relevant documents but by always retrieving fewer irrelevant documents. We can see that for English, LM has a small advantage over S on the first query set by retrieving fewer irrelevant documents in 4 queries and more in only 1. On the second query set, S retrieves more relevant documents on 17 queries (11 for LM). LM retrieves fewer irrelevant documents: in 21 queries, S retrieves more irrelevant documents and retrieves less irrelevant documents in only 4 queries. The Porter Stemmer for example, found the stem *intern* from *international* allowing variants such as *internalize* or *interned*. This illustrates that English LM can improve information retrieval systems by decreasing the number of irrelevant documents retrieved on a per query basis.

French LM performs better than S in every case, getting more queries retrieving more relevant documents and more queries retrieving fewer irrelevant documents. French Lightweight

¹As documents can be retrieved by more than one variant, results from Table 1 do not necessarily translate to the numbers in Tables 2 and 3

Morphology does not find verbs from nouns ending with a consonant (e.g. *accorder* from *accord* or *travailler* from *travail*) reducing its relevant retrieval efficiency.

Processing derivations in French is important. Ignoring the suffix derivations decreases the relevant documents retrieved, making LM' less efficient than S for returning relevant documents. The number of queries retrieving fewer irrelevant documents is almost unchanged, illustrating that processing derivation does not appear to include spurious variants.

6 Conclusion

Lightweight Morphology is a new approach on morphological analysis offering more control than stemming over the variants retrieved. The way Lightweight Morphology is defined is intuitive compared to the way stemming algorithms are defined. The consequence is that English Lightweight Morphology will outperform stemming by retrieving less irrelevant documents on a per query basis. Stemming has a small advantage at retrieving more relevant documents on a per query basis, essentially because the English Lightweight Morphology does not process some derivations. The French Lightweight Morphology was superior in both having more queries retrieving more relevant documents and fewer irrelevant documents.

Future research for Lightweight Morphology involves creating specifications for other natural languages and seeing where to improve the limits of pattern-matching rules. JavaTM programming language rules should be reworked and replaced by a specific language performing advanced string manipulation. Finally, other tests should be performed on other collections and with other query words.

7 Acknowledgements

Sun Microsystems Inc. is gratefully acknowledged for providing funding for this research. Thanks are also due to the University of New Brunswick Faculty of Computer Science for their support. Experiments for this research were carried out on equipment established with funding from the Canada Foundation for Innovation (TAPoR project).

References

- [1] *Bescherelle : La Conjugaison pour tous*. Hatier, Paris, 1997.
- [2] E. Bled and O. Bled. *Bled : orthographe-grammaire*. Hachette, Paris, 2003.
- [3] Ulrich Germann. Aligned Hansards of the 36th parliament of Canada release 2001-1a (proceedings from September 25, 1997). Resources available at <http://www.isi.edu/natural-language/download/hansard/>.
- [4] M. Grevisse. *Le Bon Usage: Grammaire française avec des remarques sur la langue française d'aujourd'hui*. Duculot, Paris, 11ème edition, 1980.

- [5] R. Krovetz. Viewing morphology as an inference process. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Linguistic Analysis, pages 191–202, 1993.
- [6] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [7] M.F. Porter. French stemmer. <http://snowball.tartarus.org/french/stemmer.html>.
- [8] W. A. Woods. Aggressive morphology for robust lexical coverage. Technical Report TR-99-82, Sun Microsystems, 1999.