Unsupervised acquisition of concatenative morphology

Lionel Nicolas[⋆◇], **Jacques Farré**[◇], **Cécile Darme**[◇]

[⋆] Institute for Specialised Communication and Multilingualism, European Academy of Bozen/Bolzano, Italy,
[◇] Team RL, Laboratory I3S, University of Nice Sophia-Antipolis, France,

lionel.nicolas@eurac.edu, *Jacques.Farre@unice.fr*, *darme@i3s.unice.fr*

Abstract

Among the linguistic resources formalizing a language, morphological rules are among those that can be achieved in a reasonable time. Nevertheless, since the construction of such resource can require linguistic expertise, morphological rules are still lacking for many languages. The automatized acquisition of morphology is thus an open topic of interest within the NLP field. We present an approach that allows to automatically compute, from raw corpora, a data-representative description of the concatenative mechanisms of a morphology. Our approach takes advantage of phenomena that are observable for all languages using morphological inflection and derivation but are more easy to exploit when dealing with concatenative mechanisms. Since it has been developed toward the objective of being used on as many languages as possible, applying this approach to a varied set of languages needs very few expert work. The results obtained for our first participation in the 2010 edition of MorphoChallenge have confirmed both the practical interest and the potential of the method.

Keywords: unsupervised, morphology, acquisition

1. Introduction

Because morphological rules are still lacking for many languages, the automatized acquisition of morphology is an open topic which interest has been attested by an annual challenge(Kurimo et al., 2010) dedicated to this task.

In this paper, we present an approach that allows to automatically compute, from raw corpora, a data-representative description of the morphology of concatenative languages, i.e., a description of morphological mechanisms that rely on prefixes and suffixes.

Our approach takes advantage of phenomena that are observable for all languages using morphological inflection and derivation but are more easy to exploit with concatenative mechanisms. Among these phenomena, a frequency-related occurrence of the forms inflected or derivated from a same lemma is highlighted and intensively exploited.

Since this approach is implemented with mostly straightforward and parameters-free formulas and has been developed toward the objective of being used on as many languages as possible, applying this approach to a varied set of concatenative languages requires very few expert work.

The whole approach works as a sequence of filters refining a list of morphological rules.

The main contributions of this piece of research are:

1. to highlight a frequency related phenomenon,
2. to present several filters general enough to be adapted,
3. to describe a sequential combination of these filters and an evaluation of its results.

The main objective of this paper is to complete a previous one(Nicolas et al., 2010) by providing more details, more examples and better explanations.

2. Related work

As the results presented have been obtained at the 2010 edition of MorphoChallenge, we focus on methods that are directly or indirectly connected to the different editions.

The approaches for the automatic acquisition of morphological knowledge can be classified among two types: the ones that build shallow morphological analyzers and the ones that acquire morphological knowledge and apply it.

Within the first type, the methods described in Creutz and Lagus (2005) and Goldsmith (2006) are the most referenced ones. In Goldsmith (2006), the authors introduce the concept of *MDL* (*Minimum Description Length*) which relies on the idea of encoding/factorizing a corpora with a set of morphemes as small as possible, i.e., the better the affixes and stems are identified, the better the corpora will be encoded/factorized. In Creutz and Lagus (2005), the authors also start with an *MDL*-approach but eventually use a combination of a *Maximum Likelihood* and *Viterbi* algorithms to better encode/factorize the forms. It has been later extended in Kohonen et al. (2009) in an attempt to handle allomorphy.

In a different manner, in Golenia et al. (2009), *MDL* is used to first determine a set of candidates stems. The remaining substrings of the forms are considered as candidates affixes and split into letters to be later agglomerated as affixes according to a metric based on the substrings' frequencies.

In Spiegler et al. (2010) as in Bernhard (2008) and Keshava (2006), the authors describe methods that originate from Harris' approach (1955) and its follows-up (Hafer and Weiss, 1974; Déjean, 1998). These approaches focus on *transition probabilities* and *letter successor variety*. The method described in (Demberg, 2007) follows the algorithm in Keshava (2006) and corrects important drawbacks. Among them is a drawback due to a statement with a direct bias towards languages that make an intensive use of the empty suffix such as English.

Within the second type of methods that, as we do, explic-

itly list morphological knowledge and apply it, we have inventoried six other methods (Lavallée and Langlais, 2010; Bernhard, 2010; Can and Manandhar, 2009; Lignos et al., 2009; Monson et al., 2008; Dasgupta and Ng, 2007). In Lavallée and Langlais (2010), the authors identify analogies, e.g., "live" is for "lively" what "cordial" is for "cordially". Each analogy receives a weight according to the number of times where the analogy is attested and the number of times the analogy could apply.
In Lignos et al. (2009), the approach is similar except that the weight is computed with the number of shared candidate stems and the number of letters of the affixes.
Oppositely, in Bernhard (2010), all possible pairs of words are compared and morphological analogies are identified according to the edition distance. The analogies are then used to link forms and a clustering algorithm is performed to group the forms of a given lemma.
In Can and Manandhar (2009), the authors first achieve a clustering method so as to group forms with similar syntactic behaviors. Morphological rules are then detected by analogy between sets of forms in different clusters. Each morphological rule receives a score computed with the number of common stems.
In Monson et al. (2008), the authors build paradigms in a "brute-force" fashion controlled by thresholds.
In Dasgupta and Ng (2007), the authors extend the approach described in Keshava (2006) by adding several features to better handle compound affixes, related form occurrence when cutting and allomorphy. The method described is similar to ours in the sense that it sequentially refines a list of candidate affixes and gradually improves its quality. As in Keshava (2006), the method applies overall on languages that make intensive use of the empty suffix.

3. General definitions and informations

Some of the computations are done thanks to letter trees (see fig. 1). An affix is said *to occur on a given node* if it has been combined with a (prefix or suffix) substring of a form and the letters of this substring label a path from the root to this node. An affix combined with n different substrings in n different forms will thus occur on n different nodes.

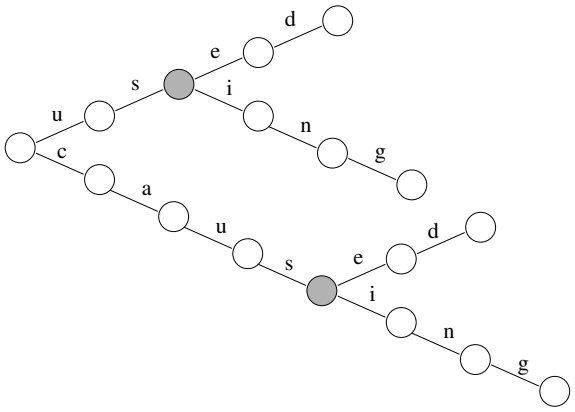

Figure 1: Simplified example of a letter tree. Suffixes "ed" and "ing" occur on gray nodes with stems "us" and "caus".

As a shortcut, a form is designated as *frequent* if its frequency is above the average frequency.

Since only raw text is provided as input, when referring to the lemmas of the forms, we actually abstractly do so in order to better explain our approach.
In this paper, morphological rules are fairly simple/naive. They consist in adding an affix to a given stem with no character deletion or substitution. Linguistic phenomena that modify the stem are acquired as different morphological rules than the ones they are derivated from.
We call morphological family a set of morphological rules. Although our examples focus on suffixes, the approach applies indifferently on prefixes or suffixes.
Finally, all substrings starting or ending a form are marked with a "#" at their beginning or their end.

3.1. Frequency related phenomenon

In a given text, some lemmas are more frequent than others. When considering a given lemma in a language, the probabilities for its inflected or derivated lexical forms to occur in a text increase with the frequency of the whole lemma. In other words, the more a lemma is frequently used in a corpus, the more probable it is to encounter a more diversified sample of its related lexical forms. For example, the various forms related to the lemma *to talk* are usually easier to encounter than the ones related to the lemma *to orate*.
Therefore, the more frequent a form is, the more chances there are to find its morphologically related forms. This phenomenon applies to most kind of text, be it specialized or general, except of course those describing exhaustively the lexical forms related to some designed lemmas.
There exist aspects, such as the style of the writer or the type of the corpora, that affect the ratio among morphologically related forms. For example, autobiographies favor the use of the first person singular.
Nevertheless, it does not alter the global chances of occurrences of the whole set of forms belonging to a same lemma. Consequently, it does not alter the fact that the more frequent a form is, the more chances there are to find some morphologically related forms within the corpus.

4. Global overview

The approach can be summarized as follows.

1. Establish an over-covering "naive" list of candidate affixes, i.e., substrings that may be affixes.
2. Detect pairs of candidate affixes that are related within morphological families. For example, for a family with three affixes A, B and C, detect the pairs $\{A, B\}$, $\{B, C\}$ and $\{A, C\}$.
3. group pairs occurring on a same node to build morphological families. For instance, group the pairs $\{A, B\}$, $\{B, C\}$ and $\{A, C\}$ present on a same node to build a family $\{A, B, C\}$.
4. Filter incorrect morphological families.
5. Split compound affixes, i.e, split suffix "ingly#" into "ing#+ly#".
6. Detect what substrings connect stems and split compound stems, i.e, detect that "-" can connect stems and split "brother-in-law" into "brother + in + law".

5. Identifying candidate affixes

This first step is designed to save computations for the more precise and computationally-intensive later filters. It computes a naive list of candidate affixes with the substrings ending or starting the forms. For example, when looking for suffixes, "aive#", "ive#", "ve#", "e#" and "#" shall be generated from the form "naive".
The list is then filtered "permissively", i.e, we avoid strict criteria such as a maximum length of characters. A candidate affix is kept when fulfilling the following conditions.

1. It occurs in at least one sub-tree present at least twice in the whole tree.
2. It occurs in at least one node covering a frequent word.
3. It is more likely to be an affix than the substrings it is combined with, i.e, it is combined with more substrings than the substrings it is combined with are themselves combined with other substrings[1].
4. It occurs more frequently on nodes with other substrings than it occurs alone.
5. It occurs at least twice with another candidate affix.

6. Identifying pairs of candidate affixes

6.1. Leading idea

This filter aims at identifying pairs of related affixes, i.e., pairs of affixes that both belongs to one or several same families. It relies on two considerations.
First, a morphological family covers at least two affixes[2] [3].
The second relies on the frequency related phenomenon described in sect. 3.1. as well as on a specific characteristic of concatenative mechanisms that allows to take advantage of this phenomenon more easily.
Indeed, contrarily to other kind of mechanisms, concatenative ones do not alter much the stem[4]. When inserting all lexical forms in a tree, the related forms of a lemma will follow a same path from the root until they pass the last common letter of the stem and spread in different branches according to their respective affix.
Consequently, all related forms of a lemma occur on a same last common node formalizing the frontier between the stem and the affixes. In order to find the related candidate affixes of a candidate affix *aff*, one only needs to pay attention to the nodes where *aff* occurs. This aspect allows to keep the search space in reasonable boundaries and exploit more easily the frequency related phenomenon.
Indeed, if *aff* truly belongs to a family *fam*, the more frequent a form containing *aff* is, the more chances there are for its morphologically related forms to be also present in the corpus. Therefore, the more probable it is for *aff* to occur on the corresponding node with other affixes of *fam*.
For example, since the different forms of the lemma "to talk" are more likely to occur than the ones of "to orate", "ing#" is more likely to co-occur on the node corresponding to "talking" with more related suffixes than on the node corresponding to "orating". Such phenomenon should *globally* apply to most pairs of nodes in the list.
Therefore, by sorting on frequency the nodes where a correct candidate affix occurs, we observe a progressively increasing co-occurrence rate with other affixes of the family. On the other hand, if the candidate affix is incorrect and/or has no relation with some random candidate affixes with which it co-occurs on some nodes, the co-occurrence rates shall be chaotic.

6.2. Practical application

A list of nodes where each candidate affix *aff* occurs is computed and then sorted according to the frequency of the form containing it.
This list is then split in sublists with the condition that the average frequency of a sublist s_i is $mult$ times higher ($mult > 1$) than the previous sublist s_{i-1}[5]. We then compute for each candidate affixes co-appearing with *aff* a co-occurrence rate $rate_i$ over each sublist and a score inc

$$inc = sum_{pos} + mult * sum_{neg}$$

where sum_{pos} is the sum of the positive value $rate_i - rate_{i-1}$, whereas sum_{neg} is the sum of the negative ones.
If $mult$ is superior to 1, negative progressions impact inc more than positive ones do. Therefore the $rate_i$ values have to globally increasing so as for inc to be positive. The co-occurrence is thus considered as increasing when inc is positive. Candidates with no increasing co-occurrence with other candidates are discarded.

6.3. Incorrect englobing or englobed pairs

One must note that this filter can identify pairs of incorrect candidate affixes because of correct ones. Indeed, any correct affix *X* related to an affix *Y* can allow the incorrect candidate affixes *subX* and *subY* starting with a same substring *sub* to be considered as related since their co-occurrence rate will also be an increasing one. For example, the English suffixes "*ing#*"/"*ed#*" can allow "*ming#*"/"*med#*" to be considered as related. We later refer at these type of pairs as *incorrect englobing pairs*.
The same fact applies for two incorrect affixes *Z* and *W* and two correct and related affixes *subZ* and *subW* starting with a same substring *sub*. For example, the pair of English suffixes "*es#*"/"*ed#*" can allow "*s#*"/"*d#*" to be considered as related. We later refer at these type of pairs as *incorrect englobed pairs*.

7. Morphological families

7.1. Building morphological families

Once pairs are identified, we recursively process the tree and build morphological families by grouping the pairs present on each node. A basic approach could be to merge

[1] We thus previously compute for any starting and ending substrings the number of substrings it is combined with.

[2] The empty string is considered as an affix, e.g., the form "think#" has a stem "think" and the empty suffix "#".

[3] We do not consider lemmas that only cover one form since morphological rules are unnecessary.

[4] Concatenative phenomena that do modify the stem, such as allomorphy, are actually acquired as different morphological rules than the ones they are derivated from.

[5] The first sublist is the set of nodes corresponding to the forms with the lowest frequency.

together all the pairs found on a node. For example, if the pairs $\{A, B\}$, $\{B, C\}$ and $\{A, C\}$ are present on a same node, this basic approach would build a family $[A, B, C]$. Nevertheless, it is not rare for two different families to be present on a same node. For example, the Spanish verbs "sentir" (*to feel*) and "sentar" (*to sit*) belong to two different families but share the same stem "sent". The basic approach is thus likely to build incorrect families that merge together several families.

In order to avoid such problem, the following iteration is applied on each node until no candidate affixe remains:

1. each candidate affix "votes" for the other candidate affixes with which it shares a pair;
2. a family is built with the pairs of the candidate affix that has received most votes;

Several different families present on a same node will not be merged together unless their most "popular" affixes are the same. Since a same family can be built on several nodes, we record for each the nodes where they have been built.

Four kinds of families can be generated:

1. correct complete or incomplete families,
2. incorrect ones brought by incorrect englobing pairs,
3. incorrect ones brought by incorrect englobed pairs,
4. completely incorrect ones.

7.2. Cutting forms

Because it is also used when filtering families, we now detail the algorithm that we used to cut forms . However, one must note that the set of families provided to the algorithm depends on the step of the approach in which it is used.
Morphological families are used to split every word as *prefix(es) + stem + suffix(es)*. A family is said to apply on a node if it covers *n* ($n > 1$) substrings[6] occurring on the node and thus generates a possible set of *n* cuts, one for each covered form occurring on the node.
For each form included in several possible sets of cuts, a choice is achieved by eliminating them sequentially according to three criteria:

- the greatest number of cuts,
- the smallest distance from the root to the node,
- the largest size for the corresponding family.

The first criteria relies on the idea that the more forms are covered by a family, the more correct the resulting set of cuts is. The second one favors longer affixes. Finally, the third one emphasizes the fact that larger families are usually the most accurate ones.
If after those tree steps, more than one set remains, we simply select the first one. Indeed, since the remaining sets cover as much forms, cut on the same node and the families that have been applied are all equivalent in size, the competing families are likely to be sub-families of a non-acquired bigger family.

7.3. Filtering morphological families

Filtering on sub-families
This filter directly addresses the incomplete families and the completely incorrect ones. As explained in sect. 3.1., depending on the lemma, more or less related forms are found in the input corpus and thus, more or less complete families are generated. A correct family of *n* affixes shall appear in sub-families with *n*, *n-1*, ...,*1* of its affixes. A family with *n* affixes is thus kept if "validated" by the occurrence of at least one family with *n-1* of its affixes[7]. For instance, a family $[A, B, C]$ is validated if any of the families $[A, B]$, $[B, C]$ or $[A, C]$ is generated. All families validating another one are discarded (mostly sub-families) as well as families that have not been validated (mostly the biggest completely incorrect families).
One must notice that if two equivalent families with *n+1* affixes sharing *n* of their affixes are generated, this filter will keep both unless a family with *n+2* affixes covering them appears.
Also, this filter is not effective with incorrect families brought by incorrect englobing or englobed pairs. Indeed, the sub-families of the correct families they are derivated from will provide the incorrect sub-families necessary to pass this filter.
It also proved to be less effective over small completely incorrect families since they require less sub-families to be validated. These small incorrect families are usually built from infrequent forms with no other related forms present in the corpus.

Filtering on frequent forms
This filter tends to compensate the previous filter regarding small completely incorrect families. It relies on the idea that morphological families are frequency-independent, i.e., they apply indifferently on frequent or infrequent lemmas. A correct family should thus cover at least, in one of the nodes from which it has been built from, one of the forms considered as frequent.

Filtering englobing families
This third filter follows the idea that the letters common to all related forms should belong to the stem. It thus tackles the incorrect families brought by incorrect englobing pairs by simply rejecting all families composed of affixes starting with an unique common first letter.
For example, $[r, rs, ring, red]$ is a family acquired with the incorrect candidate stem "bothe" and shall not be kept.

Filtering dominated families
We call dominated families the ones that are never selected by the cutting algorithm because they have less affixes than the families they compete with or they only apply on node that occur too deep in the tree. These dominated families are mostly non-filtered sub-families or incorrect ones brought by englobed pairs.

[6] A family is not applied if not covering at least two forms.

[7] Families with two affixes are automatically validated.

Indeed, englobed pairs are found deeper in the tree than the correct pairs they are "derivated from". Consequently, the family built from englobed pairs cannot cover the other affixes since they are in different sub-trees.
For example, let us consider a family $[ab, ac, bd, be]$ built thanks to the pairs $\{ab, ac\}$,$\{bd, be\}$,$\{ab, bd\}$, etc. The pairs $\{ab, ac\}$,$\{bd, be\}$ can allow the incorrect englobed pairs $\{b, c\}$,$\{d, e\}$ to be identified and the families $[b, c]$ and $[d, e]$ to be built. Those two incorrect families shall always be dominated by $[ab, ac, bd, be]$.
We thus run the cutting algorithm with all the families and discard the ones that have not even been selected once.

8. Splitting compound affixes

The affixes acquired can either be singleton like the English suffixes "ing#" and "ly#" or compound as "ingly#".
If an affix *aff3* in a family *fam1* is to be split as two affixes *aff1* and *aff2*, we consider that *aff3* is obtained by refining an affix *aff1* with a family *fam2* containing the affix *aff2*.
We thus consider that *aff1* as some kind of "stem" where *fam2* can apply. Consequently, there should be other affixes in *fam1* obtained by refining *aff1* with other affixes of *fam2*. Therefore, we list the other affixes in *fam* that could be a "stem" for *aff3* by following the idea that they should be more present than absent on the nodes where *aff3* occurs. For example, for every node where the suffix "ingly#" occurs, one can also expects to find the suffix "ing#".
We then apply the cutting algorithm as if we were dealing with regular forms. The family that covers most elements, including *aff3*, is selected, *aff3* is split as *aff1+aff2* and the process is recursively applied on *aff2*.

9. Splitting compound stems

So as to split compound stems, we first determine what substrings can connect them. For example, in order to split the compound stems of the form "grand-mother", we need to identify the substring "-" as a valid "connector". We could observe that these *connectors* act like double-affixes since they tend to connect two surrounding stems the same way suffixes are connected to the first one and prefixes are connected to the second one.
We also observed that, if enough data are provided, the most frequent forms tend to be identified along with connectors as "fake" affixes and provide an useful occasion to guess those connectors. For example, in English, the substrings "#grand", "#first-" are identified as prefixes whereas the substrings "-based#", "man#" are identified as suffixes. So as to identify the fake affixes and extract the corresponding connector, we apply the cutting algorithm to all the forms. We then establish two lists of *starting* and *ending* substrings corresponding to the combination of the stems with the prefixes and suffixes they have been found with. For example, if the English stem "appear" is found with the prefixes and suffixes "#re", "#dis", "ing#" and "ed#", the substrings "#reappear","#disappear","appearing#" and "appeared#" are used as *starting* and *ending* substrings.
We then identify all the prefixes containing *starting* substrings and all the suffixes containing *ending* substrings. The part of these "fake" affixes that do not belong to the *starting* or *ending* substrings are considered as candidate connectors. A connector is kept if it is both found in one fake prefix and one fake suffix. For example, in our experiments on English, the connector "-" has been found in the "fake" prefix "#first-" and in the "fake" suffix "-based#".
Finally, the stem of a given form is split if the form combines a *starting* substring, a connector and an *ending* substring. For example, the English stem for the form "speedboats" was split into the two stems of the forms "speed" and "boats" since "speedboats" combines the *starting* substring "#speed", the empty connector and the *ending* substring "boats#".

10. Samples of morphological families

e# , eable# , ed# , ement# , ements# , er# , ers# , es# , ing# , ment#
, 's# , -like# , ed# , er# , er's# , ers# , ers'# , ing# , s#
e# , ed# , er# , er's# , ers# , ers'# , ership# , es# , ing#
e# , ed# , es# , ing# , ion# , ions# , ions'# , or# , ors#

Figure 2: Sample of English suffix families acquired

e# , en# , end# , ende# , enden# , ender# , endes# , er# , ern# , t# , te# , ten# , ung# , ungen#
e# , en# , end# , ende# , endem# , enden# , ender# , endes# , er# , t# , te# , ten# , ung# , ungen#
, e# , en# , end# , ende# , enden# , ender# , endes# , er# , t# , te# , ten# , ung# , ungen#
, e# , em# , en# , er# , ere# , erem# , eren# , erer# , eres# , es# , este# , esten#

Figure 3: Sample of German suffix families acquired

As one can observe above, the families can be numerous and include indifferently inflections and derivations.

11. Comparison with related works

.

11.1. General discussion

Most approaches, devised and tested with the languages understood by their authors, tend to require adaptations when applied on another language. If these adaptations are not trivial, so as to know whether the results are relevant or not, the person using the tool needs both competences to adapt and tune the tool and to understand the morphology of the acquired language. Obviously, such competences can drastically reduce the range of users to a smaller number of skilled ones. Our approach has been developed towards the objective of avoiding such restriction.
It is achieved by following the idea that various phenomena can be exploited without narrowing too strictly at a given moment the search space. The search space is thus narrowed by a succession of filters, each one taking advantage of a given phenomenon. Each filter follows the idea that if the language has a certain aspect then the filter should be able to *at least* reduce the search space to a certain degree where its relevance/coherence are not/less subject to bias.
Indeed, in every step of our method, there is a certain "caution" regarding the criteria applied. This "caution" intends to guarantee the application of our approach with no formula adaptation or variable tuning. For instance, no maximum length is set for candidate affixes; morphological families require at least two affixes (not three or more); their suffixes shall start with at least two (not three or more) different letters; a form is cut if only one (not two or more)

other possibly related form appears; no prior knowledge, even fairly global such as the hyphen for compounding, is provided; etc.

11.2. Step-by-step comparison

Candidates affixes
Other approaches generally consider the entire set of possible affixes and let the following steps handle them or directly restrict them with a maximum length of characters. In Golenia et al. (2009), the candidate affixes are the substrings that are not part of previously detected stems.

Pairs of candidates affixes
Only methods that list morphological rules intend to explicitly identify relatedness between candidates affixes. The relatedness is often characterized by a score. Morphological rules are either filtered according to a given threshold (Lavallée and Langlais, 2010; Bernhard, 2010; Dasgupta and Ng, 2007) or their score do not allow them to correctly compete when cutting forms (Lignos et al., 2009).

Morphological families
Building set of affixes is considered in three methods (Goldsmith, 2006; Can and Manandhar, 2009; Monson et al., 2008). Among these, two (Goldsmith, 2006; Can and Manandhar, 2009) report rather small sets.
On the other hand, Monson et al. (2008) exploits, as we do, the concept of subsets of affixes in order to validate bigger set of affixes. The construction and filtering of completely incorrect sets, small correct ones and incorrect ones brought by *englobing* pairs are controlled by thresholds. They however do not deal with families brought by *englobed* pairs.

Cutting forms
Regarding methods that explicitly list morphological rules, a cut is usually considered, as we do, when two related form occur. One exception is the method described in Dasgupta and Ng (2007) that allows the absence of related form according to the frequency of the form to cut. Such approach partially implies the frequency-related occurrence of morphologically related forms.
The other methods do not explicitly require related affixes (Goldsmith, 2006; Creutz and Lagus, 2005; Spiegler et al., 2010; Keshava, 2006), or a common stem for two forms. They rather compute, for every possible cut, a score that tends to be higher when the cut is on the frontier between stem and affix. For these methods, the occurrence of other related forms in the corpus is therefore not requested but they do participate in scoring whether a certain substring might be a true stem or not.
If various morphological analyses are possible, other approaches either select several analysis above a certain threshold (Golenia et al., 2009; Spiegler et al., 2010; Lavallée and Langlais, 2010; Lignos et al., 2009; Bernhard, 2010; Monson et al., 2008), or they produce unambiguous analysis by selecting, as we do, analysis that maximize a given criteria or score(Creutz and Lagus, 2005; Keshava, 2006; Dasgupta and Ng, 2007).

Splitting compound affixes
Except in Dasgupta and Ng (2007), other approaches that do intend to handle compound affixes(Creutz and Lagus, 2005; Spiegler et al., 2010; Keshava, 2006; Lavallée and Langlais, 2010; Lignos et al., 2009; Can and Manandhar, 2009), simply cut shorter affixes when possible and reiterate the process on the remaining substrings. Consequently, most approaches are likely to over-cut the French form "*parleras*/(you) will speak" before the last 's' since the French form "*parlera*/(he) will speak" exists and the concatenation of an 's' is a correct morphological rule.

Splitting compound stems
The approaches that intend to handle compound stems (Creutz and Lagus, 2005; Spiegler et al., 2010; Lignos et al., 2009; Can and Manandhar, 2009) often rely, as we do, on the idea that the contained stems should be more frequent than the compound stem itself. However, this decision is usually determined according to the frequencies of the forms involved in the choice when we actually take advantage of the affixes found with the contained stem. Such frequency-based approach is more likely to fail when facing compound stems, such as "basketball", that tend to be more frequent than their contained stems.
Except in Lignos et al. (2009), no other method identifies *connectors* strings but rely on the occurrence of smaller forms to split a bigger one. Thus, no other method actually identify substrings dedicated to the composition of stems such as the hyphen [8] or the "o" in French ("*latino*/*américain*/latin-american").

12. Evaluation

12.1. MorphoChallenge

Fortunately for the morphology acquisition task, an annual challenge focusing on morphological analysis from raw data has been organized every year from 2005 until 2010(Kurimo et al., 2010). This challenge provides a set of evaluation tools which represent a consensual way to estimate the quality of an approach. In order to feed these tools, one needs to produce morphological analysis of forms as sequence of morpheme labels. In our evaluations, we directly used the generated stems and affixes as morpheme labels, i.e, our labels are spelling-motivated ones.
As explained on the 2010 edition website (Mik, 2010), since the task involves unsupervised learning, the evaluation tools provided do not expect the algorithms to come up with morpheme labels that match the linguistic ones. However, it expects for two forms containing a same morpheme according to participants' algorithms to also have a common morpheme in the gold standard.
Nevertheless, it is important to take into account that the morpheme labels provided by the gold standard are syntactically-motivated and not affix specific. For instance, the English suffix "s#" indicates the plural of a noun or the third person singular of a verb and is thus designed in the gold standard with two different labels. Therefore, since our approach is unable to generate syntactically-motivated labels, incorrect pairs of words shall be identified in the morphological analysis where they have morphemes with identical spelling-motivated labels. This phenomenon is known as *syncretism*. In a similar but opposite way, there can be several affixes for a single label. For example, the

[8] Many methods, although unsupervised, consider the compounding effect of the hyphen as a basic prior knowledge.

English suffixes "s#" and "es#" can both represent the third person singular of a verb and are then designed by the same label when they occur in the gold standard. Therefore, once again, since our approach is unable to generate syntactically-motivated labels, many pairs of words having morphemes with syntactically-motivated and equivalent labels shall be identified in the gold standard but not in the morphological analysis where they shall be designed with different spelling-motivated labels. This phenomenon is known as *allomorphy*. Since all morphologies are ambiguous at some point, no method relying on spelling-motivated labels can achieve a perfect score.

In addition to the previous two phenomena (syncretism and allomorphy), the sophisticated evaluation methods handle two other phenomena: morphophonology and ambiguity.

Morphophonology occurs when applying a morphological rule alters the surface form of stems or affixes. For example, in the word "wives", the stem-final 'f' of wife is modified when the plural suffix is added.

Ambiguity happens with homonyms. For example, the French form "fiche" has two possible morphological analysis: one relating it with the verb "*ficher*/to file" and another one relating it with the common noun "*fiche*/card".

Just like the other methods, ours is unable to deal with syncretism, allomorphy and ambiguity. Only two unsupervised method intends, to some extent, to handle allomorphy (Kohonen et al., 2009; Dasgupta and Ng, 2007). Our approach does however handle morphophonology, provided that the phenomenon is regular-enough to be acquired as a different morphological family. Nevertheless, it just transfers the problem to allormorphy. For example, the pair of forms "wife/wives", "knife/knives" or "shelf/shelves", allows to create the pair of related affixes $fe\#, ves\#$. Consequently, the pairs of forms are correctly analyzed as having the same stem. But on the other hand, the corresponding suffixes represent a new case of allomorphy for the singular and plural labels of common nouns.

12.1.1. MorphoChallenge's evaluation metrics

An important change in the 2010 edition has been the adoption for future challenges of a new metric named *EMMA* (Spiegler and Monson, 2010) instead of the *MC* metric (Kurimo et al., 2009) used so far. This decision has been motivated by the fact that *EMMA* correlates far better with the performance of real-world NLP processing tasks which embed the morphological analyses than the *MC* metric does.

This new evaluation metric does bring an important change since it barely correlates with the older *MC* metric. Indeed, *EMMA* presents the same advantages as the *MC* metric but is not susceptible to two types of gaming that have impacted previous MorphoChallenge competitions: ambiguity Hijacking and shared morpheme padding.

As explained in (Spiegler and Monson, 2010), the *MC* metric is not robust when providing ambiguous analysis: it tends to boost recall without harming much precision.

12.2. Results

Our approach has been essentially developed by studying the results produced for French, Spanish and English. Nevertheless, our evaluations have been computed over English, German and Turkish. It is important to note that no tuning has been performed from a language to another[9].

The 2010 edition has introduced a new semi-supervised contest that allows to take advantage of a part of the gold standard. A direct consequence has been a reduced number of participants for the fully unsupervised task (only seven). In addition, because some training corpora or gold standard have changed since the last edition, a direct comparison between the results of the 2009 and 2010 results for the unsupervised methods remains subjective.

	MC			EMMA		
	F-Measure		Rank	F-Measure		Rank
	Best	MorphAcq		Best	MorphAcq	
English	64	59	4	81	78	$\cong 3$
German	47	37	5	65	61	2
Turkish	45	31	4	49	46	3

Figure 4: Evaluation results.

As one can observe, our approach never manage to surpass the state of the art. The *MC* metric actually gave us very poor scores. On the other hand, the *EMMA* metric placed our method among the best methods participating.

As we are unable to understand Turkish and German, the detailed study of our results mainly focused on English. This study showed that most of our cuts are performed as expected, i.e., right on the border of the stems.

As explained, the evaluation tools logically take into account allomorphy and syncretism. Consequently, our biggest loss of recall is our inability to recognize syntactically-equivalent affixes and group them under a same label. In a similar but opposite manner, our main loss of precision is our inability to split a same affix in two syntactically different labels.

The same comment should also apply for Turkish and German. However, the lower recall obtained for both Turkish and German could also be a consequence of a still unidentified drawback. Indeed, whereas these morphologically rich languages rely more on inflection than English does, the sizes of the biggest families obtained for Turkish (10) and German (14) seemed rather small when compared with the size of the biggest one obtained for English (10).

13. Future work

We need first to understand why bigger families could not be generated for Turkish and German. Although it is still unsure, it is likely to be a problem due to data sparsity.

A fairly interesting feature would be to generate morphological analysis based on syntactically-motivated labels and not, as we currently do, spelling-motivated ones. This could allow us to deal with syncretism, allomorphy and the ambiguity brought by homonyms. The study of the state-of-the-art regarding automatic construction/induction of part-of-speech tagger should provide us tracks.

This unsupervised method could also be extended to a semi-automatic one that reuses already-validated morphological rules. Such extension would enhance the results by factorizing the generated families and orientate the acquisition towards missing morphological rules. A similar idea has actually been performed to acquire derivational rules

[9]The $mult$ variable mentioned in sect. 6. had been set to 2.

and links between lemmas by relying on two high-coverage French and Spanish sets of morphological rules(Walther and Nicolas, 2011).

Finally, this methods has only been used on concatenative mechanisms because we could easily restrict the search space when looking for related affixes. If some equivalent restriction could be generalized, we could extend the method to infixes.

14. Conclusion

As confirmed by our experiments and the results presented above, the approach already fulfills its initial goal of acquiring from a raw corpus a data-representative description of the concatenative mechanisms of a morphology. As the samples of morphological families provided show, anybody interested in building a description of the concatenative mechanisms can rely on it to guide and ease its efforts. Just like MorphoChallenge's evaluation tools have pointed out, there are still several aspects that can be improved. Fortunately, the sequential combination of filters provides a convenient way to perform upgrades.

The results obtained with the *EMMA* metric situates our method relatively close to the state-of-the-art without ever surpassing it. The fact that these results have been obtained without any tuning confirms both its potential and its practical interest.

We therefore believe that it can already be a great help when building a new resource and even more when dealing with languages with few documentation.

15. References

Delphine Bernhard. 2008. Simple morpheme labelling in unsupervised morpheme analysis. pages 873–880.

Delphine Bernhard. 2010. Morphonet: Exploring the use of community structure for unsupervised morpheme analysis. In *Multilingual Information Access Evaluation Vol. I, 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Revised Selected Papers*. Springer.

Burcu Can and Suresh Manandhar. 2009. Clustering morphological paradigms using syntactic categories. In *CLEF*.

Mathias Creutz and Krista Lagus. 2005. Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor 1.0. In *Helsinki University of Technology*.

Sajib Dasgupta and Vincent Ng. 2007. High-performance, language-independent morphological segmentation. In *NAACL HLT 2007: Proceedings of the Main Conference*.

Hervé Déjean. 1998. Morphemes as necessary concept for structures discovery from untagged corpora. In *NeMLaP3/CoNLL '98*, Sydney, Australia. The Association for Computational Linguistics.

Vera Demberg. 2007. A language-independent unsupervised model for morphological segmentation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, June. Association for Computational Linguistics.

John Goldsmith. 2006. An algorithm for the unsupervised learning of morphology. *Nat. Lang. Eng.*, 12(4).

Bruno Golenia, Sebastian Spiegler, and Peter Flach. 2009. Ungrade: Unsupervised graph decomposition. In *Working Notes for the CLEF 2009 Workshop, Corfu, Greece*.

Margaret A. Hafer and Stephen F. Weiss. 1974. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10(11-12):371–385.

Zellig S. Harris. 1955. From phoneme to morpheme. *Language*, 31(2):190–222.

Samarth Keshava. 2006. A simpler, intuitive approach to morpheme induction. In *In PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, pages 31–35.

Oskar Kohonen, Sami Virpioja, and Mikaela Klami. 2009. Allomorfessor: towards unsupervised morpheme analysis. In *CLEF'08: Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access*, pages 975–982, Berlin, Heidelberg. Springer-Verlag.

Mikko Kurimo, Ville Turunen, and Matti Varjokallio. 2009. Overview of morpho challenge 2008. In *CLEF'08: Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access*, pages 951–966, Berlin, Heidelberg. Springer-Verlag.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho challenge competition 2005–2010: evaluations and results. In *SIGMORPHON '10*. Association for Computational Linguistics.

Jean-Francois Lavallée and Philippe Langlais. 2010. Unsupervised morphology acquisition by formal analogy. In *Lecture Notes in Computer Science*, page 8 pages.

Constantine Lignos, Erwin Chan, Mitchell P. Marcus, and Charles Yang. 2009. A rule-based acquisition model adapted for morphological analysis. In *CLEF*.

2010. Morphochallenge 2010 website. `http://www.cis.hut.fi/morphochallenge2010/`.

Christian Monson, Alon Lavie, Jaime Carbonell, and Lori Levin. 2008. Evaluating an agglutinative segmentation model for paramor. In *SigMorPhon '08*, pages 49–58, Morristown, NJ, USA. Association for Computational Linguistics.

Lionel Nicolas, Jacque Farré, and Miguel A. Molinero. 2010. Unsupervised learning of concatenative morphology based on frequency-related form occurrence. In *Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, Helsinki, Finland, September.

Sebastian Spiegler and Christian Monson. 2010. Emma: A novel evaluation metric for morphological analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, August.

Sebastian Spiegler, Bruno Golenia, and Peter Flach, 2010. *Unsupervised Word Decomposition with the Promodes Algorithm*, volume I. Springer Verlag, February.

Géraldine Walther and Lionel Nicolas. 2011. Enriching Morphological Lexica through Unsupervised Derivational Rule Acquisition. In *WoLeR 2011 at ESSLLI : International Workshop on Lexical Resources*, Ljubljana, Slovenia.