Unsupervised Acquiring of Morphological Paradigms from Tokenized Text

Daniel Zeman
Ústav formální a aplikované lingvistiky
Univerzita Karlova
Malostranské náměstí 25
CZ-11800 Praha, Czechia
zeman@ufal.mff.cuni.cz

Abstract

This paper describes a rather simplistic method of unsupervised morphological analysis of words in an unknown language. All what is needed is a raw text corpus in the given language. The algorithm looks at words, identifies repeatedly occurring stems and suffixes, and constructs probable morphological paradigms. The paper also describes how this method has been applied to solve the Morpho Challenge 2007 task, and gives the Morpho Challenge results. Although the present work was originally a student project without any connection or even knowledge of related work, its simple approach outperformed, to our surprise, several others in most morpheme segmentation subcompetitions. We believe that there is enough room for improvements that can put the results even higher. Errors are discussed in the paper; together with suggested adjustments in future research.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: I.2.7 Natural Language Processing; I.2.6 Learning; H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

General Terms

Algorithms, Experimentation, Languages

Keywords

Morphology, Morphological analysis, Unsupervised methods

1 Introduction

Morphological analysis (MA) is an important step in natural language processing, needed by subsequent processes such as parsing or translation. Unsupervised approaches to MA are important in that they help process less studied (and corpus-poor) languages, where we have small or no machine-readable dictionaries and tools. Ideally, an unsupervised morphological analyzer (UMA) would learn how to analyze a language just by looking at a large text in that language, without any additional resources, not even mentioning an expert or speaker of the language.

The usual required output of MA is the segmentation of each input word into *morphemes,* i.e. smaller units bearing lexical or grammatical meaning. For instance, the English word *books* would be segmented as *book+s*. A supervised morphological analyzer could further put in the information that the meaning of the suffix *s* is “plural”. There is no way how a UMA could learn the label “plural” from an unlabeled text; however, it can learn the segmentation itself, by observing that many English words appear both with and without the *s* suffix.

In many languages, the morphemes are classified as *stems* and *affixes*, the latter being further subclassified as *prefixes* (preceding stems) and *suffixes* (following stems). A frequent word pattern consists of one stem, bearing the lexical meaning, with zero, one or more prefixes (bearing lexical or grammatical meaning) and zero, one or more suffixes (bearing often grammatical meaning). In languages such as German, *compound words* containing more than one stem are quite frequent. While a stem can appear without any affixes, affixes hardly appear on their own, without stems. For the purposes of this paper, a morphological *paradigm* is a collection of affixes that can be attached to the same group of stems, plus the set of affected stems.

Although the segmentation of the word does not provide any linguistically justified explanation of the components of the word, the output can still be useful for further processing of the text. Having got a paradigm, we can generate all unseen morpheme combinations satisfying that paradigm. We can recognize stems of new words. Thus, we are able to group all words with the same stem. The hope is that one stem means one lexical meaning. All words in a group will share the lexical meaning and differ in the grammatical one. By dropping just part of the meaning (hopefully the less important) we reduce the data sparseness of more complex models like syntactic parsing, machine translation, search and information retrieval.

There is a body of related work that grows faster and faster since the first Morpho Challenge workshop in 2005. Déjean (1998) first induces a list of 100 most frequent morphemes and then uses those morphemes for word segmentation. His approach is thus not fully unsupervised. Keshava and Pitler (2006) combine the ideas of Déjean and Harris. On the Morpho Challenge 2005 datasets, they achieved the best result for English, but they did remarkably worse for Finnish and Turkish. In contrast, Dasgupta and Ng (2007a) report robust performance with best results for all languages. Other UMA learning algorithms exploit the Minimum Description Length (MDL) principle. Specifically, EM is used to iteratively segment a list of words using some predefined heuristics until the length of the morphological grammar converges to a minimum. Brent et al. (1995) were the first to introduce an information-theoretic notion of compression to represent the MDL framework. Goldsmith (2001) also used an MDL-based approach but applied a new compression system with different measuring of the length of the grammar. Creutz (2003) uses probabilistic distribution of morpheme length and frequency to rank induced morphemes. He outperforms Goldsmith for Finnish but gets worse results for English.

The work presented in this paper originated as a student project half a decade ago, without any knowledge of the abovementioned papers (or those few that already existed). The original goal was to learn paradigms, as defined here. It expects a list of words as input, without actually knowing the frequencies of the words, or knowing how to exploit them. It was tested on Czech, where the most important segment boundary is that between stem and suffix (this is not to say that there are no prefixes or compounds in Czech; there are!) Thus the system assumes that there are only two types of words: atomic (they have only the stem) and two-morpheme words (stem + suffix). This is probably the main weakness of the presented system, and will be addressed later in the discussion.

For the sake of Morpho Challenge, we just ran the paradigm finder over the training corpora, and then searched for the learned stems and suffixes in the test data. There were no attempts (yet) to enrich the system using ideas from the related work. Aware of these flaws, we expected to occupy the last positions in all rankings, and we were pleasantly surprised to realize that the system was never the worst one[1] and sometimes even ended above average. This is encouraging, as there clearly are several possible ways of improving the results. We discuss some of them in the concluding section. We leave, however, for the future research to answer whether the system can retain its simplicity while adopting those ideas.

The rest of the paper is organized as follows: in Section 2, we introduce the method of paradigm acquisition; the most interesting part is in Section 2.1, where paradigms are filtered. In Section 3, we explain how we segment a new word, given a set of paradigms. We present the Morpho Challenge results of our system in Section 4 and discuss possible improvements in Section 5.

2 Paradigm acquisition

Recall that we define a paradigm as two sets: a set of suffixes (endings) that can be attached to stems under that paradigm, and a set of eligible stems, to which these suffixed can be attached. As said earlier, we do not permit more than one morpheme boundary (i.e. more than two morphemes) in a word.

Example: The word *bank* can be segmented as *bank, ban+k, ba+nk, b+ank*.

There are *n* possible segmentations of a word of length *n*, and we iterate over them for each training word. For each stem-suffix pair, we record separately that the suffix was seen with the stem, and that the stem was seen with the suffix. At the end, we have for each suffix a list of stems with which they were seen. We group together suffixes with exactly the same sets of stems. The set of suffixes in the group, plus the set of stems they share, is an (unfiltered) paradigm.

[1] This does not apply to the information retrieval task, where our system occupied the worst rank in most rankings.

2.1 Filtering

The list of paradigms obtained so far is huge and redundant. For instance, if all suffixes in a paradigm begin with the same letter, there is another paradigm which differs only in that the letter has been shifted to the stem. The following example is from Finnish:

Paradigm A
Suffixes: a, in, ksi, lla, lle, n, na, ssa, sta
Stems: erikokoisi funktionaalisi logistisi mustavalkoisi objektiivisi rajallisi subjektiivisi tuotannollisi uudenlaisi

Paradigm B
Suffixes: ia, iin, iksi, illa, ille, in, ina, issa, ista
Stems: erikokois funktionaalis logistis mustavalkois objektiivis rajallis subjektiivis tuotannollis uudenlais

Paradigm C
Suffixes: sia, siin, siksi, silla, sille, sin, sina, sissa, sista
Stems: erikokoi funktionaali logisti mustavalkoi objektiivi rajalli subjektiivi tuotannolli uudenlai

Paradigm D
Suffixes: isia, isiin, isiksi, isilla, isille, isin, isina, isissa, isista
Stems: erikoko funktionaal logist mustavalko objektiiv rajall subjektiiv tuotannoll uudenla

We have to filter the paradigms in order to make them useful. We apply the following filtering rules:

2.1.1 More suffixes than stems

Both stem and suffix can be as short as one character. Then how do we recognize that a paradigm with one stem *s* and tens of thousands of suffixes is just crazy? We consider suspicious all paradigms where there is more suffixes than stems. Those paradigms are discarded without compensation.

2.1.2 Uniform letter on the stem-suffix border

As in the Finnish example above, with a uniform letter (or group of letters) on the stem-suffix boundary, we get a set of matching paradigms where the letter(s) is on one or the other side of the boundary. Unlike in the Finnish example, we are not always guaranteed that the corresponding Paradigm B actually does not contain other stems or suffixes, which make the projection irreversible. Example (from Czech):

Paradigm A
Suffixes: l, la, li, lo, ly
Stems: kouři nosi pádi

Paradigm B
Suffixes: il, ila, ili, ilo, ily, ů
Stems: kouř nos pád

In this case, the second paradigm adds the suffix *ů* to the bag, which means that we could not induce Paradigm A from B. On the other hand, the Paradigm B cannot contain additional stems. Consider, for instance, adding a new stem *udobř* to Paradigm B (and removing the *ů* suffix). It would mean that there is a word *udobřil* in the training data. One of the possible segmentations of that word is *udobři-l*, and the same can be done with all the other suffixes, thus we must have had the stem *udobři* in Paradigm A. But we did not.

Similarly, we can proceed from the longer suffixes to the shorter ones. When all suffixes begin with the same letter, there must be a corresponding Paradigm B, where the letter is shifted to the stems. The Paradigm B can contain additional stems, as in the following example:

Paradigm A
Suffixes: il, ila, ili, ilo, ily
Stems: kouř nos pád

Paradigm B
Suffixes: l, la, li, lo, ly
Stems: kouři nosi pádi sedě

While Paradigm B can add stems, it cannot add suffixes. Consider adding a suffix *t* (and removing the stem *sedě*). It would mean that the words *kouřit, nosit, pádit* were in the training data, and thus the suffix *it* should have appeared in Paradigm A.

Now it is obvious that the boundary letters create room for paradigm filtering. The question is, should we prefer longer stems, or longer suffixes? We decided to prefer shorter suffixes. If all suffixes in a paradigm begin with the same letter, we discard the paradigm, being sure that there is another paradigm with those border letters in stems. That other paradigm may contain some other stems as well, which further strengthens our conviction that the border letter was not a genuine part of the suffixes.

2.1.3 Subsets of paradigms

A frequent problem is that stems have not been seen with all applicable suffixes. Consider the following example (from real Czech data):

A.suffixes = {*ou, á, é, ého, ém, ému, ý, ých, ým, ými*}
B.suffixes = {*ou, á, é, ého, ém, ému, ý, ých, ým*}
C.suffixes = {*ou, á, é, ého, ém, ý, ých, ým, ými*}
D.suffixes = {*ou, á, é, ého, ém, ý, ých, ým*}

As a matter of fact, stems of all four paradigms should belong to the paradigm A but not all of them occurred with all A suffixes. As one important motivation of UMA is to cover unknown words, it is desirable to merge the subset paradigms with their superset A. Unfortunately, this can sometimes introduce stem+suffix combinations that are not permitted in the given language.

When talking about set inclusion on paradigms, we always mean the sets of suffixes, not stems. If the suffixes of Paradigm B form a subset of suffixes of Paradigm A, and there is no C, different from A, such that B is also subset of C, then merge A with B (which means: keep suffixes from A, and stems from both).[2]

The implementation of this rule is computationally quite complex. In order to identify subset relations, we would have to step through n^2 paradigm pairs (n is the current number of paradigms, over 60,000 for our Czech data), and perform k comparisons for each pair (in half of the cases, k is over 5). As a result, tens of billions of comparisons would be needed.

That is why we do not construct the complete graph of subsets. We sort the paradigms with respect to their size, the largest paradigm having size (number of suffixes) k. We go through all paradigms of the size k–1 and try to merge them with larger paradigms. Then we repeat the same with paradigms of the size k–2, and so on till the size 1. The total number of comparisons is now much lower, as the number of paradigms concurrently decreases.

For each paradigm, we check only the closest supersets. For instance, if there is no superset larger by 1, and there are two supersets larger by 2, we ignore the possibility that there is a superset larger by 3 or more. They are linked from the supersets larger by 2. If an ambiguity blocks simplifying the tree, it is not a reason to block simplifying on the lower levels.

2.1.4 Single suffix

Paradigms with a single suffix are not interesting. They merely state that a group of words end in the same letters. Although we could identify unknown words belonging to the same group and possibly segment them along the border between the non-matching and matching part, there is not much to be gained from it. There is also no guarantee that the matching end of the word is really a suffix (consider a paradigm with suffix *n* and “stems” from thousands of words ending in *n*). So we discard all single-suffix paradigms and thus further simplify the paradigm pool.

[2] If the other superset C exists, it is still possible that the merging will be enabled later, once we succeed to merge A with C.

2.2 More paradigm examples

For illustration, we provide some of the largest (with most suffixes) paradigms for the four languages of Morpho Challenge 2007 and Czech:

English

e, ed, es, ing, ion, ions, or
calibrat consecrat decimat delineat desecrat equivocat postulat regurgitat

e, ed, es, ing, ion, or, ors
aerat authenticat disseminat enumerat percolat pollinat promulgat

0[3], d, r, r's, rs, s
analyze chain-smoke collide customize energize enquire naturalize scuffle telecommute transcribe

Finnish

0, a, an, ksi, lla, lle, n, na, ssa, sta, t
asennettava avattava hinattava koordinoiva korvattava leijuva mahdollistama painettava pelattava runtelema saartama sijoitettava tilattava tulostettava tuotava uusiutuva vaadittava verrattava vuokrattava

en, ksi, lla, lle, lta, n, na, ssa, sta, sti, t
aatteellise ainaise aluepoliittise ennenaikaise fysikaalise geologise hengenvaarallise kemiallise keskiaikaise kosmise kuninkaallise kuvallise kuvitteellise legendaarise motorise muotoise oikeudellise oloise parlamentaarise patologise päiväkirurgise päätoimise radioaktiivise sosiaalipoliittise sosialidemokraattise toissijaise uudenlaise vapaavalintaise yliluonnollise

a, en, in, ksi, lla, lle, lta, na, ssa, sta
ammatinharjoittaji avustavi jakavi muuttaji omaavi parannettavi puolueettomi sairastavi sijoittuvi varkauksi verrattavi

German

0, m, n, r, re, rem, ren, rer, res, s
aggressive bescheidene deutliche dunkle flexible langsame mächtige ruhige schwierige strenge umweltfreundliche

0, e, em, en, er, es, keit, ste, sten
entsetzlich gutwillig lebensfeindlich massgeblich reichhaltig unbarmherzig unerbittlich unermüdlich vorhersehbar warmherzig

0, m, n, r, re, ren, res, rweise, s
anständige erfreuliche glückliche natürliche professionelle sinnvolle traditionelle traurige vernünftige vorsichtige

Turkish

0, de, den, e, i, in, iz, ize, izi, izin
anketin becerilerin birikimlerin gereksinimin giysilerin görüntülerin güvenin objektifin olabileceğin yemeğin

[3] 0 means empty suffix.

0, dir, n, nde, ndeki, nden, ne, ni, nin, yle
aleti arabirimi etiketi evreleri geçilmesi geçişleri iletimi iliği kanseri protokolleri segmenti sürmesi temini yetiştiriciliği şiddeti

0, a, da, daki, dan, ı, ın, ız, ızı
bakışın baskıların detayların fırının kabloların kağıtların koleksiyonların olasılığın operasyonların sayımın tezgahın yaklaşımın yanıtların yazılımın

Czech

ou, á, é, ého, ém, ému, ý, ých, ým, ými
gruzínsk italsk lékařsk ministersk městsk někter olympijsk poválečn pražsk tropick závěrečn člensk

0, a, em, ovi, y, ů, ům
divák dlužník obchodník odborník poplatník právník předák vlastník útočník činovník

a, ami, ou, u, y, ách, ám
buňk dívk otázk podmínk pohledávk přestávk schránk stovk válk

3 Segmenting a word

Given a set of paradigms for a language, how do we apply it to segment a word in that language? Actually, we only use the sets of all stems and all suffixes in the Morpho Challenge task. We do not exploit the information that a stem and a suffix occurred in the same paradigm. Yet the acquisition of paradigms described in the previous section is still important, as it greatly reduces the number of learned stems and suffixes.

Again, we consider all possible segmentations of each analyzed word. For each stem-suffix pair, we look up the table of learned stems and suffixes. If both stem and suffix are found, we return that particular segmentation as a possible analysis. (Note that more than one segmentation can satisfy the condition, and thus ambiguous analyses are possible.) If no analysis is found this way, we return analyses with known suffixes or known stems (but not both). If no analysis is found either way, we return the atomic analysis, i.e. the entire word is a stem, the suffix is empty.

4 Results

The Morpho Challenge 2007 task does not (and actually cannot) require that the morphemes in segmentation be labeled in any particular way. Due to possible phonological changes caused by inflection of words, the segmenters are not even required to denote the exact position of the morpheme border in the word. Therefore, the only information that can be compared with a gold standard is the number of morphemes in the word, and the fact that two words share a morpheme with the same label on specified positions. The precise description of the evaluation algorithm is available at the Morpho Challenge website.[4] We present only the results of the Competition 1 in this paper.[5] The complete results are available at http://www.cis.hut.fi/morphochallenge2007/results.shtml.

In comparison to other systems, our system usually did better w.r.t. recall than w.r.t. precision. The best rank achieved by our system was for Turkish, while for the other languages we ended up below average. For each language, we provide our rank and the number of ranked systems (in addition to the percentages). P is precision, R is recall, F is their harmonic mean.

[4] http://www.cis.hut.fi/morphochallenge2007/evaluation.shtml

[5] In the Competition 2, the segmentation results are evaluated indirectly by using them in an information retrieval task. Our system was among the poorest in all rankings of the Competition 2 but we have currently no plausible explanation.

	English		
	P	R	F
%	52.98	42.07	46.90
rank	10	5	9
# ranked	13		

German		
P	R	F
52.79	28.46	36.98
9	9	9
12		

	Finnish		
	P	R	F
%	58.84	20.92	30.87
rank	8	6	6
# ranked	9		

Turkish		
P	R	F
65.81	18.79	29.23
8	2	2
9		

The processing of each language took from several minutes to several hours. Finnish needed the most time due to its enormous agglutinative morphological system. German is not an agglutinative language but its long compound words also increased time requirements. Turkish was faster not because it is less complex but simply because of the much smaller data set.

Not surprisingly, the slowest part of the algorithm is the subset pruning.

5 Discussion

The presented approach is a truly unsupervised one, as it does not need any language-specific tuning ever (compare with the lists of most frequent morphemes in some related work). However, there are many lessons to be learned from other systems and tested during future research. Some of those ideas do not violate the independence on any particular language, and hopefully will not complicate the computation too much. Some ideas follow:

- Our system does not (but it should) exploit the word/morpheme frequencies in the corpus. Very rare words could be typos and could introduce nonsensical morphemes.
- Our system reduces morpheme segmentation to just one stem (mandatory) and one suffix (optional). Such limitation is too severe. At least we ought to enable prefixes. It could be done by repeating the process described in this paper, but now the second part would be a stem and the first part an affix. Using the new model, we could recognize prefixes in the stems of the old model, and using the old model, we could recognize suffixes in the new one.
- Even that is fairly limited. There are composite suffixes (as in English *compose+r+s*) and composite prefixes (as in German *ver+ab+schieden*). A good morphological analyzer should identify them (not to mention that they are likely to appear in the gold standard data).
- Finally, compounds make the possible number of morphemes virtually unlimited. (An anecdotic German example is *Hotentot+en+potentat+en+tante+n+atentät+er*.) A possible partial solution is to do a second run through the stems and identify combinations of two or more smaller stems. However, as seen from the example, suffixes are involved in compound creation as well.
- Morphological grammars of many languages contain rules for phonological changes (for instance, *deny* vs. *deni* in English *denial*, Czech *matk+a, matc+e, matč+in*, German *Atentat* vs. *Atentät+er*). Supervised MA systems have incorporated such rules in order to succeed (e.g., see Koskenniemi (1983) or Hajič (2004)). Dasgupta and Ng (2007) induce phonological rules for suffixes longer than 1 character, however, the above Czech example suggests that it may be needed for suffixes of length 1 as well.

6 Conclusion

We have presented a paradigm acquisition method that can be used for unsupervised segmentation of words into morphemes. The approach is very simple; however, even such a simple system turned out to be reasonably successful. It gives us the hope that by incorporating the ideas discussed in Section 5, we can catch up with at least some of the better systems from Morpho Challenge 2007.

7 Acknowledgements

This research has been supported by the following projects: Czech Academy of Sciences, the “Information Society” program, project No. 1ET101470416, and Ministry of Education of the Czech Republic, project MSM0021620838.

8 References

Delphine Bernhard. 2006. *Unsupervised Morphological Segmentation Based on Segment Predictability and Word Segment Alignment.* In: PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes. Southampton, England.

Michael R. Brent, Sreerama K. Murthy, Andrew Lundberg. 1995. *Discovering Morphemic Suffixes: A Case Study in Minimum Description Length Induction.* In: Proceedings of the 5th International Workshop on AI and Statistics. Fort Lauderdale, Florida.

Mathias Creutz. 2003. *Unsupervised Segmentation of Words Using Prior Distributions of Morph Length and Frequency.* In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. Sapporo, Japan.

Mathias Creutz, Krista Lagus. 2005. *Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor 1.0.* In: Computer and Information Science, Report A81, Helsinki University of Technology.

Sajib Dasgupta, Vincent Ng. 2007a. *High-Performance, Language-Independent Morphological Segmentation.* In: Proceedings of NAACL HLT 2007, pp. 155–163. Rochester, New York.

Sajib Dasgupta, Vincent Ng. 2007b. *Unsupervised Word Segmentation for Bangla.* In: Proceedings of ICON. Adelaide, Australia.

Hervé Déjean. 1998. *Morphemes as Necessary Concepts for Structures Discovery from Untagged Corpora.* In: Workshop on Paradigms and Grounding in Natural Language Learning, pp. 295–299.

Vera Demberg. 2007. *A Language-Independent Unsupervised Model for Morphological Segmentation.* In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, pp. 920–927. Praha, Czechia.

Dayne Freitag. 2005. *Morphology Induction from Term Clusters.* In: Proceedings of CoNLL, pp. 128–135. Ann Arbor, Michigan.

John Goldsmith. 2001. *Unsupervised Learning of the Morphology of a Natural Language.* In: Computational Linguistics 27(2), 153–198.

Jan Hajič. 2004. *Disambiguation of Rich Inflection (Computational Morphology of Czech).* 330 pp. Univerzita Karlova, MFF, Ústav formální a aplikované lingvistiky. Praha, Czechia.

Zellig Harris. 1955. *From Phoneme to Morpheme.* In: Language, 31(2): 190–222.

Samarth Keshava, Emily Pitler. 2006. *A Simple, Intuitive Approach to Morpheme Induction.* In: PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes. Southampton, England.

Kimmo Koskenniemi. 1983. *Two-level Morphology: A General Computational Model for Word-form Recognition and Production. Publication No. 11.* University of Helsinki, Department of General Linguistics, Helsinki, Finland.

Patrick Schone, Daniel Jurafsky. 2001. *Knowledge-free Induction of Inflectional Morphologies.* In: Proceedings of the NAACL, pp. 183–191. Pittsburgh, Pennsylvania.

Matthew G. Snover, Michael R. Brent. 2001. *A Bayesian Model for Morpheme and Paradigm Identification.* In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, pp. 482–490. Toulouse, France.

David Yarowsky, Richard Wicentowski. 2000. *Minimally Supervised Morphological Analysis by Multimodal Alignment.* In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 207–216. Hong Kong, China.